ONTOLOGICAL FOUNDATIONS FOR STRUCTURAL CONCEPTUAL MODELS

GIANCARLO GUIZZARDI

ONTOLOGICAL FOUNDATIONS FOR STRUCTURAL CONCEPTUAL MODELS

Telematica Instituut Fundamental Research Series

001 G. Henri ter Hofte, *Working apart together: Foundations for component groupware*
002 P. J.H. Hinssen, *What difference does it make? The use of groupware in small groups*
003 D.D. Velthausz, *Cost Effective Network Based Multimedia Information Retrievel*
004 L. van de Wijngaert, *Matching Media: Information Need and New Media Choice*
005 R.H.J. Demkes, *COMET: A Comprehensive Methodology for Supporting Telematics Investment Decisions*
006 O. Tettero, *Intrinsic Information Security: Embedding Information Security in the Design Process of Telematics Systems*
007 M. Hettinga, *Understanding Evolutionary Use of Groupware*
008 A. van Halteren, *Towards and Adaptable QoS Aware Middleware for Distributed Objects*
009 M. Wegdam, *Dynamic Reconfiguration and Load Distribution in Component Middleware*
010 I. Mulder, *Understanding Designers, Designing for Understanding*
011 R. Slagter, *Dynamic Groupware Services – Modular Design of Tailorable Groupware*
012 N.K. Diakov, *Monitoring Distributed Object and Component Communication*
013 C.N. Chong, *Experiments in Rights Control: Expression and Enforcment*
014 C. Hesselman, *Distribution of Multimedia Streams to Mobile Internet Users*

Ontological Foundations for Structural Conceptual Models

Giancarlo Guizzardi

Enschede, The Netherlands, 2005

CTIT PhD Thesis Series, No. 05-74
Telematica Instituut Fundamental Research Series, No. 015 (TI/FRS/015)

Cover Design: Studio Oude Vrielink, Losser and Jos Hendrix, Groningen
Book Design: Lidwien van de Wijngaert and Henri ter Hofte
Printing: Universal Press, Veenendaal, The Netherlands
Cover Painting: **"In-formation"** by Melissa Guizzardi, Trento, Italy, 2005.

Samenstelling promotiecommissie:

Voorzitter, secretaris:	prof.dr.ir. A.J. Mouthaan (Universiteit Twente)
Promotor:	prof.dr.ir. C.A. Vissers (Universiteit Twente)
Assistent Promotoren:	dr. L. Ferreira Pires (Universiteit Twente)
	dr.ir. M.J. van Sinderen (Universiteit Twente)
Leden:	prof.dr. P.M.G. Apers (Universiteit Twente)
	prof.dr. H. Herre (University of Leipzig)
	prof.dr. G. Wagner (Brandenburg University of Technology at Cottbus)
	prof.dr. R.J. Wieringa (Universiteit Twente)
	dr. P.E. van der Vet (Universiteit Twente)

CTIT PhD.-thesis series, No. 05-74
ISSN 1381-3617; No. 05-74
Centre for Telematics and Information Technology, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands

Telematics Instituut Fundamental Research Series, No. 015
ISSN 1388-1795; No. 015
ISBN 90-75176-81-3
Telematics Instituut, P.O. Box 589, 7500AN Enschede, The Netherlands

ONTOLOGICAL FOUNDATIONS FOR STRUCTURAL CONCEPTUAL MODELS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. W.H.M. Zijm,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 27 oktober 2005 om 16.45 uur

door
Giancarlo Guizzardi
geboren op 22 april 1975
te Vitória, Brazilië

Dit proefschrift is goedgekeurd door:
prof.dr.ir. C.A. Vissers (promotor), dr.L. Ferreira Pires (assistent-promotor) en dr.ir. M. J. van Sinderen (assistent-promotor).

To Anizio Guizzardi and
Rita de Cássia R. L. Guizzardi,
For making the world so big
and the earth so small

To Renata S.S. Guizzardi,
For being the heart of my reasoning
and the reason of my heart

As Coisas (Arnaldo Antunes, 1993)

As coisas têm peso,
massa, volume, tamanho,
tempo, forma, cor,
posição, textura, duração,
densidade, cheiro, valor,
consistência, profundidade,
contorno, temperatura,
função, aparência, preço,
destino, idade, sentido.
As coisas não têm paz.

The Things (Arnaldo Antunes, 1993)

The things have weight,
mass, volume, size,
time, shape, color,
position, texture, duration,
density, smell, value,
consistency, depth,
boundaries, temperature,
function, appearance, price,
fate, age, significance.
The things have no peace.

Preface

"The idea came to me as one switches on a light, one day
when by chance there fell into my hands an old dusty diagram,
the work of some unknown predecessor of mine…
Since a chemist does not think, indeed does not live without models,
I idly went about representing them for myself,
drawing on paper the long chains of silicon, oxygen,
iron and magnesium, with the nickel caught between their links…
and I did not feel much different from the remote
hunter of Altamira who painted an antilope
on the rock wall so that the next day's hunt would be lucky."
Primo Levi, The Periodic Table, 1975

A model is an abstraction of reality according to a certain conceptualization. Once represented as a concrete artifact, a model can support communication, learning and analysis about relevant aspects of the underlying domain. As in the passage above by the brilliant Italian writer Primo Levi, a represented model (a *dusty diagram*) created by an *unknown predecessor* is a medium to preserve and communicate a certain view of the world, and can serve as a vehicle for reasoning and problem solving, and for acquiring new knowledge (maybe having striking new *ideas*!) about this view of the world.

As a concrete artifact, a represented model must be expressed in some "suitable" language. For instance, in the chemical domain evoked by Levi, a language that we would consider "suitable" would include features such as: comprise constructs that represent concepts such as atoms, molecules and their links; be clear and intuitive for chemists to use; not allow for the construction of diagrams that represent situations that are deemed impossible by the laws of chemistry, etc... In summary, we could say that such a language should be truthful to the domain in reality it is supposed to represent.

In the work presented in this thesis, we systematically study some of the relations between a modeling language and a set of real-world phenomena in a given domain. Thus, one of the questions which are addressed here is: how can we assess the *ontology adequacy* of a given modeling language? In simple terms, ontological adequacy is a measure of how close the models produced using a modeling language are to the situations in reality they are supposed to represent.

In this thesis, we show how modeling languages can be evaluated and (re)designed with the purpose of improving their ontological adequacy, by proposing a systematic evaluation method for comparing a metamodel of the concepts underlying a language to a *reference ontology* of the corresponding domain in reality.

However, unlike Levi, the languages in which we have interest here are not the ones of chemistry, but the general *conceptual modeling* languages that are used in computer science to create domain models in areas such as artificial intelligence, software engineering, domain engineering, database design and integration, enterprise modeling, information systems engineering, among many others. Accordingly, the *reference ontology* which is needed is not one of a specific domain such as chemistry, but a formal (i.e., domain independent) system of categories and their ties that can be used to construct models of specific domain in realities, i.e., a *Foundational Ontology*. Moreover, since conceptual modeling languages are intended to support *human activities* such as communication, domain learning and problem solving, this foundational ontology must be one that takes human cognition explicitly into account.

The main objective of this work is, thus, to contribute to the theory of conceptual modeling by proposing a reference ontology that can be used to provide *ontological foundations* for general conceptual modeling concepts, and to analyze, (re) design and provide real-world semantics for general conceptual modeling languages. In particular, we focus here on *structural conceptual models* (also named domain models, information models, semantic data models). Consequently, the corresponding foundational ontology developed here is an *ontology of endurants* (objects), as opposed to perdurants (events, processes). More specifically, this ontology addresses issues such as: (i) the general notions of types and their instances; (ii) objects, their intrinsic properties and property-value spaces; (iii) the relation between identity and classification; (iii) distinctions among sorts of types (e.g., kinds, roles, phases, mixins) and their admissible relations; (iv) distinctions among sorts of relational properties; (v) Part-whole relations.

This ontology has been developed by adapting and extending a number of theories coming from the areas of formal ontology in philosophy. The chosen theories have been corroborated by thought experiments in philosophy and/or are supported by empirical evidence in cognitive psychology. Once developed, every sub-theory of the ontology is used in the creation of methodological tools (e.g., modeling profiles, guidelines and design patterns). The expressiveness and relevance of these tools are shown throughout the thesis to solve some classical recurrent problems found in the conceptual modeling and ontological engineering literature.

The thesis demonstrates the applicability and usefulness of both the language evaluation and re-engineering method, and of the foundational ontology proposed by developing a case study. The target of this case study is the fragment of the Unified Modeling Language (UML) that deals with the construction of structural models. In this case study we (i) evaluate the ontological correctness of current conceptual representations produced using the language; (ii) provide guidelines for how the UML constructs should be used in conceptual modeling; (iii) justify extensions to the language in order to capture important ontological distinctions.

As a result of this process, we manage to produce a conceptually cleaner, semantically unambiguous and ontologically well-founded version of the UML fragment that is mostly used for conceptual modeling, namely, the UML class diagrams.

After that, we also carry out a second case study which uses this UML version proposed to analyze and integrate several semantic web ontologies in the scope of a context-aware services platform.

The subject of this thesis is of an inherently complex, abstract and interdisciplinary nature. For this reason, we felt obliged to carefully justify our ontological choices and, in many passages, to construct a minimum theoretical background for the argumentation that would follow. As a result, the text of this dissertation is longer than the average PhD thesis in computer science. Nonetheless, we think this disadvantage is compensated by the benefits of having a detailed and precise understanding of the issues discussed here.

Acknowledgements

"The lines I trace with my feet
walking to the museum
are more important and more beautiful
than the lines I find there
hang up on the walls"
Friedensreich Hundertwasser

This thesis is about *objects* (things, endurants). However, I strongly believe that even more important than the produced object (the artifact), doing a PhD is about the *process* of becoming something else. There are several people that deserve my immense gratitude and appreciation for helping me in many possible ways in this transformation process of growing as a person and as a scientist that culminated with this thesis.

I will start by thanking my promoter Chris Vissers for giving me the opportunity to develop the research reported in this PhD thesis. I am a great admirer of his seriousness towards science and his attitude with respect to understanding things in their essence. I also would like to thank my daily supervisors: Marten van Sinderen and Luís Ferreira Pires, who painstakingly read and commented almost every paragraph of the original monograph. Their comments played an essential role in improving the quality and accessibility of this dissertation. I have learned many things from them, and I have a deep respect for their commitment and loyalty to their research group.

Two people played a fundamental role in the development of the ideas presented here: Gerd Wagner and Nicola Guarino. Gerd has been a great scientific mentor and I doubt I would have made this particular PhD thesis at all if it were not for him. I am indebted to his patience, commitment, encouragement, strategic guidance, and for the various fruitful and exciting discussions in Eindhoven or in countless phone meetings we had over the years. Nicola's pioneering work on the application of philosophical ontology in computer science has inspired me for the past eight years. It was a great pleasure to meet and work with him since 2003. We have met in a very critical period of my PhD, and his immediate understanding and strong support for my ideas were of indescribable importance for giving me courage and desire for pursuing this specific thesis. These two people are among the best examples I know of how scientific excellence and kindness go very well together.

I am grateful to Heinrich Herre for giving me the opportunity of an invaluable collaboration in the context of the GOL/GFO research project in Leipzig. I have a great admiration for how, despite all the obstacles, he has manage to lead one of the most interesting, serious and inspiring research efforts in the area of formal ontology nowadays.

I am honoured to have Professor Peter Apers, Professor Heinrich Herre, Dr. Paul van der Vet, Professor Gerd Wagner and Professor Roel Wieringa in my committee. I am grateful to them for devoting their time to read my dissertation and to participate in my defense.

In these years in the ASNA group in the University of Twente, I had the opportunity and pleasure to meet and interact with a number of interesting people that also had a positive influence on my work and my life. A non-exhaustive list of these people includes José Gonçalves Pereira Filho, João Paulo Andrade Almeida, Alex Slingerland, Diego Ríos, Patricia Dockhorn Costa, Pablo Gomes Ludermir, José Laurino dos Santos, Clever Farias, Marcos Salvador, Ciro Barbosa, David Remondo Bueno, Ricardo Neisse, Gloria Tuquerres, Stanislav Pokraev, Remco Dijkman, Nilolay Diakov, Christian Tsolov, Maarten Wegdam, Remco van de Meent, Robert Párhonyi, Val Jones, Minh van Le, Helen Kool, Marlous Weghorst, and Annelies Klos. Two people among these that deserve my special gratitude are José Gonçalves Filho and João Paulo Almeida. José has been an important and constant influence in my scientific career in these nine years of friendship and collaboration. It was a pleasure to have the opportunity to work closely with him again in the years of 2002 and 2003 when he held a visiting professor position at the University of Twente. João, besides being an invaluable friend, was my officemate for a couple of years. I will always remember with great joy our exciting (and endless) discussions about life, the universe, and specially, the *meaning of everything*.

During many occasions since 2003, I have had the opportunity to visit and work in the Laboratory for Applied Ontology (LOA) led by Nicola Guarino in Trento. The fruitful interdisciplinary environment set at LOA had an immediate positive influence on my work. Moreover, the friends and colleagues I have made there during my first extended stay (from May to September, 2003) strongly influenced my decision to return in other research visits, and to accept with great pleasure a research position there since april 2005. I am grateful to Claudio Masolo, Laure Vieu, Emanuele

Bottazzi, Alessandro Oltramari, Roberta Ferrario, Stefano Borgo, John Mylopoulos, Paolo Giorgini, Luiz Olavo Bonino, Laurent Prevot, Pawel Garbacz and Robert Trypuz for the many hours of intellectually stimulating and pleasant discussions in Trento. Claudio and Laure deserve my special gratitude for the way they made me fell at home there from the beginning.

I am indebted to Ricardo Falbo for many years of friendship and stimulating collaboration. He was the person who introduced me to the topic of ontologies eight years ago.

I also would like to thank some additional friends that contributed significantly for making so much worth the experience of the past five years in The Netherlands and in Italy. These include Dalva Silva Souza, Gustavo Moraes, Anton Bovens, Mariana Costa, Thomas Bovens, Peter de Jong, Diana Ros, Marike Hasperhoven, Flavia Machado (in The Netherlands), Luiz Olavo and Luciana Bonino (our Italian Family). A special recognition goes to my quasi-brother Leonardo Dardengo. He was an essential part of this whole adventure, and it was very comforting having him close by for its dark and joyful moments.

I am immeasurably proud of my brother Antonini and my sisters Melissa and Tricia. I am grateful to them, for all their love, care and everlasting support in these many years of physical distance. I am also thankful to Melissa for her work of art entitled *In-formation*, which was specially created for the cover of this thesis.

Finally, there are no words I could possibly write to articulate my gratitude to my parents, Anizio and Rita, and to my beloved wife Renata. This thesis is dedicated to them, for everything they taught me.

Giancarlo Guizzardi
Trento, Italy, September 2005

Contents

Chapter 1

Introduction

This thesis contributes to the definition of general ontological foundations for the area of conceptual modeling. This chapter presents the background of the thesis (section 1.1) and motivates the relevance of the work reported here (section 1.2). It also defines the main objectives of our research (section 1.3) and its scope (section 1.4). The chapter concludes by presenting the approach we follow to accomplish these objectives together with an overview of the thesis structure (section 1.5).

1.1 Background

Telematics is an area concerned with the support of the interactions between people or automated processes or both, by applying information and communication technology (ICT). In general terms, information and communication technology has a radical impact on its users, their work, and their working environments. In its various manifestations, ICT processes data, gathers information, stores collected materials, accumulates knowledge, and expedites communication. In fact, it plays a role in many, if not most, of the everyday operations of today's business world (Chen, 2000).

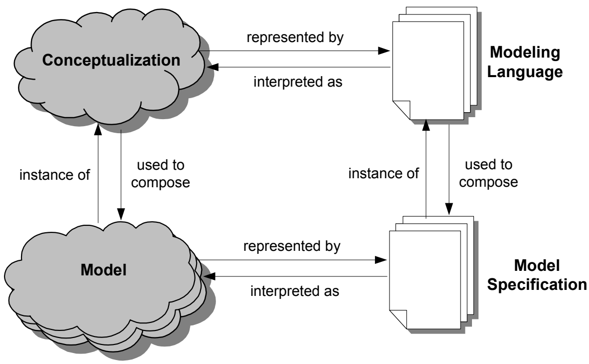
Telematic Systems are developed to support the enaction of *telematic services*. Users of telematic services are placed in a social context and, in order to satisfy the needs of these users, telematic services have to be strongly related to the design of the activities in the social context that these services support (Vissers et al., 2000).

An important constituent of the context in which a telematics service is embedded is the so-called *subject domain* (or *universe of discourse*) of this service. For instance, a *medical treatment reservation service* refers to concepts in a universe of discourse comprising entities such as patients, treatments, medical insurance, physicians, medical units, among others. The correct operation of this service, thus, depends on the correct *representation* of this subject domain. In particular, the representations of situations in reality used by a given system should stand for actual state of affairs of its subject domain. For example, if two people are said to be married in a system, or if a student is said to have graduated by a given university, these should reflect the actual state of affairs holding in reality.

Abstractions of a given portion of reality are constructed in terms of concepts, i.e., abstract representations of certain aspects of entities that exist in that domain. We name here a *conceptualization* the set of concepts used to articulate abstractions of state of affairs in a given domain. The abstraction of a given portion of reality articulated according to a domain conceptualization is termed here a *model.*

Conceptualizations and models are abstract entities that only exist in the mind of the user or a community of users of a language. In order to be documented, communicated and analyzed, these entities must be captured in terms of some concrete artifact. The representation of a conceptual model is named here a *model specification*. Moreover, in order to represent a specification, a *specification* (or *modeling*) *language* is necessary. The relation between conceptualizations, models, specifications and modeling languages is depicted in figure 1.1 below.

Figure 1-1 Relations between conceptualization, Model, Modeling Language and Specification



A language can be seen as determining all possible specifications (i.e., all *grammatically valid* specifications) that can be constructed using that language. Likewise, a conceptualization can be seen as determining all possible models (standing for state of affairs) admissible in that domain (Guarino, 1998). Therefore, for example, in a conceptualization describing genealogical relations, there cannot be a model in which a person is his own biological parent, because such a state of affairs cannot obtain in reality.

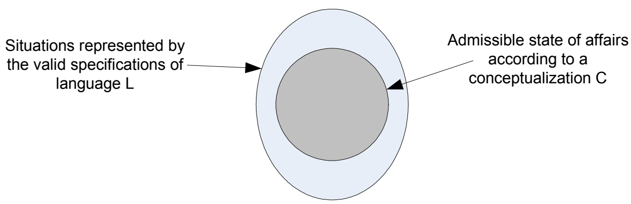
In this thesis, we are interested in the so-called class of *conceptual modeling languages*, as opposed to, for instance, languages aimed primarily at systems design and implementation. In a seminal paper, John Mylopoulos (Mylopoulos, 1992) defines the discipline of conceptual modeling as

"the activity of formally describing some aspects of the ***physical*** *and* ***social*** *world around us for purposes of* ***understanding*** *and* ***communication****...Conceptual modelling supports structuring and inferential facilities that are* ***psychologically grounded****. After all, the descriptions that arise from conceptual modelling activities are intended to be used* ***by humans, not machines...*** *The adequacy of a conceptual modelling notation rests on its contribution to the construction of models of reality that promote a common understanding of that reality among their human users."*

The specification of a conceptual model is, hence, a description of a given subject domain independent of specific design or technological choices that should influence particular telematics systems based on that model. Conceptual specifications are used to support *understanding* (*learning*), *problem-solving*, and *communication*, among stakeholders about a given subject domain. Once a sufficient level of understanding and agreement about a domain is accomplished, then the conceptual specification is used as a blueprint for the subsequent phases of a system's development process.

The quality of a telematics system and services, therefore, depend to a large extent on the quality of the conceptual specifications on which their development is based. The latter, in turn, is strongly dependent of the quality of the conceptual modeling language used in its description. For instance, if a modeling language is imprecise and coarse in the description of a given domain, then there can be specifications of the language which, although grammatically valid, do not represent admissible state of affairs. This situation is depicted in figure 1.2.

Figure 1-2
Consequences of a Modeling Language as an imprecise representation of a domain conceptualization

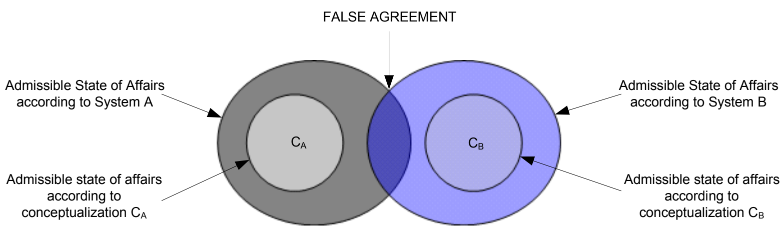


The difference between the two sets illustrated in figure 1.2 gives us a measure of the truthfulness to reality, or the so-called *domain appropriateness* of a given conceptual modeling language (Krogstie, 2000). In summary, we can state that the more we know about a given domain and the more precise we are on representing it, the bigger the chance that we have of constructing computational systems and services that are consistent with the reality of that domain.

Having a precise representation of a given conceptualization becomes even more critical when we want to integrate different independently developed models (or systems based on those models). Suppose the situation in which we want to have the interaction between two independently developed systems, which commit to two different conceptualizations. In order for these systems to function properly together, we must guarantee that they ascribe compatible meanings to real-world entities of their shared subject domain. In particular, we want to reinforce that they have compatible sets of admissible situations, whose union (in the ideal case) equals the admissible state of affairs delimited by the conceptualization of their shared subject domain. The ability of systems to interoperate (i.e., operate together), while having compatible real-world semantics is known as *semantic interoperatibility* (Vermeer, 1997).

Now, suppose we have the situation depicted in figure 1.3. C_A and C_B represent the conceptualizations of the subject domains of systems A and B, respectively. As illustrated in figure 1.3, these conceptualizations are not compatible. However, because these systems are based on poor representations of these conceptualizations, their sets of possible situations considered overlap. As a result, systems A and B agree exactly on situations that are neither admitted by C_A nor by C_B. To put it simply, although these systems seem to have a shared view of reality, the portions of reality that each of them aims at representing are not compatible together. This problem, termed *The False Agreement Problem* was first highlighted in (Guarino, 1998).

Figure 1-3 False Semantic Agreement between two Communicating Entities



Another important quality criterion for conceptual specifications is *pragmatic efficiency*. Since these specifications are meant to be used by humans, their conceptual clarity and ability to support communication, understanding and reasoning about the domain plays a fundamental role. This quality criterion of conceptual specifications is also termed *comprehensibility appropriateness* (Krogstie, 2000).

Thus, on one hand, a modeling language should be sufficiently expressive to suitably characterize the conceptualization of its subject domain, on the other hand, the semantics of the produced specifications should be clear, i.e., it should be easy for a specification designer to recognize what language constructs mean in terms of domain concepts. Moreover, the specification produced using the language should facilitate the user in understanding and reasoning about the represented state of affairs.

In this thesis we defend that the suitability of a conceptual modeling language to represent a set of real-world phenomena in a given domain (i.e., its domain and comprehensibility appropriateness) can be systematically evaluated by comparing the level of homomorphism between a concrete representation of the world view underlying the language (captured in a *specification of the language metamodel*), with an explicit and formal representation of a conceptualization of that domain, which is termed here a *reference ontology*.

In philosophy, ontology is the most fundamental branch of metaphysics. It is a mature discipline, which has been systematically developed in western philosophy at least since Aristotle. The business of ontology "…is to study the most general features of reality" (Peirce, 1935), as opposed to the several specific scientific disciplines (e.g., physics, chemistry, biology), which deal only with entities that fall within their respective domain. However, there are many ontological principles that are utilized in scientific research, for instance, in the selection of concepts and hypothesis, in the axiomatic reconstruction of scientific theories, in the design of techniques, and in the evaluation of scientific results (Bunge, 1977, p.19). Thus, to quote the physicist and philosopher of science Mario Bunge: "every science presupposes some metaphysics".

In the beginning of the 20th century, the German philosopher Edmund Husserl coined the term Formal Ontology as an analogy to Formal Logic. Whilst Formal Logic deals with formal[1] logical structures (e.g., truth, validity, consistency) independently of their veracity, Formal Ontology deals with formal ontological structures (e.g., theory of parts, theory of wholes, types and instantiation, identity, dependence, unity), i.e., with formal aspects of objects irrespective of their particular nature. The unfolding of Formal Ontology as a philosophical discipline aims at developing a system of general categories and their ties, which can be used in the development of scientific theories and domain-specific common sense theories of reality.

---

[1]The adjective *Formal* here refers to its more ancient meaning, namely, referring only to *Form*, in the sense of independent of *Content*. The use of *formal* as synonym for *precise* or *mathematical* originates from the fact that mathematical theories are typically *Formal* in the first sense.

More recently, ontology has been applied in a multitude of areas in computer science. In many cases, however, the term is employed with a more liberal meaning and, instead of referring to a general (i.e., domain-independent) system of categories, it is also used to refer to specific theories about material domains (e.g., law, medicine, archeology, molecular biology, etc.), named *domain ontologies*. Thus, if in the philosophical sense, ontology is the *study of existence and modes of existence in a general sense*, in computer science, a domain ontology is the study of what exists in a given domain or universe of discourse.

The activity of constructing domain ontologies is known in the literature as *Ontological Engineering*. An ontological engineering process typically comprises activities such as: *Purpose Identification and Requirements Specification*, *Ontology Modeling*, *Ontology Codification*, *Reuse and Integration*, *Evaluation* and *Documentation* (see, for instance, Falbo & Guizzardi & Duarte, 2002; Gómez-Pérez & Fernández-López & Corcho, 2004). Here, we consider a domain ontology as a special type of conceptual specification and, hence, ontology modeling as a special type of conceptual modeling.

Therefore, in figure 1.1, if by a conceptualization we mean a conceptualization of a material domain, then by modeling language we mean a *domain-specific modeling language*. In contrast, if in figure 1.1 by a conceptualization we mean a formal (i.e., domain- independent) conceptualization, then by a modeling language we mean a *general conceptual modeling language* (or *ontology representation language*).

The design of domain-specific modeling languages is a current and important research topic (Kelly & Tolvanen, 2000; Tolvanen, Gray & Rossi, 2004; Bottoni & Minas, 2003) in conceptual modelling and, as we show on chapter 2, some results of this thesis also contribute to the area of domain-specific language evaluation and design. Nonetheless, the focus of this work is not on domain-specific languages and domain ontologies but, conversely, on general conceptual modeling languages and their underlying formal conceptualizations, if only because (as we demonstrate in chapters 2 and 3), the design of the former presupposes the existence of a suitable general conceptual modeling language. Thus, henceforth we simply use the term *conceptual modeling language* when referring to a general conceptual modeling language.

Conceptual (Ontology) Modeling is a fundamental discipline in computer science, playing an essential role in areas such as database and information systems design, software and domain engineering, design of knowledge-based systems, requirements engineering, information integration, semantic interoperability, natural language processing, enterprise modeling, among many others. In particular, domain ontologies have a central position in the so-called *Semantic Web* vision (Berners-Lee, Hendler, Lassila, 2001). In this context, web resources (information nodes and computational services) have their semantics informed by association with one or more domain ontologies. For example, the systems A and B in the pattern of figure 1.3 could correspond to two independently developed *semantically annotated web* services (McIlraith & Son & Zeng, 2001). Since web services can be considered as special kinds of telematics services (Ferreira Pires et al., 2004), the results developed throughout this thesis amount to a contribution to the general area of telematic services. However, more generally, the systems A and B in the pattern of figure 1.3 could also correspond to two interacting software agents, social organizations, or human stakeholders. Therefore, the results presented here contribute more broadly to all the areas aforementioned in computer science in which conceptual modeling play an essential role.

In summary, we defend in this thesis that the truthfulness to reality of a given system, as well as the semantic interoperability of concurrently developed systems, strongly depend on the availability of conceptual modeling languages that are able of making explicit and precise representations of the conceptualizations of their underlying subject domains. Therefore, two central research questions are: How can we define a suitable *formal conceptualization* (and consequently a *formal ontology*) that a conceptual modeling language should commit to? How can we (re)design a *conceptual modeling language* that conforms to this formal conceptualization (ontology)? These questions are answered throughout this thesis.

1.2 Motivation

Nowadays, many languages exist that are used for the purpose of creating representations of real-world conceptualizations. These languages are sometimes named *domain modeling languages*[2] (e.g., LINGO), *ontology representation languages* (e.g., OWL), *semantic data modeling languages* (e.g., ER), among other terms. We shall refer to these languages as conceptual modeling languages henceforth.

Although these languages are employed in practice for conceptual modeling, they are not designed with the specific purpose of being truthful to reality. For instance, LINGO (Falbo & Menezes & Rocha, 1998; Falbo & Guizzardi & Duarte, 2002) was designed with the specific objective of achieving a positive trade-off between expression power of the language and the ability to facilitate bridging the gap between the conceptual and

[2] The term *domain modeling language* is used in this sense to refer to domain-independent languages which can be used to create specifications of different material domain conceptualizations, not to refer to domain-specific modeling languages as previously discussed.

implementation levels. This preoccupation also seems to be present in Peter Chen's original proposal for ER diagrams (Chen, 1976). OWL (Horrocks & Patel-Schneider & van Harmelen, 2003) has been designed with the main purpose of achieving computational efficiency in an automatic reasoning process. Some other languages, such as Z (Spivey, 1988) and CC Technique (Dijkman & Ferreira Pires & Joosten, 2001), take advantage of the simplicity of the well-defined mathematical framework of set theory. Finally, some of the languages used nowadays for conceptual modeling were created for different purposes, the most notorious example being the UML (OMG, 2003c), which initially focused on software design.

As we show in this thesis, the worldviews underlying these languages (their ontological metamodels), cannot be considered as adequate conceptualizations of reality. As a consequence, they fall short in offering their users suitable sets of modeling concepts for constructing precise and explicitly characterized representations of their subject domains of interest.

We defend here that the focus of a conceptual modeling language should be on *representation adequacy* (i.e., truthfulness to reality and pragmatic efficiency). Conceptual modeling is *primarily* about *"describing some aspects of the physical and social world around us for purposes of understanding and communication"*, not systems design. Moreover, conceptual modeling languages should be highly-expressive, even at the cost of sacrificing computational efficiency and tractability. After all, although conceptual modeling can greatly benefit from efficient tool support in activities such as model manipulation and visualization, storage, syntactic verification and reasoning, among others, *"the descriptions that arise from conceptual modelling activities are intended to be used [primarily] by humans, not machines."*

Currently, there is no commonly agreed language for describing real-world phenomena in computer science. For this reason, in order to overcome the deficiencies of existing modeling languages for this purpose, a number of recent research efforts have investigated the use of Formal Ontological theories to evaluate and redesign these languages, as well as equip them with adequate real-world semantics. Examples include (Shanks & Tansley & Weber, 2003; Evermann & Wand, 2001b; Bodart et al., 2001; Opdahl & Henderson-Sellers, 2001; Green & Rosemann, 2000).

The approach proposed here differs from the ones mentioned above in two main characteristics: First, each of the approaches presented focus on specific sets of concepts. For example, the ontological analysis presented in (Opdahl & Henderson-Sellers, 2001) is targeted at *part-whole relations*, the one of (Bodart et al, 2001) is targeted at *properties*, and the one of (Evermann & Wand, 2001b) analyses *classes*, *class hierarchies* and *properties* (among other non-structural concepts, such as interaction). Our approach is broader in scope and, hence, can be considered in this sense an extension of these efforts. Consequently, it provides a comprehensive set of ontological theories, which covers all fundamental conceptual modeling concepts, and tackles a number of conceptual modeling problems that have not yet been satisfactorily addressed by any of the existing approaches in the conceptual modeling literature.

Second, the type of ontological investigation carried out here is different from the investigation in these other approaches. One characteristic common to all the efforts aforementioned is that they employ the same ontological theory, namely an ontology named BWW (Bunge-Wand-Weber) based on the original methaphysics proposed in (Bunge, 1977, 1979). Mario Bunge is a physicist and a philosopher of science and his theory is meant to serve as a foundation for specific scientific disciplines. As a consequence, it subscribes to an approach of ontological investigation that is committed to capture the intrinsic nature of the world in a way that is independent of conceptualizing agents and, consequently, an approach in which cognition and human language play a minor or non-existent role.

As we demonstrate in the development of this thesis, an ontology that can be used for providing foundations for conceptual modeling should be a philosophically well-founded one, but also one that aims at capturing the ontological distinctions underlying human cognition and common sense. Nonetheless, this ontology should not be regarded as less scientific, in the sense that the very existence of its constituting categories can be empirically uncovered by research in cognitive sciences (Keil, 1979; Xu & Carey, 1996; Mcnamara, 1986) in a manner that is analogous to the way philosophers of science have attempted to elicit the ontological commitments of the natural sciences.

In summary, the position defended here subscribes to Mylopoulos' dictum (Mylopoulos, 1992) that *"[t]he adequacy of a conceptual modelling notation rests on its contribution to the construction of models of reality that promote a common understanding of that reality among their human users."*

1.3 Objectives

In this thesis, we aim at contributing to the theory of conceptual modeling and ontology representation. Our main objective here is to provide ontological foundations for the most fundamental concepts in conceptual modeling. These foundations comprise a number of ontological theories, which are built on established work on philosophical ontology, cognitive psychology, philosophy of language and linguistics. Together these theories amount to a system of categories and formal relations known as a *foundational ontology* (Masolo et al., 2003a).

Besides philosophical and cognitive adequacy, we intend our foundational ontology to be precise. Therefore, we make use of some modal logics concepts to formally characterize the entities that constitute our ontology. In case the ontological distinctions proposed cannot be properly characterized by standard formal approaches, we have proposed some extensions to these standard formal approaches to accomplish the characterization required.

Once constructed, we have used this foundational ontology as a *reference model* prescribing the concepts that should be countenanced by a well-founded conceptual modeling language, and providing real-world semantics for the language constructs representing these concepts.

In the reference ontology proposed, we have focused on providing foundations for the most fundamental and widespread constructs for conceptual modeling, namely, types and type taxonomies, roles, attributes, attribute values and attribute value spaces, relationships, and part-whole relations.

Besides the theoretical work, we have addressed existing conceptual modeling problems, and contributed to the creation of sound engineering tools that can be used in the conceptual modeling practice. These have been realized in the form of ontological design patterns, capturing standard solutions to recurrent conceptual modeling problems, and methodological directives. However, more importantly, we have instantiated the approach defended here, by proposing a concrete conceptual modeling language that incorporates the foundations captured in our reference ontology.

The ontology proposed serves as a reference for designing new conceptual modeling language, but also for analyzing the ontological adequacy of existing ones. However, to conduct these activities in a principled manner, we have established a systematic relation between a modeling language and the ontology representing the real-world conceptualization of a given domain. Once this relationship has been precisely understood, we have analyzed and redesigned a specific modeling language, namely, the Unified Modeling Language (UML) (OMG, 2003c). The objective has been to propose an ontologically well-founded version of UML that can be used as an appropriate conceptual modeling language. The choice for UML lies on two main points: (i) the current status of UML as *de facto* standard modeling language; (ii) the growing interest in its adoption as a language for conceptual modelling and ontology representation (OMG, 2003a; Kogut, 2002). Because of these reasons, the re-designed version of UML is in itself an important research contribution of this thesis.

Finally, in order to demonstrate the suitability of the conceptual modeling language proposed we have developed a case study in a domain where we can exercise both (i) the capabilities of the language in precisely characterising the domain elements; (ii) the use of the language in supporting the semantic integration of different domain models. In particular, in (ii), we have shown the importance of a suitable conceptual modelling language in making explicit the ontological commitments of the conceptualizations underlying the individual models and, consequently, in helping to prevent false agreement in their integration.

In summary, the objectives of this thesis have been:

1. To establish a systematic relation between a *modeling language* and a *reference ontology*, and to propose a methodological approach to analyze and (re)design modeling languages to reinforce *representation adequacy* exploiting this relation;

2. To construct philosophical and cognitive *foundational ontology* for conceptual modeling and to formally characterize the elements constituting this ontology;

3. To demonstrate the usefulness of the ontological categories and theories that were proposed to address existing *conceptual modeling problems*;

4. To demonstrate the adequacy of the approach proposed in (1) and of the foundational ontology constructed in (2) by analyzing and redesigning an existing conceptual modeling language for representation adequacy;

5. To demonstrate the adequacy of the ontologically well-founded conceptual modeling language produced in (4) in the activity of improving the domain representation of existing conceptual specifications, and supporting their semantic integration.

1.4 Scope

The focus of this thesis is on general (i.e., domain independent) conceptual modeling languages. For this reason, we focus here on the construction of formal ontological theories instead of (domain-specific) material ones.

Our objective is to provide foundations for *structural* (i.e., static) aspects of conceptual modeling languages, as opposed to dynamic ones. This class of languages includes languages known as *data modeling frameworks*, *ontology representation languages*, *knowledge representation languages*, *semantic data modeling languages*, among others. To put it in simple terms, we restrict ourselves here to *objects*, the *types* they instantiate, the *roles* they play in certain contexts, their *constituent parts*, their *intrinsic and relational properties*, and the *structures in which their features are valued*, among other things. In contrast, we do not elaborate on *processes* and *events*. To put it in philosophical terms, the foundational ontology developed here is an *ontology of Endurants* (*continuants*) not one of *Perdurants* (*occurents*) (van Leeuwen, 1991; Masolo et al., 2003a). This is far from denying to the latter the status of ontological entities. Actually, in (Guizzardi & Wagner, 2005a), we elaborate on the role of an *ontology of perdurants* as an extension of the work presented in this thesis. In summary, the restriction of the discussion promoted here to the ontological category of endurants is merely a matter of scope.

The objective of this thesis is also to evaluate the suitability of languages to represent phenomena in a given domain. In terms of quality criteria for modeling languages, our scope is on *expressiveness* and *clarity*. Thus, it is not the objective here to discuss specific language technologies related to the definition of metamodel specifications, concrete syntax or formal semantics. Moreover, we do not discuss aspects related to systems design and, in particular, we do not address the impact on design choices of the modeling concepts proposed here. Finally, the target of our work is on conceptual modeling concepts and languages conceived for representation adequacy, aimed at being employed by human users in activities such as communication, domain understanding (learning) and analysis. Therefore, the study of properties such as computational efficiency and tractability of these languages fall outside the scope of this work.

1.5 Approach and Structure

The structure of this thesis reflects the successive elaboration of the objectives identified in section 1.4. The approach followed here to accomplish these objectives is detailed in the sequel.

(O1). <u>Objective 1:</u> To establish a systematic relation between a *modeling language* and a *reference ontology*, and to propose a methodological approach to analyze and (re)design modeling languages to reinforce *representation adequacy* exploiting this relation

This objective is accomplished in chapters 2 and 3 of this thesis. We start chapter 2 by discussing the various aspects that comprise a system of representations, or simply, a language. After briefly discussing the issues of (abstract and concrete) syntax, (formal and real-world) semantics and pragmatics, we concentrate on the definition of an evaluation framework that can be used to precisely evaluate the suitability of a language to represent phenomena according to a given real-world conceptualization. In our approach, this property can be systematically evaluated by comparing the level of homomorphism between a concrete representation of the world view underlying the language (captured in the specification of a *metamodel of the language*), and an explicit and formal representation of a conceptualization of that domain, or a *reference ontology*. The framework proposed comprises a number of properties (*lucidity, soundness, laconicity, completeness*) that must be reinforced for an isomorphism to take place between these two entities.

Although the focus of our work is on general conceptual modeling languages, the framework and the principles presented can be applied to the design of conceptual modeling languages irrespective of the generalization level to which they belong. In particular, they can also be used for the design of domain-specific modeling languages. In chapter 2, the approach presented is illustrated with a small case study in the design of a domain-specific visual modeling language for the domain of genealogy. The evaluation and redesign of a general conceptual modeling language is the main case study of this thesis, which is presented in chapter 8.

In chapter 3, we elaborate on some of the concepts of this framework by presenting a formal characterization of a *conceptualization* and its *intended models* (the models standing for *admissible state of affairs*), the *ontological commitment* of a language, and of the role of an *ontology* to approximate the valid specifications of a language to the intended models of its underlying conceptualization.

The main objective of chapter 3 is, however, to discuss the topic of ontologies both from philosophical and computer science points of view. We first give a historical perspective on ontology from a philosophical perspective, and discuss the importance of ontological investigations for science, in general, and for conceptual modeling, in particular. The formal characterization aforementioned is also used in this chapter to harmonize the original uses of ontology in philosophy with the several senses the term is employed in computer science. By doing this, we offer a precise definition of the meaning of the term, which is assumed for the remaining of this work.

(O2). Objective 2: To construct a philosophical and cognitive *foundational ontology* for conceptual modeling and to formally characterize the elements constituting this ontology.

(O3). Objective 3: To demonstrate the usefulness of the ontological categories and theories that are proposed to address existing conceptual modeling problems.

The accomplishment of these objectives constitutes the core of this thesis. The construction of the foundation ontology proposed here is organized in four complementary chapters in the following manner:

(a). In Chapter 4, we provide a theory for defining ontological distinctions on the category of conceptual modeling object universals, as well as constraints on the construction of *taxonomic structures* using these distinctions. By using a number of formally defined meta-properties, we can generate a *typology of universals*, which in turn can be used to give real-world semantics for important conceptual modeling concepts such as *types*, *roles*, *phases* and *mixins*. Besides providing unambiguous definition for these concepts, the elements of our theory function as a methodological support for helping the user of the language to decide how to represent elements that denote universal properties in a given domain. The usefulness of this approach is demonstrated in this chapter by showing how the theory can be used to evaluate and improve the conceptual quality of class hierarchies and concept taxonomies. Finally, in order to provide a suitable formal characterization of the ontological distinctions and postulates present in this theory, we present some extensions to a traditional system of modal logics;

(b). In Chapter 5, we concentrate on the topic of *part-whole relations*. First, we extend the insufficient axiomatization offered for these relations in present conceptual modeling languages, by considering a number of theories of parts from formal ontology in philosophy (Mereologies)(Simons, 1987). Thus, by building on the literature of meronymic[3] relations on linguistics and cognitive sciences, we extend the formal notion of parthood to a typology composed of four different conceptual part-whole relations. The elements in this typology are also characterized by additional formal meta-properties (e.g., essentiality, exclusiveness, separability, transitivity);

(c). In Chapter 6, we present the core of the foundational ontology proposed here, by addressing the categories of *attributes, attribute values and attribute value spaces, relationships* and *weak entities*. This fragment of our ontology is presented in a parsimonious theory, which is used to provide unambiguous real-world semantics for these concepts. In particular, this chapter offers a simple, precise and ontologically well-founded semantics for the problematic concept of relations, but also one that can accommodate more subtle linguistic distinctions. As it is demonstrated, this foundation for relations has a direct impact in

[3]**Meronym**: a word that names a part of a larger whole; *brim* and *crown* are meronyms of *hat*. The contrary idea is that of **Holonym**, i.e., a word that names a whole of which a given word is part. In this example, *hat* is a holonym for *brim* and *crown* (WordNet, 2005).

improving the representation of these entities in conceptual specifications. Additionally, it provides a principled basis for an ontological interpretation and for the specification of structured datatypes;

(d). In Chapter 7, we employ some of the results of Chapter 6 to fully describe the modeling concept of roles. The chapter also serves as an exemplification of the usefulness of the categories proposed in our foundational ontology. Firstly, by employing the categories and postulates of the theory of universals constructed in Chapter 4, we propose an ontological design pattern capturing a solution to a recurrent and much discussed problem in role modeling. Secondly, with some definitions offered in Chapter 6, we have investigated the link between some of the formal meta-properties defined for part-whole relations and those meta-properties by which roles are characterized. Thirdly, by borrowing some results from Chapters 4 and 6, we have managed to harmonize some different conceptions of roles used in the literature. Finally, by building on an existing theory of transitivity of linguistic functional parthood relations, and on some material from Chapter 6, we have proposed a number of visual patterns that can be used as methodological support for the identification of the scope of transitivity for the most common type of part-whole relations in conceptual modeling.

(O4). Objective 4: To demonstrate the adequacy of the approach proposed to fulfil (O1) and of the foundational ontology constructed to fulfil (O2) by analyzing and redesigning an existing conceptual modeling language for representation adequacy.

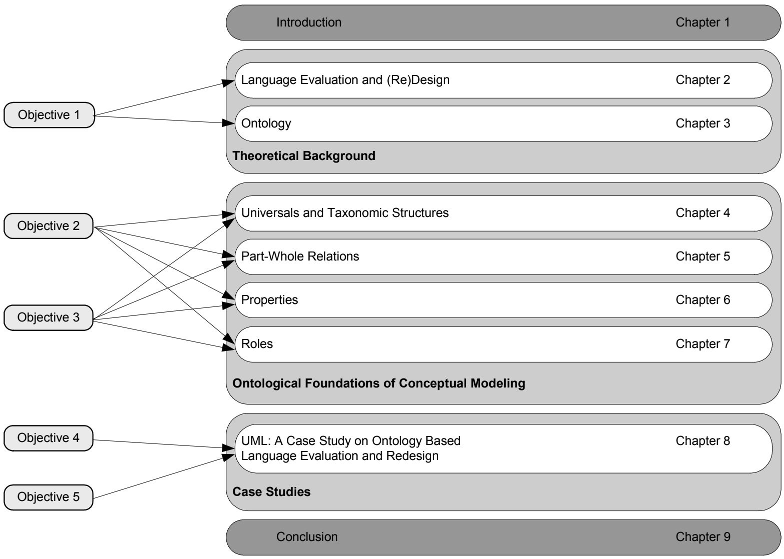
(O5). Objective 5: To demonstrate the adequacy of the ontologically well-founded conceptual modeling language produced to accomplish (O4) in the activity of improving the domain representation of existing conceptual specifications, and supporting their semantic integration.

In Chapter 8, we present the two major case studies of this thesis. As a first case study to exemplify the adequacy of the framework and foundation ontology proposed, we use the latter as a reference for analyzing the ontological appropriateness of the Unified Modeling Language (UML) for the purpose of conceptual modeling. Moreover, by employing the systematic evaluation method comprising the framework, we have identified a number of deficiencies and recommended modifications to the UML metamodel specification accordingly. As a result of this process, we have managed to produce a conceptually cleaner, semantically unambiguous and ontologically well-founded version of the language.

As an attempt to shield as much as possible the user of a conceptual modeling language from the complexity of the underlying ontological theory, we take the approach of (whenever possible) representing the ontological principles underlying a language in terms of syntactical constraints of this language. As a consequence, we manage to produce a modeling language whose grammatically valid specifications approximate as much as possible the intended models of its underlyimg conceptualization.

Finally, in Chapter 8, we also carry out a second case study, which uses the version of UML proposed in that chapter to analyze and integrate several semantic web ontologies in the scope of a context-aware service platform. This case study aims at demonstrating the adequacy of this version of UML as a conceptual modeling language and as a tool to support minimizing the false agreement problem previously discussed. Accordingly, it has also demonstrated the suitability of the ontological foundations underpinning this language for these purposes.
An overview of structure of this thesis is presented in figure 1.4 below.

Figure 1-4 Overview of the thesis structure relating the objectives of this thesis with the chapters in which they are accomplished



Chapter 2

Language Evaluation and Design

The objective of this chapter is to discuss the relation between a modeling language and a set of real-world phenomena that this language is supposed to represent. By exploring this relation, we propose a systematic way to design and evaluate *modeling languages* with the objective of reinforcing truthfulness to the corresponding domain in reality and conceptual clarity of the specifications produced using those languages.

In section 2.1, we discuss the various aspects that comprise a system of representations, or simply, a language. In that background section, we briefly discuss the issues of (abstract and concrete) syntax, (formal and real-world) semantics and pragmatics.

In section 2.2, we concentrate on the definition of an evaluation framework that can be used to systematically assess the *real-world semantics* of an artificial modeling language, i.e. how suitable a modeling language is to model phenomena according to a given real-world conceptualization.

Since conceptualizations are abstract entities, in order to precisely conduct the evaluation framework advocated in section 2.2, a concrete representation of a conceptualization must be made available. In this thesis, a domain conceptualization is expressed in terms of a shared conceptual specification of the domain, named here a *domain ontology*. In section 2.3, we discuss the role that ontologies of material domains (such as law, medicine, archaeology, genetics) play in formalizing the semantic domain of domain-specific languages and informing properties that can be exploited in the design of efficient visual pragmatics for these languages. In particular, we illustrate our approach with an example in the domain of genealogy. Since one of our main objectives in this thesis is to evaluate and improve the quality of *general conceptual modeling languages*, in section 2.3 we also discuss the characteristics that are required for a domain independent meta-conceptualization that to which a general conceptual modeling language should commit.

In section 2.4, we discuss the role that explicit represented domain ontologies can play in semantically interoperating (e.g., integrating, comparing, translating) models produced in languages that diverge in syntax and semantics but whose underlying real-world conceptualizations overlap. This topic, albeit related, falls outside the central concern of this thesis. For this reason, it is only briefly discussed.

In section 2.5, we present some final considerations.

2.1 Elements of Language Design

According to (Morris, 1938) a language comprises three parts: ***syntax***, ***semantics*** and ***pragmatics***. Syntax is devoted to *"the formal relation of signs to one another"*, semantics to *"the relation of signs to real world entities they represent"* and pragmatics to *"the relation of signs to (human) interpreters"*. In the following subsections we elaborate upon these three definitions.

2.1.1 Syntax

In (Harel & Rumpe, 2000), the authors discuss the distinction between the purist notion of information and its syntactical representation as data, which is the medium used to communicate and store information. Data is essential to give a concrete and persistent status to some information, but it is in itself, however, vacuous in terms of meaning. Thus, to extract the information behind a piece of data, an ***interpretation*** is necessary that assigns meaning to it. The same piece of information may be represented as different data (e.g., *"December 31st, 2002"* and *"the last day of the year 2002"* refer to the same entity). Likewise, the same piece of data may serve as a ***representation*** for different things for different people at different points in time. The difference between data and information resembles the difference between a language syntax and semantics, respectively.

In order to communicate, agents must agree on a common communication language. This fixes the sets of signs that can be exchanged (syntax) and how these signs can be combined in order to form valid expressions in the language (syntactical rules). In sentential languages, the syntax is first defined in terms of an alphabet (set of characters) that can be grouped into valid sequences forming words. This is called a lexical layer and it is typically defined using regular expressions. Words can be grouped into sentences according to precisely defined rules defined in a context-free grammar, resulting in an abstract syntax tree. Finally, these sentences are constrained by given context conditions. The list of valid words of a language is called its *vocabulary.*

In diagrammatic (graphical) languages, conversely, the vocabulary of the language is not defined in terms of linear sequence of characters but in terms of pictorial signs. The set of available graphic modeling primitives forms the lexical layer (the ***concrete syntax***) and the language's ***abstract syntax*** is typically defined in terms of an abstract visual graph (Erwig, 1999) or a metamodel specification. The latter alternative applies to the OMG's Model Driven Architecture (MDA) initiative (Bézivin & Gerbé, 2001) and it is the one that is considered in this thesis. Finally, the language metamodel specification is enriched by context conditions given in some constraint description language, such as, OCL or first-order logic (FOL). In either case, context conditions are intended to constrain the language syntax by defining the set of correct (well-formed) sentences of the language. Some of these constraints are motivated by semantic considerations (laws of the domain being modeled) while others can be motivated by pragmatic issues (discussed in section 2.1.3).

In summary, in the context of this thesis, a *language metamodel specification* (defining the abstract syntax of a language) defines the rules for the creation of well-formed models in that language. In other words, it defines the set of grammatically correct models that can be constructed using that language. In contrast, the vocabulary, or concrete syntax of that language, provides a concrete representational system for expressing the elements of that metamodel.

2.1.2 Semantics

Besides agreeing on a common vocabulary, participants in a communication process must share the same meaning for the syntactical constructs being communicated, i.e., they must interpret in a compatible way the expressions of the communication language being used. Without a proper way to extract information, the data being exchanged is worthless. Semantics deals with the meaning of signs and symbols or, in other words, with the information behind a piece of data.

Although most authors agree that semantics concern the relation between the expressions of a language and their meanings, opinions diverge when explaining the nature of this relation. As a consequence, there is a long-standing philosophical dispute concerning the meaning of "meaning" (Gärdenfors, 2000). A major divergence on the ontological status of the semantic relation concerns whether semantics is *referential* or not. In referential semantics, there are some kinds of objects (things in the world or mental entities) that are the meanings of linguistic expressions. Within the area of philosophy of language there is also a *functionalist* tradition of meaning which is non-referential. According to (Gärdenfors, ibid.), the most well-known proponent of this view is the later Wittgenstein, who defended the view that is often summarized by the slogan *"meaning is use"*, i.e., that the (linguistic) meaning of a linguistic expression is its (canonical, proper) communicative function, i.e., its potential contribution to the communicative function of utterances of which it forms part. To put it simply, in a functionalist approach the meaning of a word is not given by a reference to an extra-linguistic entity, but by a set of circumstances and intentions in which the word is used.

From now on we concentrate on the referential approach, due to its suitability to the enterprise pursued in this thesis and its compatibility with the semantic tradition of fact-based conceptual modeling languages (Hirschheim, Klein & Lyytinen, 1995), which are the focus of this work.

In general terms, a semantic definition for a language $\mathcal{L}$ consists of two parts: (i) a semantic domain D; (ii) a semantic mapping (or interpretation function) from the syntax to the semantic domain, formally I: $\mathcal{L} \rightarrow \mathcal{D}$ (Harel & Rumpe, 2000). The semantic mapping tells us about the meaning of each of the language's expressions in terms of an element of the semantic domain.

In computer science, the relation between the semantic domain and the domain conceptualization can be interpreted according to two different stances to semantics. In AI research, semantics denote *"some form of correspondence specified between a surrogate and its intended referent in the world; the correspondence is the semantics for the representation"* (Davis & Shrobe & Szolovits, 1993). In other words, semantics is a mapping (interpretation) from the language vocabulary to concepts that stand for entities in the real world. Conversely, approaches with heritage in other traditions of programming consider semantics without a commitment to a specific real-world conceptualization. In these approaches, the term "semantics" is used to denote rules for program compilation or automated interpretation. According to these approaches, the semantic domain is a mathematical domain not necessarily related to a real-world conceptualization. In the AI approach, conversely, the semantic domain is typically a *material domain*, such as, engineering, business, medicine or telematics. In (Ferreira Pires, 1994; Vissers & van Sinderen & Ferreira Pires, 1993), these are called *formal* and *architectural semantics*, respectively. From now on, we use the term *domain conceptualization* when referring to a real-world conceptualization and the term *mathematical conceptualization* when referring to purely mathematical one. Architectural semantics, under the term ***real-world semantics*** (Partridge, 2002; Sheth & Kashyap, 1992; Vermeer, 1997; Vermeer & Apers, 1996), have been perceived as fundamental for semantically interoperability and semantic integration of information sources. In (Partridge, 2002), for instance, it is claimed that: *"Underlying the variety of forms of integrating data and applications there is a common semantic task - what can be called the 'semantic matching'. There is a reasonably clear recognition* *that the analysis stage of this task needs to focus on identifying the entities that the data describes, i.e. the 'real-world semantics'"*.

In (Ferreira Pires, 1994; Vissers & van Sinderen & Ferreira Pires, 1993), it is argued that designers should concentrate on the elaboration of models, using the modeling language merely as a vehicle for the representation of design characteristics. A modeling language can only be useful for its community of users, if its vocabulary, syntax, semantic and pragmatics, are defined based on the needs of this community for the elaboration of specifications. This view, supported here, emphasizes the precedence of real-world concepts over mathematical concepts in the design and evaluation of modeling languages or, in other words, the precedence of real-world semantics over purely mathematical semantics.

In section 2.2 we discuss in detail the relation between a language (as a set of interrelated modeling primitives) and a domain conceptualization, in the definition of the real-world semantics of this language.

2.1.3 Pragmatics

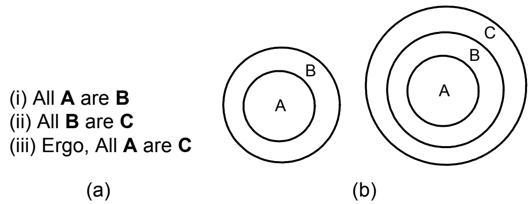
Following Morris' definition (Morris, 1938), pragmatics concerns *"the relation of signs to (human) interpreters"*. Human users have contact with a modeling language (and with specifications produced in this language) via the language's concrete syntax.

A system of concrete syntax may be considered to be a collection of objects and some relations between these objects. The type (visual or otherwise) of a particular representation, and more generally of a language, is determined by the characteristics of the *symbols* used to express these objects and relations (Gurr, 1999). Throughout this discussion we use the term symbol in a broad sense. Thus, our symbol definition includes visual features (e.g., morphological, geometric, spatial and topological relations) used by diagrammatic languages to represent objects and relations.

In sentential languages, there is a clear separation between vocabulary, syntax and semantics. The syntactic rules which permit construction of sentences, may be completely independent of the chosen vocabulary, and may be clearly distinguished from a definition of semantics. For example, let P be a propositional logic whose vocabulary consists of the propositions p and q, and the symbols $\wedge$ and $\neg$ representing the logical connectives 'and' and 'not', respectively. The syntactic and semantic rules for P tell us, respectively, how to construct and interpret formulas using this vocabulary. However, we may substitute the symbols $\{p,q, \wedge,\neg\}$ for $\{X,Y,\&,\sim\}$ throughout P to produce a logic which is effectively equivalent to P. Alternatively, we could retain the vocabulary and syntax of P, while altering the semantics to produce a vastly different logic (ibid.).

The same does not hold for graphical languages, in which vocabulary, syntax and semantics are not clearly separable. For example, a graphical vocabulary may include shapes such as circles, squares, arcs and arrows, all of differing sizes and colors. These objects often fall naturally into a hierarchical typing which almost certainly constrains the syntax and, furthermore, informs about the semantics of the system. Likewise, spatial relations, such as inclusion, are part of the vocabulary but clearly constrain the construction of potential diagrams and are likely to be mapped onto semantic relations with similar logical properties (ibid.). This idea is illustrated by Figure 2.1 below, in which two different languages are used to express logical syllogisms. The sentential language of (Figure 2.1.a) and the visual language of (Figure 2.1.b) are semantically equivalent. Despite that, the inference step that culminates with conclusion (iii) is performed in a much more straightforward way in the language of Euler's circles (figure 2.1.b). This classic example shows how semantic information can be directly captured in a visual symbol. Here a sequence of valid operations is performed which cause some consequence to become manifest in a diagram, where that consequence is not explicitly insisted upon by the operations. This is because the partial order properties of the set inclusion relation are expressed via the similarly transitive, irreflexive and asymmetric visual properties of proper spatial inclusion in the plane, i.e. the representing relation has the same semantic properties as the represented relation. This argument is given a formal account by (Shimojima, 1996) where immediate inferences like in (Figure 2.1.b) are termed *"free-rides"*.

Figure 2-1 Logical Syllogism represented in a sentential language (a) and in the visual language of Euler's Circles (b)



In sentential languages, the relationships between symbols are necessarily captured in terms of the concatenation relation, which must then be interpreted by some intermediate abstract syntax (Gurr, 1999). In visual languages, intrinsic properties of the representation system can be systematically used to directly correspond to properties in the represented domain. Gurr uses the term *directness* to denote the correspondence between properties of representations and the properties of which they represent, and the term *systematicity* to denote the systematic application of directness in the design of systems of concrete syntax.

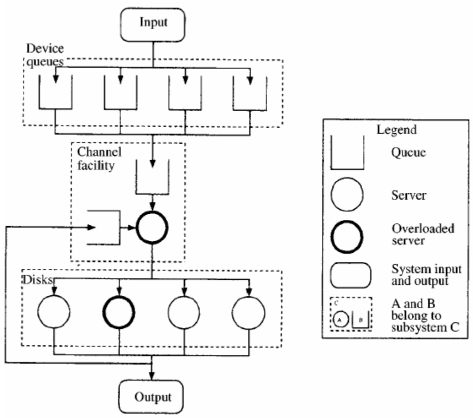
A notion of key importance to the discussion promoted here is the one of *implicature* proposed by the philosopher of language H. P. Grice (Grice, 1975, 1978). An implicature is anything that is inferred from an utterance but that is not a condition for the truth of the utterance. Grice distinguishes between the meaning of a sentence and what is implied by it. For instance, the utterance "*John has 14 children*" commonly implicates "*John has precisely 14 children*", even though it would be compatible with John having 20 children. Likewise, the sentences "*You should work as well as study*" and "*You should study as well as work*", despite of perhaps being semantically equivalent, can implicate different things. A conversational implicature is founded on the assumption that conversational participants will adhere to, what Grice names, a *cooperative principle.* According to Grice, the cooperative principle governs informative conversational discourse and is based upon the assumption that conversational participants adhere to certain rules, named *conversational maxims,* and, as consequence, that they will construct and recognize utterances as carrying intended implied information and, avoiding unwanted implications. Conversational maxims state that a speaker is assumed to make contributions in a dialogue which are *relevant*, *clear*, *unambiguous*, *brief*, *not overly informative* and *true according to the speaker's knowledge*. When a statement breaches any one of these maxims, additional meaning is inferred. To illustrate this idea the following dialogue is presented:

A. How are you doing?
B. Still alive.

In this example, the conversational maxim of relevance is broken. This leads to the interpretation *by implicature* – in addition to the obvious interpretation that the person is alive – that the person is not in good health or is experiencing some difficulties.

The relevance of notion such as *systematicity*, *implicatures*, and *conversational maxims* to the design of (visual) modeling languages can be motivated as follows. When correctly employed, systematicity can result in major increases in the effectiveness of the diagram for performing specific tasks. For instance, in (Hahn & Kim, 1999), the authors show how three semantically equivalent languages, namely, UML activity, sequence and collaboration diagrams can differ drastically in terms of effectiveness when employed for model integration. In contrast, whenever misused or neglected, implicatures can communicate incorrect information and induce the user to make incorrect inferences about the semantics of the domain. To illustrate this idea, we show in figure 2.2, 2.3 and 2.4 three versions of the same diagram. This example is cited by (Gurr, 1999) and has been taken from (Marks & Reiter, 1990). Figure 2.2 represents a computer's disk subsystem. The visual concrete syntax also expresses relations and intrinsic properties of represented domain entities via either icons or other visual features such as the use of spatial inclusion within a box drawn with a dotted line denoting membership of a subsystem. In figure 2.3, the inherent ordering of graphical symbols used to represent the nodes can be incorrectly interpreted that all nodes in the represented network fall into a single conceptual category, in which they are similarly ordered. In figure 2.4 we can find several examples of directly inferred incorrect information: (i) one of the device queues in the upper section of the diagram has been laid out irregularly which can be interpreted that this queue must be different than the others somehow; (ii) in the lower section of the diagram the disk-symbols are organized as two separated groups, wrongly implicating that this division reflects some grouping in the domain; (iii) the line-width used for the lower queue in the channel facility differs from the one used for all other queues in the diagram; (iv) a different font has been used for the channel facility's text label, wrongly implying that it must have different subsystem status.

Figure 2-2
Representation of a computer's disk subsystem



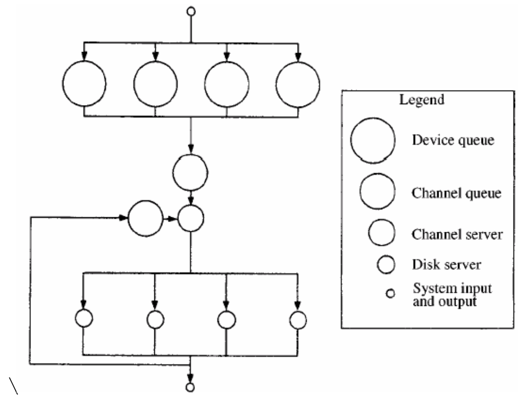


Figure 2-3 Semantically equivalent alternative representation of the model depicted in figure 2-2

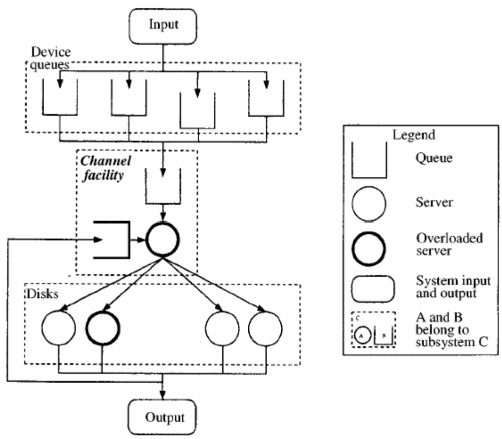


Figure 2-4 Semantically equivalent alternative representation of the model depicted in figure 2-2

All three diagrams of Figures 2.2, 2.3 and 2.4 faithfully portray all the features of the assumed network model, i.e., diagrams 2.3 and 2.4 cannot be considered misleading through purely semantic considerations. The problematic implications of these diagrams do not follow from the incorrect use of the symbols but rather by various aspects of its semantics which are not explicitly specified, but that can be misleading (Gurr, 1999).

While examples of Figures 2.3 and 2.4 illustrate the potential traps of ignoring pragmatic aspects of diagrams, other research has shown that the efficacy of diagrams for communicating information can be increased by the correct usage of such aspects. Studies by Petre and Green (Petre & Green, 1992) of engineers using CAD systems for designing computer circuits, demonstrated that the most significant difference between novices and experts is in the use of layout to capture domain information. In such circuit diagrams, the layout of components is not specified as being semantically significant. Nevertheless, experienced designers exploit layout to carry important information, e.g. by grouping together components that are functionally related. By contrast, certain diagrams produced by novices were considered poor because they either failed to use layout or, in particularly 'awful' examples, were especially confusing through the misuse of common layout conventions informally adopted by experienced engineers (Gurr, 1999).

2.1.4 Real-World Semantics

One of the main success factors behind the use of a modeling language is its ability to provide to its target users a set of modeling primitives that can directly express relevant domain abstractions. Domain abstractions are constructed in terms of concepts, i.e., abstract representations of certain aspects of entities that exist in a given domain that we name here a *domain conceptualization.* An abstraction of a certain *state of affairs* expressed in terms of a set of domain concepts, i.e., according to a certain conceptualization, is termed a *model* in this work (Ferreira Pires, 1994; van Sinderen, 1995). Take as an example the domain of genealogical relations in reality. A certain conceptualization of this domain can be constructed by considering concepts such as *Person*, *Man*, *Woman*, *Father*, *Mother*, *Offspring*, *being the father of*, *being the mother of,* among others. By using these concepts, we can articulate a conceptual model of certain facts in reality such as, for instance, that a man named John is the father of another man named Paul.

Conceptualizations and Models are abstract entities that only exist in the mind of the user or a community of users of a language. In order to be documented, communicated and analyzed they must be captured, i.e. represented in terms of some concrete artifact. This implies that a language is necessary for representing them in a concise, complete and unambiguous way.

Figure 2.5 represents the relation between a language, a conceptualization and the portion of reality that this conceptualization abstracts. This picture depicts the well-known *Ullmann's triangle* (Ullmann, 1972). This triangle derives from that of Ogden and Richards (Ogden & Richards, 1923) and from Ferdinand de Saussure (de Saussure, 1986), on whose theories practically the whole modern science of language is based.

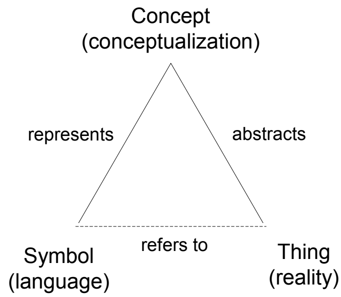


Figure 2-5 Ullmann's Triangle: the relations between a thing in reality, its conceptualization and a symbolic representation of this conceptualization

The *represents* relation concerns the definition of language $\mathcal{L}$ real-world semantics. The dotted line between language and reality in this figure highlights the fact that the relation between language and reality is always intermediated by a certain conceptualization (Baldinger, 1980). This relation is elaborated in Figure 2.6 that depicts the distinction between a model and its representation, and their relationship with the conceptualization and representation language. In the scope of this work the representation of a model in terms of a representation language $\mathcal{L}$ is called a *model specification* (simply *specification*, or *representation*) and the language $\mathcal{L}$ used for its creation is called a *modeling* (or *specification*) *language*.

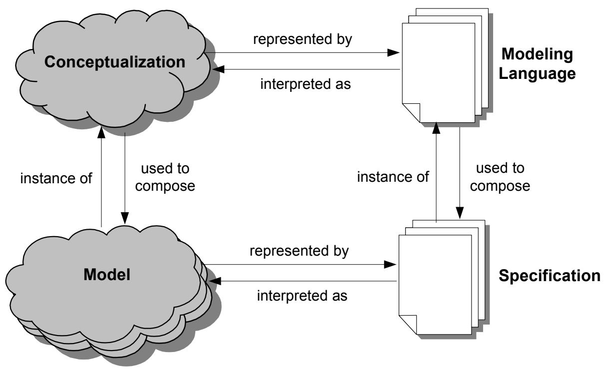


Figure 2-6 Relations between conceptualization, Model, Modeling Language and Specification

As previously mentioned, we defend the precedence of real-world concepts over formal concepts and implementational issues in the design of conceptual modeling languages. In particular, this thesis is not concerned with specific aspects of language technology such as formal syntax and semantics but with the so-called *domain appropriateness* and *comprehensibility appropriateness* of a given language (Krogstie, 2000; Halpin, 1998). In order for a specification $\mathcal{S}$ to faithfully represent a model $\mathcal{M}$, the modelling primitives of the language $\mathcal{L}$ used to produce $\mathcal{S}$ should faithfully represent the domain conceptualization $\mathcal{C}$ used to articulate the represented model

$\mathcal{M}$. The *domain Appropriateness* of language is a measure of the suitability of a language to model phenomena in a given domain, or in other words, of its truthfulness of a language to a given domain reality. On a different aspect, different languages and specifications have different measures of pragmatic adequacy. *Comprehensibility appropriateness* refers to how easy is for a user a given language to recognize what that language's constructs mean in terms of domain concepts and, how easy is to understand, communicate and reason with the specifications produced in that language.

The measures of these two quality criteria for a given language and domain are aspects of the *represents* relation depicted in figure 2.6. In the following section we elaborate on these aspects and present a framework to systematically evaluate the domain and comprehensibility appropriateness in accordance with a domain conceptualization. The *abstract* relation in figure 2.5 is discussed in chapter 3.

2.2 A Framework for Language Evaluation

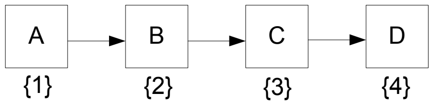
The purpose of the current chapter is to discuss the design and evaluation of artificial modeling languages for capturing phenomena in a given material domain according to a conceptualization of this domain. Before we aim at this target at a language level, i.e., at a level of a system of representations, we start discussing the simpler relation between a particular specification and a particular abstraction of a portion of reality, i.e., a particular model.

In (Gurr, 1998, 1999), the author presents a framework to formally evaluate the relation between the properties of a representation system and the properties of the domain entities they represent. According to him, representations are more or less effective depending on the level of homomorphism between the algebras used to represent what he terms the *representing* and the *represented* world, which correspond to the *specification* and *model* in our vocabulary, respectively.

Isomorphism is defined in mathematics as follows. Let *A* and *B* be two algebras, we say that a mapping between *A* and *B* is homomorphic iff the structure of relations over the elements of *A* is preserved by the relations which hold over the corresponding elements in *B*. However, if a homomorphism exists from *A* to *B* it does not mean that they are identical. It may be, for instance, that there are elements in *B* which are not mapped onto any element in *A*. Alternatively there can be individual elements in *B* which are mapped by to more than one element of *A*. If there is a homomorphic mapping between *A* and *B* and every element of *B* is mapped to by a unique element of *A* then the two algebras are identical and are said to be isomorphic (Gurr, 1999).

For example, consider the diagram depicted in figure 2.7 as a representation of four integers (e.g., from 1 to 4) and their ordering, where integers are represented by squares labeled with capital letters and the less-than relation is represented by the transitive closure of *is-arrow-connected.* In this case we may use the algebras ({1, 2, 3, 4},{<}) and ({*A*, *B,* *C,* *D*},{transitive closure of *is-arrow-connected*}) to describe the model and specification (diagram), respectively. One possible representational mapping between these entities maps 1-4 to *A*-*D*, respectively, and < to the transitive closure of *is-arrow-connected.* The individuals in the model which are represented by each of the symbols A-D are shown between brackets in the picture 2.7. This kind of semantic correspondence (that also appears in figure 2.9, 2.11, 2.13, 2.14 and 2.16) is not part of the language itself and is shown in the pictures with the only purpose of enhancing clarity of our explanations.

Figure 2-7 A representation for the numbers domain



Gurr argues at length that the stronger the match between a model and its representing diagram, the easier is to reason with the latter. The easiest case is when these matches are isomorphisms. The implication of this for the human agent who interprets the diagram is that his interpretation correlates precisely and uniquely with an abstraction being represented. By contrast, where the correlation is not an isomorphism then there may potentially be a number of different models which would match the interpretation.

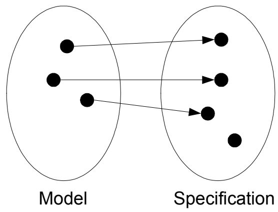
The evaluation framework proposed by Gurr focuses on evaluating the match between individual diagrams and the state of affairs they represent. In (Wand & Weber, 1989, 1990; Weber, 1997), another framework is defined for evaluating *expressiveness* and *clarity* of modeling grammars, i.e., with the focus on the system of representations as a whole. In our work, these two proposals are merged in one single evaluation framework. We focus our evaluation on the level of the system of representations. Nevertheless, as it will be shown in the following subsections, by considering desirable properties of the mapping of individual diagrams onto what they represent, we are able to account for desirable properties of the modeling languages used to produce these diagrams, extending in this way Wand & Weber's original proposal.

In (Gurr, 1999), four properties are defined, which are required to hold for a homomorphic correlation to be an isomorphism: *lucidity*, *soundness*, *laconicity* and *completeness*. These properties are discussed as follows.

2.2.1 Lucidity and Construct Overload

A specification $\mathcal{S}$ is called ***lucid*** with respect to (w.r.t.) a model $\mathcal{M}$ if a (representation) mapping from $\mathcal{M}$ to $\mathcal{S}$ is *injective*. A mapping between $\mathcal{M}$ and $\mathcal{S}$ is injective iff every entity in the specification $\mathcal{S}$ represents ***at most*** one (although perhaps none) entity of the model $\mathcal{M}$. An example of an injective mapping is depicted in figure 2.8.

Figure 2-8 Example of a Lucid representation mapping from Model to Specification



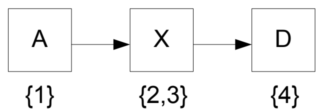
The notion of lucidity at the level of individual diagrams is strongly related to the notion of *ontological clarity* at the language level as discussed in (Weber, 1997; Wand & Wang, 1996; Wand & Weber, 1989). In (Weber, 1997), the author states that the ontological clarity of a modeling grammar is undermined by what they call ***construct overload***: *"construct overload occurs when a single grammatical construct can stand for two or more ontological constructs, The grammatical construct is overloaded because it is being used to do more than one job."*

The notions of *lucidity* and *ontological clarity* albeit related are not identical. A construct can be overloaded in the language level, i.e. it can be used to represent different concepts, but every manifestation of this construct in individual specifications is used to represent only one of the possible concepts. An example of construct overload w.r.t. the conceptualization proposed in chapters 4 to 7 occurs in the *relation* construct in the *relational data model* (Codd, 1970), which is used both to represent *object types* (e.g. *Person*, *Car*, *Student*) and types whose instances are *mutual properties* shared by individual objects (e.g., *enrolment*, *marriage*). However, it is not the case that in a particular model the relation construct is used to represent a type whose instances are both objects and mutual properties. Figure 2.9 exemplifies a non-lucid representation. In this case, the construct X is used to represent two entities of the model, namely the numbers 2 and 3. In this case, although the representation system does not have a case of construct overload (since labeled boxes only represent numbers and arcs only represent the *less-than* relation between numbers) the resulting specification is non-lucid. In summary, the absence of construct overload in a language does not directly prevent the construction of non-lucid representations in this language. Additionally, construct overload does not entail non-lucidity.

Nevertheless, non-lucidity can also be manifested at a language level. We say that a language (system of representation) is non-lucid according to a conceptualization if there is a construct of the language which is non-lucid, i.e., a construct that when used in a specification of a model (instantiation of this conceptualization) stands for more than one entity of the represented model. As demonstrated in chapter 6, the UML construct of an association class can be seen as a non-lucid construct (according to the conceptualization presented in that chapter) since it represents simultaneously a *mutual property* shared by a multitude of entities (e.g., *marriage*) and a *relation*[4] (e.g., *being-the-husband-of, being-the-wife-of, being-married-to*) induced by this property. Non-lucidity at the language level can be considered as a special case of construct overload that does entail non-lucidity at the level of individual specifications.

Construct overload is considered an undesirable property of a modeling language since it causes ambiguity and, hence, undermines clarity. When a construct overload exists, users have to bring additional knowledge not contained in the specification to understand the phenomena which are being represented. For instance, in the relational data model example given above, users have to examine the relations to determine whether an object type or a mutual property type is being represented. A non-lucid representation language entails non-lucid representations which clearly violate the Gricean conversational maxim that requires contributions to be neither ambiguous nor obscure. In summary, a modeling language should not contain construct overload and every instance of a modeling construct of this language should represent only one individual of the represented domain abstraction.

Figure 2-9 Example of a Non-Lucid diagram



2.2.2 Soundness and Construct Excess

A specification $\mathcal{S}$ is called ***sound*** w.r.t. a model $\mathcal{M}$ if a (representation) mapping from $\mathcal{M}$ to $\mathcal{S}$ is *surjective*. A representation mapping from $\mathcal{M}$ to $\mathcal{S}$ is surjective iff the corresponding interpretation mapping from $\mathcal{S}$ and $\mathcal{M}$ is total, i.e. iff every entity in the specification $\mathcal{S}$ represents ***at least*** one entity

[4] The notions of object type, mutual property (relator type) and relation among others are defined and discussed in depth in chapter 6 of this thesis.

of the model $\mathcal{M}$ (although perhaps several). An example of a surjective representation mapping is depicted in figure 2.10.

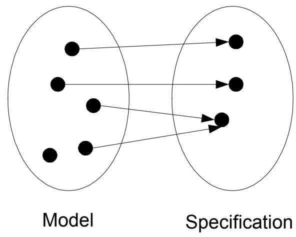


Figure 2-10 Example of a Sound representation mapping from Model to Specification

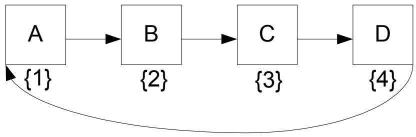
An example of an unsound diagram is illustrated in figure 2.11. The arc connecting the labeled boxes D and A does not stand for any relation in the represented world.

Unsoundness at the level of individual specifications is strongly related to unsoundness at language level, a property that is termed ***construct excess*** by Weber: *"construct excess occurs when a grammatical construct does not map onto an ontological construct"* (Weber, 1997).

In analyzing information system grammars in terms of an ontological theory, Weber advocates that the common notion of an *optional attribute* can be considered a case of construct excess. He claims that according to the theory used, there is no such a thing as an optional property. For instance, it is not the case that a person has the *"optional property"* of having children but, instead that there is a subtype of Person, say Parent, whose instances all have this property. Although construct excess can result in the creation of unsound specifications, soundness at the language level does not prohibit the creation of unsound specifications. For example, there is no construct excess in the language used to produce the specification of figure 2.11.

An unsound diagram violates the Gricean cooperative principle since any represented construct will be assumed to be meaningful by users of the language. Since no mapping is defined for the exceeding construct, its meaning becomes uncertain, hence, undermining the clarity of the specification. According to (Weber, 1997), users of a modeling language must be able to make a clear link between a modeling construct and its interpretation in terms of domain concepts. Otherwise, they will be unable to articulate precisely the meaning of the specifications they generate using the language. Therefore, a modeling language should not contain construct excess and every instance of its modeling constructs must represent an individual in the domain.

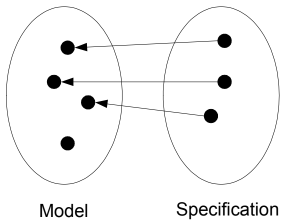
Figure 2-11 Example of an Unsound diagram



2.2.3 Laconicity and Construct Redundancy

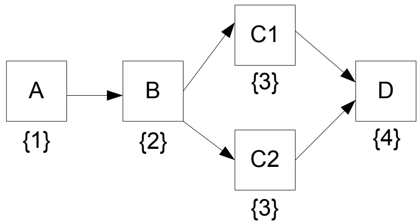
A specification $\mathcal{S}$ is called ***laconic*** w.r.t. a model $\mathcal{M}$ if the interpretation mapping from $\mathcal{S}$ to $\mathcal{M}$ is *injective*, i.e. iff every entity in the model $\mathcal{M}$ is represented by ***at most*** one (although perhaps none) entity in the representation $\mathcal{S}$. An example of an injective interpretation mapping is depicted in figure 2.12. The notion of *laconicity* in the level of individual specifications is related to the notion of ***construct redundancy*** in the language level in (Weber, 1997): *"construct redundancy occurs when more than one grammatical construct can be used to represent the same ontological construct."*

Figure 2-12 Example of a Laconic Interpretation mapping from Specification to Model



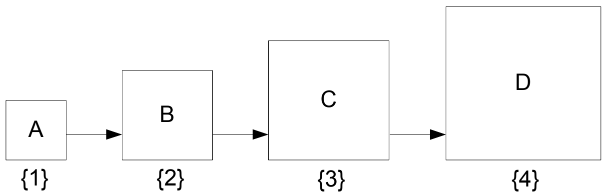
Once again, despite of being related, laconicity and construct excess are two different (even opposite) notions. On one hand, construct redundancy does not entail non-laconicity. For example, a language can have two different constructs to represent the same concept. However, in every situation the construct is used in particular specifications it only represents a single domain element. On the other hand, the lack of construct redundancy in a language does not prevent the creation of non-laconic specifications in that language. An example of a non-laconic diagram is illustrated in figure 2.13. In this picture, the same domain entity (the number 3) is represented by two different elements (C_1 and C_2) although the representation language used does not contain construct redundancy.

Figure 2-13 Example of a Non-Laconic diagram



Non-laconicity can also be manifested at the language level. We say that a language is non-laconic if it has a non-laconic modeling construct, i.e. a construct that when used in a specification of a model causes an entity of this model to be represented more than once. For instance, take a version of the labeled boxes language used so far and let the less-than relation between numbers be represented both as the *transitive closure of the is-arrow-connected* and by the *is-smaller-than* relation between labeled boxes. All specifications using this representation (e.g. figure 2.14) are deemed non-laconic. Non-laconicity at the language level can be considered as a special case of construct redundancy that does entail non-laconicity at the level of individual diagrams.

Figure 2-14 Example of a Non-Laconic diagram generated by a Non-Laconic language



In (Weber, 1997), the authors claim that construct redundancy *"adds unnecessarily to the complexity of the modeling language"* and that *"unless users have in-depth knowledge of the grammar, they may be confused by the redundant construct. They might assume for example that the construct somehow stands for some other type of phenomenon."* Therefore, construct redundancy can also be considered to undermine representation clarity. Non-laconicity also violates the Gricean principle, since redundant representations can be interpreted as standing for a different domain element. In summary, a modeling language should not contain construct redundancy, and elements in the represented domain should be represented by at most one instance of the language modeling constructs.

2.2.4 Completeness

A specification $\mathcal{S}$ is called *complete* w.r.t. a model $\mathcal{M}$ if an interpretation mapping from $\mathcal{S}$ to $\mathcal{M}$ is *surjective*. An interpretation mapping from $\mathcal{S}$ to $\mathcal{M}$ is surjective iff the corresponding representation mapping from $\mathcal{M}$ to $\mathcal{S}$ is total, i.e., iff every entity in a model (instance of the domain conceptualization) is represented by ***at least*** one (although perhaps many) entity in the representation *S*. An example of a *surjective* interpretation mapping is depicted in figure 2.15.

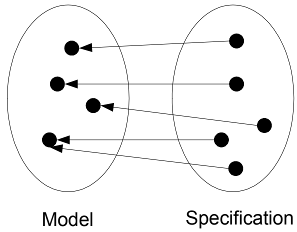


Figure 2-15 Example of a Complete Interpretation mapping from Specification to Model

The notion of completeness at the level of individual specifications is related to the notion of *ontological expressiveness* and, more specifically, *completeness* at the language level, which is perhaps the most important property that should hold for a representation system. A modeling language is said to be *complete* if every concept in a domain conceptualization is covered by at least one modeling construct of the language. Language incompleteness entails lack of expressivity, i.e., that there are phenomena in the considered domain (according to a domain conceptualization) that cannot be represented by the language. Alternatively, users of the language can choose to overload an existing construct, thus, undermining clarity.

In chapter 8 of this thesis, when evaluating the Unified Modeling Language (UML) in terms of the conceptualization proposed in chapters 4 to 7, examples of incompleteness abound. To mention two cases, there are several different sorts of *object types* and *part-whole relations* in the conceptualizations proposed in chapter 4 and 5, respectively, which are not directly represented by any construct of the language. In both cases, the distinct concepts present in the conceptualization are overloaded by the language constructs of *class* and *aggregation/composition*, respectively.

An incomplete modeling language is bound to produce incomplete specifications unless some existing construct is overloaded. However, the converse is not true, i.e. a complete modeling language can still be used to produce incomplete specifications. An example of the latter is shown in figure 2.16. In this picture, a domain element (the 3 $<$ 4 relation) is not present in the representation. In accordance with the detailed account of Grice's cooperative principle (specifically, that all necessary information is included), specification and language designers should attempt to ensure completeness as most readers assume this to be true. In summary, a modeling language should be complete w.r.t. a domain conceptualization and every element in a domain abstraction (instance of this domain conceptualization) must be represented by an element of a specification built using this language.

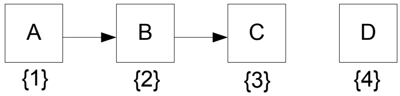


Figure 2-16 Example of an Incomplete diagram

2.3 Conceptualization and Ontology

In section 2.2, we advocate that the suitability of a language to create specifications in a given domain depends on how *"close"* the structure of the specifications constructed using that language resemble the structure of the models (domain abstractions) they are supposed to represent. To put it more technically, a specification $\mathcal{S}$ produced in a language $\mathcal{L}$ should be, at least, a homomorphism of the model $\mathcal{M}$ that $\mathcal{S}$ represents.

The framework presented in section 2.2 is ultimately based on the analysis of the relation between the structure of a modeling language and the structure of a domain conceptualization.

What is referred by *structure of a language* can be accessed via the description of the specification of *conceptual model underlying the language*, i.e., a description of worldview embedded in the language's modeling primitives. In (Milton & Kamierczak, 2004), this is called the *ontological metamodel of the language*, or simply, the *ontology of the language*. From a philosophical standpoint, this view is strongly associated with Quine (Quine, 1969), who proposes that an ontology can be found in the *ontological commitments* of a given language, that is, the entities the primitives of a language commit to the existence of. For example, Peter Chen's Entity Relationship model (Chen, 1976) commits to a worldview that accounts for the existence of three types of things: *entity*, *relationship* and *attribute*.

This idea can be understood in analogy to the distinction between a conceptual model and design model in information systems and software engineering. Whilst the former is only concerned with modeling a view of the domain according to a given application, the latter is committed to translating the model of this view on the most suitable implementation according to the underlying implementational environment and also considering a number of non-functional requirements (e.g., security, fault-tolerance, adaptability, reusability, etc.). Likewise, the specification of the conceptual model underlying a language is the description of what the *primitives of a language* are able to represent in terms of real-world phenomena. In some sense (formally characterized in chapter 3), it is the representation of a conceptualization of the domain in terms of the language's vocabulary. In the *design* of a language, these conceptual primitives can be translated into a different set of primitives. For example, it can be the case that a conceptual primitive is not directly represented in the actual abstract syntax of a language, but its modeling capabilities (the real world concept underlying it) can be translated to several different elements in the language's abstract syntax due to non-functional requirements (e.g., pragmatics, efficiency). Nonetheless, the design of a language is responsible for guaranteeing that the language's syntax, formal semantics and pragmatics are conformant with this conceptual model. From now on, the *Modeling Language* icon depicted in figure 2.6 represents the specification of the conceptual model underlying the language, or what we shall name the ***language metamodel specification***.

The *structure of domain conceptualization* must also be made accessible through an explicit and formal description of the corresponding portion of reality in terms of a concrete artifact, which is termed here a *domain reference ontology*, or simply, a ***domain ontology***. The idea is that a reference ontology should be constructed with the sole objective of making the best possible description of the domain in reality w.r.t. to a certain level of granularity and viewpoint. Additionally, it should be constructed using formal and methodological tools which have been developed by the area of *formal ontology* (Husserl, 1970) in philosophy. Finally, it should be application independent and not be biased towards a specific mathematical model or formal theory.

The notion of ontology as well as its role in the explicit representation of conceptualizations is discussed in depth and given a formal characterization in chapter 3.

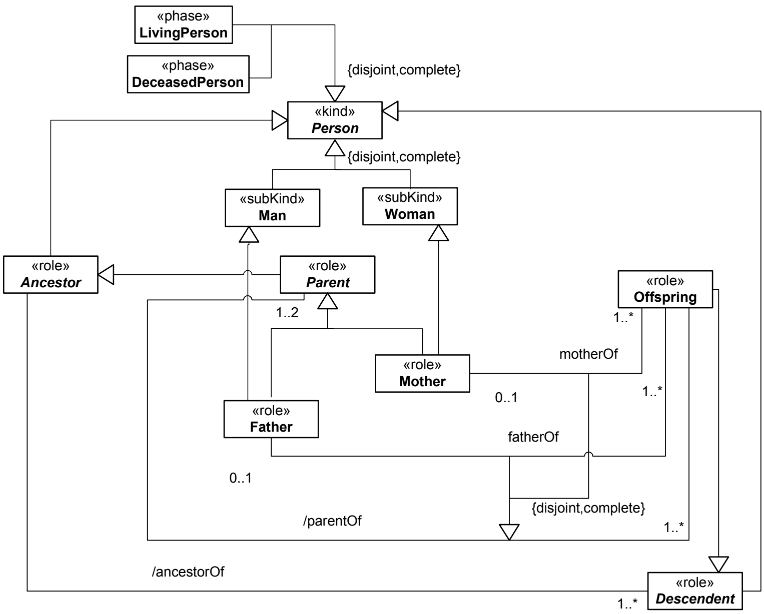
2.3.1 Designing a Visual Modeling Language for the Domain of Genealogy: An Example

In the sequel we illustrate these rather abstract notions discussed so far with an example in the domain of genealogical relations. This domain is simple and familiar and, hence, can be useful for illustration purposes. Nonetheless, it is still not a completely artificial example since there are, in practice, standard modeling languages to represent genealogical relations and a number of applications that make use of these standards (Howells, 1998). A certain conceptualization of this domain can be articulated by considering concepts such as *person*, *living person*, *deceased person*, *father*, *mother*, *offspring*, *fatherOf* and *motherOf*. These concepts are related to each other and have their interpretation constrained by axioms imposed on their definitions. Figure 2.17 depicts a domain ontology representing a possible conceptualization in this domain. The modeling primitives of the *UML profile* (OMG, 2003b) used to represent this ontology are part of the ontology representation language proposed throughout this work. The class stereotypes present in this specification (*kind*, *subkind*, *role* and *phase*), in particular, are discussed in depth in chapter 4.

The diagram in this picture is complemented by the following axioms:

– A person *x* is a *parentOf* person *y* iff *x* is *fatherOf y* or *x* is *motherOf y*;
– A person *x* is a *ancestorOf* person *y* iff *x* is *parentOf y* or there is a person *z* such that *z* is an *parentOf y* and *x* is *ancestorOf z*;
– A person cannot be its own ancestor (i.e., the *ancestorOf* relation is irreflexive);
– If a person *x* is an *ancestorOf* person *y* then *y* cannot be an *ancestorOf x* (i.e., the *ancestorOf* relation is asymmetric);
– If a person *x* is an *ancestorOf* person *y* and *y* is an *ancestorOf* person *z* then *x* is an *ancestorOf z* (i.e., the *ancestorOf* relation is transitive).

Figure 2-17 An Ontology for the Genealogy Domain



By representing a conceptualization of this domain in terms of this concrete artifact we can design a language to express phenomena in this domain capturing characteristics that this conceptualization deems relevant. For instance, according to this domain ontology, *Person* is an abstract type, i.e., one that cannot have direct instances. This is represented in this notation by depicting the name of type in italics. The abstract type *Person* is partitioned in two independent suptyping structures:

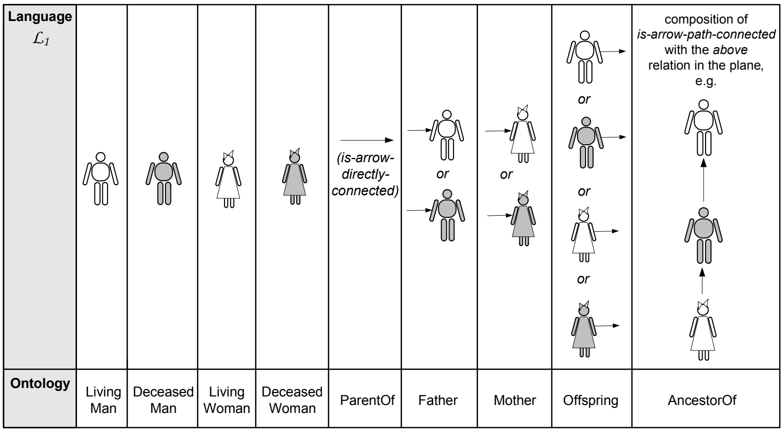
1. **Man, Woman:** this partition represents that every individual person (instance of type *Person*) in the universe of discourse is either a man (an instance of *Man*) or woman (instance of *Woman*). Moreover, due to the «subKind» stereotype, it states that every man is *necessarily* a man (in the modal logics sense), i.e., every instance of the type man is a man in every possible situation considered by the model. We say in this case that *Man* is a *rigid* classifier (see chapter 4). *Mutatis Mutandis* the same applies to instances of the type *Woman*. Finally, it states that both individual man and woman obey a *principle of identity* supplied by the type Person (due to the presence of the «kind» stereotype). The notion of a principle of identity is discussed in depth in chapter 4. For now, we can say that every instance of *Person* maintains its identity (i.e., it is the same Person) in every circumstance considered by the model;

2. **LivingPerson, DeceasedPerson:** this partition represents that every individual person in the universe of discourse is either a living person or a deceased one. However, in contrast to the **⟨Man,Woman⟩** partition, an instance of LivingPerson is not necessarily so (in the modal sense), i.e., LivingPerson is a *anti-rigid* classifier (see chapter 4). Every instance of *LivingPerson* is only *contingently* an instance of *LivingPerson* (again in the modal sense). That is to say that for every x such that x is *LivingPerson* there is a counterfactual situation in which x is not a *LivingPerson*, which in this case, implies that x is a DeceasedPerson in this counterfactual situation. One more, *Mutatis Mutandis,* the same applies to instances of *DeceasedPerson*. These facts are implied by the presence of «phase» stereotyped classifiers and the associated constraint that they must be defined in a partition.

A cross-relation of these two partitions give us four concrete classifiers, i.e., classifiers that can have direct instances, let us name them *LivingMan*, *DeceasedMan*, *LivingWoman* and *DeceasedWoman*. Every instance of person in a given situation is necessarily an instance of one of these classifiers. A suitable modeling language must have modeling primitives that conform to these constraints. Other constraints of the possible models according to this ontology include: that every *offspring* can have at maximum one *father* and one *mother*; the *ancestorOf* relation (defined to hold between instances of *Person*) is irreflexive, antisymmetric and antitransitive. The set of constraints captured in this ontology represents part of the conceptualized *structure of* *the domain*. This structure must be taken into account in the design (an evaluation) of a language to model genealogical relations.

By having a conceptualization (abstract entity) represented in terms of a domain ontology, and by applying the framework discussed in section 2.2, one can, in a precise manner, design a suitable modeling language for that given domain. In this particular case, we are able to design a language $\mathcal{L}_1$ whose metamodel specification $S_{\mathcal{L}1}$ is isomorphic to the ontology of figure 2.17. The primitives of this language are presented in figure 2.18 below.

Figure 2-18 Domain Concepts and their representing modelling primitives in $\mathcal{L}_1$



A number of characteristics should be observed about language $\mathcal{L}_1$. To start with, the language contains modeling primitives that represent all concrete classifiers (*LivingMan*, *DeceasedMan*, *LivingWoman*, *DeceasedWoman)* and non-derived relations present in the ontology (*fatherOf, motherOf)*. Consequently, we can say that the language is expressive enough to represent all characteristics of the domain that are considered relevant by the underlying ontology. Moreover, in the mapping from ontology *O* of figure 2.17 to the metamodel specification $S_{\mathcal{L}1}$ (of language $\mathcal{L}_1$) there is no case of construct redundancy, construct overload or construct excess.

In regards to the property of completeness, when mapping the elements of a domain ontology to a language metamodel specification we must guarantee that these elements are represented in their full formal descriptions. In other words, the language metamodel specification $S_{\mathcal{L}1}$ representing the domain ontology *O* of figure 2.17 must also represent in its well-formedness rules this ontology's full axiomatization. In formal, model-theoretic terms, this means that these entities should have the same set of logical models. In chapter 3, we discuss this topic in depth and present a formal treatment of this idea. The set of logical models of *O* represent the state of affairs in reality deemed possible by a given domain conceptualization. In contrast, the set of logical models of *S* stand for the world structures which can be represented by the grammatically correct specifications of language $\mathcal{L}_1$. In summary, we can state that if a domain ontology *O* is fully represented in a language metamodel specification $\mathcal{S}_{\mathcal{L}1}$ of $\mathcal{L}_1$, then the only grammatically correct models of $\mathcal{L}_1$ will be those which represent state of affairs in reality deemed possible by the domain conceptualization represented by *O*.

In this example, since we assume that *O* is fully represented in $\mathcal{S}_{\mathcal{L}1}$, as a consequence, the specifications depicted in figures 2.19.a, 2.19.b and 2.19.c cannot be considered syntactically valid specification of language $\mathcal{L}_1$.

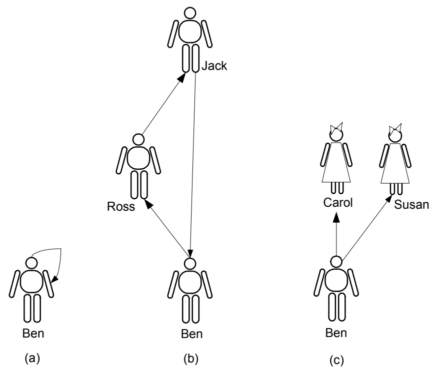


Figure 2-19 Examples of grammatically invalid models in language $\mathcal{L}_1$

In contrast, a valid specification in language $\mathcal{L}_1$ is depicted in 2.20 below.

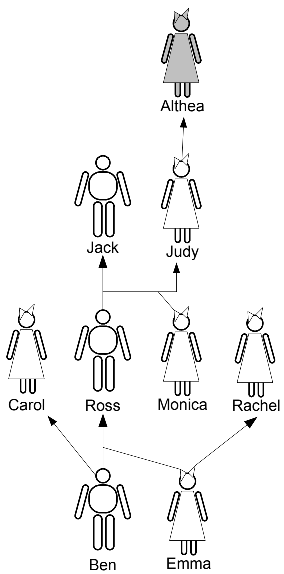


Figure 2-20 Example of a valid model in language $\mathcal{L}_1$

Another aspect that should be noticed is how the ontology of figure 2.17 contributes for improving pragmatic efficiency in the concrete syntax of $\mathcal{L}_1$. In (Gurr, 1999), when introducing the framework that has been adapted in section 2.2, the author uses regular algebraic structures to model a domain conceptualization. We claim that many additional benefits arise from a more complete representation of the domain conceptualization than the algebra used in his work. We defend the idea that the more we know about a domain the better we can design pragmatically effective languages. In particular, there are important meta-properties of domain entities (e.g., *rigidity*, *relational dependency*) that are not captured by ontologically-neutral mathematical languages (see chapter 3, section 3.4.2). The failure to consider these meta-properties hinders the possibility of accounting for other direct aspects of visual syntaxes. In the case of language $\mathcal{L}_1$, one can observe that:

1. The types *Man* and *Woman* are kinds and, thus, *rigid types*, which means that instances of these types will continue to be so as long as they exist in the model. In contrast, an individual man (or woman) can have the (intrinsic) properties of *begin alive* or *being dead* in different situations. In any case, the man which is alive in one circumstance and dead in another is the same man, i.e., he maintains his identity across situations. In language $\mathcal{L}_1$, the icons[5] used to represent instances of Person maintain the stable visual percept, which represents the dichotomy of the rigid types *Man* and *Woman*. The phases living and deceased are represented as variations of this visual percept, that is, the *same* visual percept can appear in different situations as one of the two variations;

2. The types modeled as concrete *Roles* in the ontology are *Father*, *Mother* and *Offspring*. These types are not only *anti-rigid* but also *relationally dependent*, i.e., the *same* instance *x* of Father can exist in another situation in the model without being a Father. Moreover, to be a Father is to be a Man who has (at least) one Offspring, i.e., for *x* to be a Father he must share a relational property with another individual who is an instance of Offspring. In $\mathcal{L}_1$, the *Parent* role is represented by the adjacency *relation* between the icon representing a Person and the arrow-head of the symbol representing the *parentOf* relation. Additionally, the Offspring role is represented by the adjacency relation between the icon representing a Person and the arrow-tail of the symbol representing the parentOf relation. This representation choice highlights the dependency of these roles on relations, which in turn, stand for mutual properties

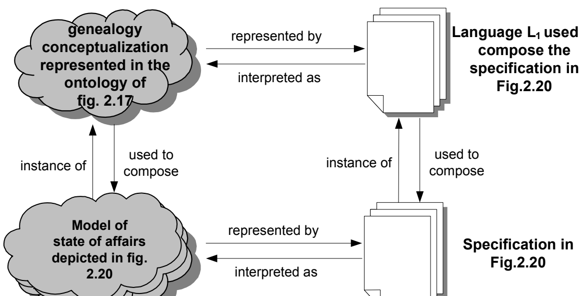
[5] Icon is used here in the Peircean sense, i.e. as a sign that physically resembles what it 'stands for' (Peirce, 1960).

shared by their relata. Moreover, it allows for the representation that x *qua* Man maintains its identity in the scope of different relations and across different situations.

3. The *ancestorOf* relation is represented by the above relation in the plane associated with the *arrow-path-connectedness*, i.e., if x and y are two persons that are path-connected and x is above y in the plane then x is an ancestorOf y. The composed relation *above-path-connect* is also irreflexive, asymmetric and transitive, i.e., a strict partial order relation. These are exactly the same meta-properties as enjoyed by the ancestorOf relation. For this reason, the conclusion that (x ancestorOf z) if (x ancestorOf y) and (y ancestorOf z) is directly inferred from (x is above-path-connected-to z) if (x above-path-connected-to y) and (y above-path-connected-to z).

By instantiating the pattern of figure 2.6 to this domain we obtain the correspondence depicted in figure 2.21. The ontology *O* of figure 2.17 is a concrete representation of a given conceptualization of the genealogy domain. In this case, we have the ideal situation that the metamodel of language $\mathcal{L}_1$ is identical to this ontology. The genealogy concepts represented in *O* are used to articulate models of individual state of affairs in reality. A specification in language $\mathcal{L}_1$ (such as the one of figure 2.20) is a concrete artifact representing one of these models.

Figure 2-21 Instantiating the pattern of figure 2.6 for the domain of genealogy



In this example, the suitability of language $\mathcal{L}_1$ is evaluated w.r.t. a specific conceptualization and, more precisely, w.r.t. a specification of this conceptualization captured in the domain ontology of figure 2.17. A question that one may ask is how to evaluate the adequacy of a conceptualization and the corresponding ontology? In other words, how good is a conceptualization as an abstraction of a given universe of discourse or portion of reality (see *abstracts* relation in figure 2.5) and how good is an ontology as a representation of this conceptualization. The construction of a particular conceptualization, i.e. a particular shared abstraction of reality, should be based on the consensus of a community of domain experts, and could be supported by both theoretical and empirical evidence. An ontology, conversely, is a concrete artifact which is developed by an engineering process (Gómez-Pérez, Fernández-López, & Corcho, 2004), which includes the activities of requirements specification and evaluation (among others). We shall resume the discussion about these issues in chapter 3.

2.3.2 Meta-Conceptualization and General Conceptual Modeling Languages

A domain such as the genealogy discussed in previous section is what we have previously named a *material domain* (Smith, 1989; Little, 2003) and a language, such as $\mathcal{L}_1$, designed to model phenomena in this domain is called a *domain-specific language*. The design of domain-specific languages is a current and important research topic (Kelly & Tolvanen, 2000; Tolvanen, Gray & Rossi, 2004; Bottoni & Minas, 2003) whose main benefits can be summarized by the following quote from Tolvanen, Gray and Rossi (ibid.): *"A [domain-specific language] raises the level of abstraction, while at the same time narrowing down the design space. The language follows the domain abstractions and semantics, allowing developers to perceive themselves as working directly with domain concepts. Industrial experiences of this approach show major improvements in productivity, time-to-market responsiveness and training time."*

In this work, however, the focus is not on designing or evaluating domain-specific languages. Contrariwise, one of the objectives is to propose a domain independent language, such as the UML profile used in figure 2.17, which can be used to represent domain ontologies in different material domains. Since a domain ontology is also a concrete artifact, it must be represented in some specification language $\mathcal{L}_0$. To be consistent with the position defended here, the language $\mathcal{L}_0$ used to represent individual domain ontologies should also be based on a domain conceptualization, in this case, a meta-conceptualization. This idea is depicted in figure 2.22.

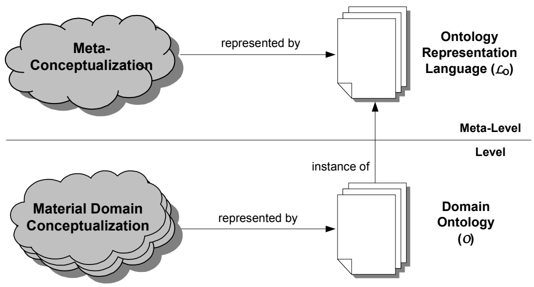


Figure 2-22 Relations between a Material Domain Conceptualization, Domain Ontology, General Meta-Conceptualization and Ontology Representation Language

Since we consider here a domain ontology as a special type of conceptual model specification, we also consider an *ontology representation language* as a special type of *general conceptual modeling language.* From now on, we simply use the term *conceptual modeling languages* to refer to the class of domain-independent languages that are used to create conceptual specifications in material domains.

With regard to figure 2.22 some questions that come to the mind are: What kinds of entities compose a meta-conceptualization? How to construct a suitable meta-conceptualization that a general conceptual modeling language should commit to? As we discuss in chapter 3 of this thesis, in the specific case of existing conceptual modeling and ontology representation languages, their design has been strongly influenced by goals other than domain appropriateness. Some of these languages have been created in an ad hoc way, in a manner which is governed much more by intuition and generalization of specific practical cases than based on a deep analysis of the underlying reality. Others have been purposefully biased by design and implementation issues or in favor of mathematical models that simplify the definition of the formal semantics of the language. Examples include conceptual modeling languages that only account for concepts that can be represented in current database of programming technology or, like most knowledge representation languages, which only consider concepts that can be treated efficiently in automatic inference process.

The position strongly advocated here is that a suitable conceptual modeling language should commit to a general (i.e., domain independent) theory of real-world categories that account for the ontological distinctions underlying language and cognition. This is exactly the business of the branch of philosophy called Formal Ontology and, in particular, descriptive metaphysics. The term Formal Ontology has been coined by Edmund Husserl (Husserl, 1970) as an analogy to Formal Logic. Whilst Formal Logic deals with formal logical structures (e.g. truth, validity, consistency), Formal Ontology deals with formal ontological structures (e.g. theory of parts, theory of wholes, types and instantiation, identity, dependence, unity) which apply to all domains. The term Formal here is, thus, less related to the sense of precise, mathematical and more related to its more ancient meaning, namely, of something that obtains in all material spheres of reality.

In summary, while domain conceptualizations and, consequently, *domain ontologies* are established by the consensus of a community of users w.r.t. a material domain, a conceptual modeling language that can be used to express these domain ontologies must be rooted in a domain independent philosophically and cognitively well-founded system of real-world categories, i.e. a *foundational (upper-level)* ontology (Schneider, 2003a; Masolo et al., 2003a; Heller & Herre, 2004).

These issues are discussed in detail in chapter 3.

2.4 Ontology-Based Semantics and Language Comparability

In the discussions carried out so far we have considered just one aspect of the relation between ontology and language. We have proposed that a *reference ontology* should be constructed as the best possible representation of a conceptualization, with the sole purpose of being a truthful representation of the domain in reality. Therefore, a modeling language in a domain can be designed (or evaluated) for domain appropriateness and comprehensibility appropriateness via the harmonization of the worldview underlying the language (i.e, the ontological metamodel of the language) with the one described by the domain ontology.

However, there is another scenario in which reference ontologies can play an important role, namely, when comparing models produced by different languages whose domain conceptualizations overlap. Suppose we have two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ that commit to the conceptualizations C_1 and C_2, respectively. Now supposed that C_1 and C_2 can be combined in a more general conceptualization C. Assume the two languages have been developed to be used in different problem-solving tasks and intend to represent different (aspects of) entities of the domain conceptualization C. As a consequence, the languages have different syntaxes and semantics. One question that arises is: how to relate (compare, translate) specifications written in $\mathcal{L}_1$ and $\mathcal{L}_2$?

Since they have different syntaxes and semantics, $\mathcal{L}_1$ and $\mathcal{L}_2$ should not be compared only in syntactical terms. Languages can have equal syntaxes with radically different semantics as much as they can have different syntaxes with equivalent mappings to a common semantic domain. For the case of automatic translation, the formal semantics of $\mathcal{L}_1$ and $\mathcal{L}_2$ cannot guarantee consistency in the results either. For example, if C is a (alternative) conceptualization that refers to a domain of family relations, the expression `(x related-to y)` in $\mathcal{L}_1$ can be wrongly translated to an expression `brother(a,b)` in $\mathcal{L}_2$, since the *related-to* relation satisfies all the formal axioms of brotherhood (irreflexive, symmetric, transitive).

Semantic comparability is a challenge in enterprise engineering due to the lack of interoperability among different (yet coexistent) process models. Many manufacturing engineering and business software applications use process information, including manufacturing simulation, production scheduling, manufacturing process planning, workflow, business process reengineering, product realization process modeling, and project management. Since each of these applications utilizes process information in a different way, process information is also represented differently in each application. This problem is even more critical, since enterprise systems must manage the heterogeneity inherent to its various sub-domains, by integrating different models into coherent frameworks (e.g., enterprise models for processes, structure, goals, deontic assignments).

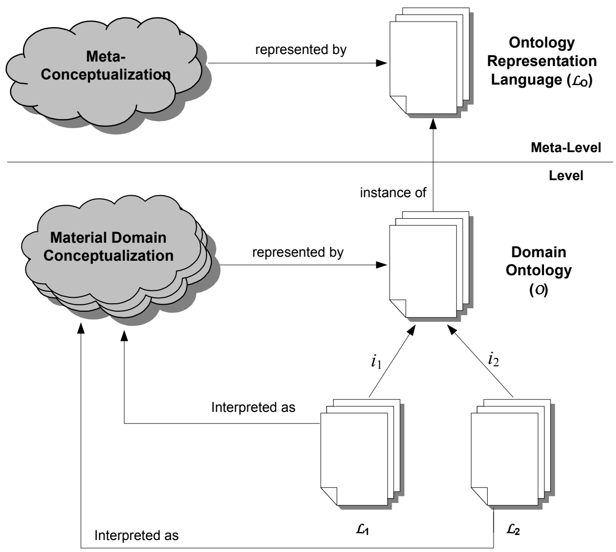
This problem has been currently addressed by the NIST PSL (Process Specification Language) Project[6] and the community working on the Semantic Web (Euzenat, 2001). For instance, (Ciocoiu & Nau, 2000) provide a formal definition of what they term *ontology-based semantics* for enabling automated model translation between different first-order declarative languages. Their proposal is that, for a domain-specific language $\mathcal{L}$, a domain ontology O can serve as a specification of the implicit assumptions of the conceptualization underlying $\mathcal{L}$. If a mapping from the constructs of $\mathcal{L}$ to formulas in O can be defined, this mapping can, in turn, be used to derive the (ontology-based) semantics of $\mathcal{L}$. In simple terms, the ontology-based semantics of a specification $\mathcal{S}$ in language $\mathcal{L}$ is defined in terms of a set of logical models U of $\mathcal{S}$ such that U has the desired property of obeying the constraints imposed by ontology O. Finally, the notion of ontology-based models is used to define translations between a specification $\mathcal{S}_1$ in language $\mathcal{L}_1$ and a specification $\mathcal{S}_2$ in language $\mathcal{L}_2$, namely, $\mathcal{S}_2$ is an *ontology-based partial translation* of $\mathcal{S}_1$ iff every ontology-based model of $\mathcal{S}_1$ is also an ontology-based model model of $\mathcal{S}_2$. Then, a specification $\mathcal{S}_2$ in language $\mathcal{L}_2$ is an *ontology-based translation* of specification $\mathcal{S}_1$ in language $\mathcal{L}_1$ iff $\mathcal{S}_2$ is an ontology-based partial translation of $\mathcal{S}_1$ and $\mathcal{S}_1$ is an ontology-based partial translation of $\mathcal{S}_2$.

[6] http://ats.nist.gov/psl/

In (Ciocoiu & Gruninger, 2000), this approach is exemplified via the translation between process models written in ILOG and IDEF3. This idea is illustrated in Figure 2.23. The solution creates logical interpretations i_1 and i_2 that relate the constructs of $\mathcal{L}_1$ and $\mathcal{L}_2$ to a common domain ontology O. The assumption here is that O is general enough so that its underlying conceptualization subsumes the parts of a conceptualization C targeted by $\mathcal{L}_1$ and $\mathcal{L}_2$. More than that, O is a formal logic theory through which the semantics of the conceptualization shared by $\mathcal{L}_1$ and $\mathcal{L}_2$ can be defined. Unlike the conceptual metamodels of $\mathcal{L}_1$ and $\mathcal{L}_2$, O is not biased towards a selection of concepts in C for the solution of specific tasks. It is a specification of domain-knowledge perceived as important by a community in order to address the needs of possibly several different applications.

We shall no longer discuss the topic of ontology-based translations in this thesis. The reader should refer to (Ciocoiu & Nau, 2000) for formal definitions of the approach just described, and to (Ciocoiu & Gruninger, 2000) for an interesting exemplification of these ideas.

Figure 2-23 Defining the Semantics of two Domain-Specific Languages via a shared Domain Ontology



2.5 Final Considerations

In this chapter we discuss the relation between a conceptual modeling language and the domain in reality that this language is supposed to represent. We propose that the *domain appropriateness* and *comprehensibility appropriateness* of a modeling language can be systematically evaluated by comparing a concrete representation of the worldview underlying the language captured in the *language metamodel specification*, with an explicit and formal representation of a conceptualization, or a *reference ontology*. We advocate that in the best case, these two models are isomorphic and, hence, we propose a framework comprising of a number of properties (*lucidity*, *soundness*, *laconicity*, *completeness*) that must be reinforced for this isomorphism to take place. This framework combines the proposals of (Wand & Weber, 1989, 1990; Weber, 1997), which aim at the evaluation of representation systems, and (Gurr, 1998, 1999), which focus on the evaluation of individual representations.

If described formally, ontologies can also be beneficial as axiomatized theories through which the semantics of modeling languages can be defined for the purpose of semantic model interoperation (e.g., integration, comparison, translation). Additionally, if described in a language that commits to a suitable meta-conceptualization, ontologies can play an important role in informing properties that support the design of pragmatically efficient systems of visual concrete syntax.

The discussion carried out in this chapter encompasses both the level of material domains and corresponding domain-specific modeling languages, and the (meta) level of a domain-independent (meta) conceptualization that underpins a general conceptual (ontology) modeling language. The idea of maintaining a discussion in general terms was intended to show that the evaluation framework and the principles discussed here can be applied to the design of conceptual modeling languages irrespective of the generalization level to which they belong.

Nevertheless, due to the objectives and scope of this thesis, in the following chapters our discussion will focus on the level of meta-conceptualizations and of general conceptual modeling languages. Although the proposals made here contribute to the area of domain-specific languages design methodologies (as acknowledged, for instance, in Girardi & Serra, 2004), the focus of this thesis is certainly not on the design of this class of languages, neither with domain conceptualizations nor domain ontologies.

Chapter 3

Ontology

In this chapter we discuss the topic of ontologies, which is a cornerstone of the work developed throughout this thesis.

We start, in section 3.1, by giving a historical perspective on ontology from a philosophical point of view. First, we present the different definitions of the term ontology in philosophy, and, subsequently, discuss the importance of ontological investigations for science, in general, and for conceptual modeling, in particular.

The past decades have observed an increasing interest in ontologies in a wide range of computer-related applications. Section 3.2 discusses four areas that, historically, have been prominently responsible for creating the demand for the use of ontologies in computer science, namely, information systems, domain engineering, artificial intelligence and the semantic web. For each of these areas we briefly discuss the expected benefits and provide examples of ontologies and ontology-based applications that have been developed along the years. In addition, we discuss the characteristics of ontology modeling languages that are typically used in some of these areas.

Although ontology has a clear definition in philosophy, there is a substantial terminological confusion in the different areas of computer science regarding what the term is supposed to denote. Section 3.3 is the most important section of this chapter and aims at a terminological clarification and at establishing a formal characterization of the way the term is used in the remaining of this work. Furthermore, the section elaborates on the relation between the definition provided and the notions of conceptualization and language as discussed in chapter 2.

In section 3.4, we present some final considerations.

3.1 Ontology in Philosophy

The Webster dictionary (Merriam-Webster, 2004) defines the word ontology as:

(D1). a branch of metaphysics concerned with the nature and relations of being;
(D2). a particular theory about the nature of being or the kinds of existents;
(D3). a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system.

The term ontology was coined in the 17th century in parallel by the philosophers Rudolf Göckel in his *Lexicon philosophicum* and by Jacob Lorhard in his *Ogdoas Scholastica* (figure 3.1). The term *Ontologia*, however, was popularized in philosophical circles only in 18th century by the publication in 1730 of the *Philosophia prima sive Ontologia* by Christian Wolff (figure 3.2).

Etymologically, *ont-* comes from the present participle of the Greek verb *einai* (to be) and, thus, the latin word ***Ontologia*** (*ont-* + *logia*) can be translated as *the study of existence*.

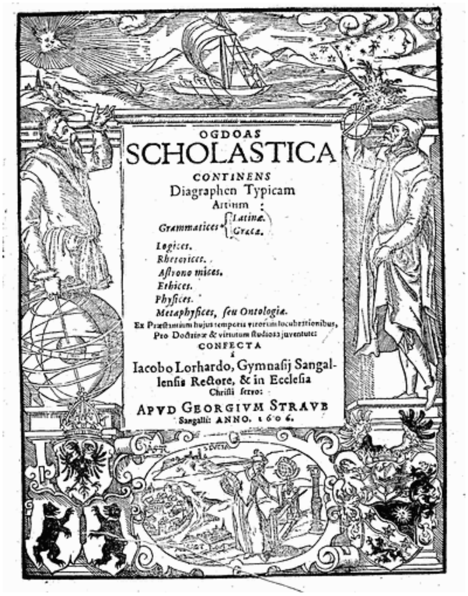


Figure 3-1 Cover of Jacob Lorhard's book *Ogdoas Scholastica* from 1606


PHILOSOPHIA
PRIMA,
SIVE
ONTOLOGIA,
METHODO SCIENTIFICA
PERTRACTATA,
QUA
OMNIS COGNITIONIS
HUMANÆ PRINCIPIA
CONTINENTUR
AUCTORE
CHRISTIANO WOLFIO,
EDITIO NOVA
PRIORI EMENDATIOR.
PROSTAT IN OFFICINA LIBRARIA RENGERIANA.

Figure 3-2 Cover of Christian Wolff's book *Philosophia prima sive Ontologia* from 1730

In the sense (D1) of the aforementioned Webster definition, ontology is the most fundamental branch of metaphysics. Aristotle was the first western philosopher to study metaphysics systematically and to lie out a rigorous account of ontology. He described (in his *Metaphysics* and *Categories*) ontology as *the science of being qua being* (Corazzon, 2004). According to this view, the business of ontology "…is to study the most general features of reality and real objects" (Peirce, 1935), i.e., the study of the generic traits of every mode of being. As opposed to the several specific scientific disciplines (e.g., physics, chemistry, biology), which deal only with entities that fall within their respective domain, ontology deals with transcategorical relations, including those relations holding between entities belonging to distinct domains of science, and also by entities recognized by ***common sense***.

Ontology aims to develop theories about, for example, persistence and change, identity, classification and instantiation, causality, among others. Ontological questions include questions such as: what kinds of entities exist? What differentiates objects from events and how are they related? What are the properties of a thing and how are they related to the thing itself? What is the essence of an object? Does essence precede existence? Are things bundles of properties? Is an object equal to the sum of its parts? Are there Natural Kinds? Is change possible without a changing thing? These are general but factual questions, only comprehensive rather than specific. They are also fundamental to science regardless if we are to talk about properties of atoms, human organs or insurance claims, if we are to develop theories of physical, mental or social events, or if we are to theorize on the parthood relations between an individual and a society, a heart and a human body and the first half and an entire football game.

There are many ontological principles that are utilized in scientific research, for instance, in the selection of concepts and hypothesis, in the axiomatic reconstruction of scientific theories, in the design of techniques, and in the evaluation of scientific results. Examples of scientific questions that are actually metaphysical ones include (Bunge, 1977, p.19):

- Is there an ultimate matter? This question triggered Heisenberg's 1956 theory of elementary particles;
- Is society anything beyond and above individuals that compose it or are there special societal laws in addition to laws governing individual behavior? This is a central dispute in the methodology and philosophy of social sciences;
- Are biological species embodiments of Platonic archetypes, or just concrete populations? Or perhaps a single individual scattered in space and time? This question is asked everyday by taxonomists (Ereshefsky, 2002).

Examples of metaphysical hypothesis underlying scientific research include (Bunge, 1977, p.16):

- There is a world external to the cognitive subject;
- The world is composed of things that are grouped into systems or aggregates;
- Every system except the universe, interacts with other systems in certain respects and is isolated from other systems in other respects;
- Nothing comes out of nothing and no thing reduces to nothingness;
- Everything changes;
- There are laws and everything abides by laws (otherwise, the experimental scientific method would not be possible);
- There are levels of organization: physical, chemical, social, psychological, biological, etc. The so-called higher levels emerge from other levels in the course of processes; but, once formed they enjoy a certain stability with laws of their own. Otherwise we would have to know everything about physics and chemistry before knowing about organisms and societies.

Moreover, recent scientific theories that make explicit use of metaphysical concepts abound. To cite one example, in (Penrose, 2000), the author uses a Platonic theory of Universals to explain the mind's ability to execute non-computable processes.

Finally, in the axiomatization of scientific theories, some of the following concepts are to occur in an explicit fashion: part, composition, system, state of affairs, relations, boundary, causality, state, event, change, property, law, possibility, process, space and time. However, the specific axioms of these theories will usually not tell us anything about these fundamental and generic concepts. Science just borrows them, leaving them in an intuitive and pre-systematic state. In other words, these generic concepts are common to a number of sciences, so that no single scientific discipline takes the trouble to regiment them (Bunge, 1977). The same holds from concepts in computer science and, in particular, in conceptual modeling. Concepts such as part and whole, instantiation and classification, attribution and relationships, causality, interaction, among others, are represented in the modeling primitives of several conceptual modeling language or, at minimum, are used in discourse of the computer science literature. Nonetheless, there is a lack of theoretical support in the area for precisely defining the meaning of these concepts. To quote Ron Weber, in his *Ontological Foundations of Information Systems* (Weber, 1997): *"A discipline that calls itself the information systems discipline ought to know what the term 'information systems' means…[, and for that to take place,] its members need to be able to explain precisely what they mean by the term 'system'."*

In summary, every science presupposes some metaphysics. However, metaphysics and science can be distinguished by the scope of their problems. Whereas the scientist deals with rather specific questions of fact, the ontologist is concerned with all the factual domains.

In the beginning of the 20th century the German philosopher Edmund Husserl coined the term ***Formal Ontology*** as an analogy to Formal Logic. Whilst Formal Logic deals with formal logical structures (e.g., truth, validity, consistency) independently of their veracity, Formal Ontology deals with formal ontological structures (e.g., theory of parts, theory of wholes, types and instantiation, identity, dependence, unity), i.e., with formal aspects of objects irrespective of their particular nature. The unfolding of Formal Ontology as a philosophical discipline aims at developing a system of general categories and their ties, which can be used in the development of scientific theories and domain-specific common sense theories of reality. In other words, ontology in the first sense of Webster's definition aforementioned contributes to the development of ontologies in the second sense.

The first ontology developed in sense (D2) is the set of theories of Substance and Accidents developed by Aristotle in his *Methaphysics* and *Categories*[7]. Since then, ontological theories have been developed by

[7] We refer to (Aristotle, 1984) for a modern edition of Aristotle's complete works.

innumerous philosophers such as G.W. Leibniz (Leibniz, 1981), C.S. Peirce (Peirce, 1935), Alfred North Whitehead (Whitehead, 1929), Bertrand Russel (Russel, 1940), Willen Van Orman Quine (Quine, 1953, 1969), Nelson Goodman (Goodman, 1951), Peter Strawson (Strawson, 1959), Edmund Husserl (Husserl, 1970), and Saul Kripke, (Kripke, 1982) to name just a few whose theories appear latter in this thesis. Moreover, philosophical ontology is currently an active area in philosophy, and formal ontologies have been built by contemporary philosophers such as Mario Bunge (Bunge, 1977), Eli Hirsch (Hirsch, 1982), Anil Gupta (Gupta, 1980), Jacques van Leeuwen (van Leeuwen, 1991), Rodderick Chisholm (Chisholm, 1996), David Armstrong (Armstrong, 1989, 97), E.J. Lowe (Lowe, 2001), Peter Simons (Simons, 1987), Peter Gärdenfors (Gärdenfors, 2000), David Wiggins (Wiggins, 2001), Kevin Mulligan (Mulligan & Simons & Smith, 1984), Barry Smith (Smith, 1998), Achille Varzi and Roberto Casati (Casati & Varzi, 1994). Finally, more recently a number of ontological systems in the sense (D2) have been constructed in projects related to computer science (Masolo et.al, 2003a; Heller & Herre, 2004), and under the auspices of a new discipline called *Applied Ontology* (Masolo et.al, 2003b). It is with this last kind of ontologies that this thesis is mainly concerned, i.e., with formal ontological theories that can be developed and *applied* in the solution to problems in the fields of computer and information sciences and, in particular, of conceptual modeling.

3.2 Ontology in Computer and Information Sciences

Since the word ontology has been mentioned in a computer related discipline for the first time (Mealy, 1967), ontologies have been applied in a multitude of areas in computer science. The first noticeable growth of interest in the subject in mid 1990's was motivated by the need to create principled representations of domain knowledge in the knowledge sharing and reuse community in AI, which motivated the creation of forums such as the conference series FOIS (Formal Ontology and Information Systems)[8]. Nonetheless, an explosion of works related to the subject only happened in the past two years. Just to illustrate this point, the paper submission rate from the first International Semantic Web Working (SWWS) Symposium in 2001 (Cruz, 2001) to the third edition of the International Semantic Web Conference (ICSW) (Fensel & Sycara & Mylopoulos, 2003) has increased by almost 400%.

---

[8] http://www.fois.org/

According to (Smith & Welty, 2001), historically there are three areas mainly responsible for creating a demand for the application of ontologies in computer science, namely, (i) database and information systems; (ii) software engineering (in particular, domain engineering); (iii) artificial intelligence. In the sequel, we discuss the use of ontologies in these areas. Additionally, we also include a discussion of the topic in the context of the Semantic Web, due to the important role played by this area in the current popularization of the term.

3.2.1 Ontology in Information Systems

According to (Smith, 2004), the term *"ontology"* in the computer and information science literature appeared for the first time in 1967, in a work on the foundations of data modeling by S. H. Mealy, in a passage where he distinguishes three distinct realms in the field of data processing, namely: (i) *"the real world itself"*; (ii) *"ideas about it existing in the minds of men"*; (iii) "*symbols on paper or some other storage medium".* Mealy concludes the passage arguing about the existence of things in the world regardless of their (possibly) multiple representations and claiming that *"This is an issue of ontology, or the question of what exists"* (Mealy 1967. p. 525.). In the end of this passage, Mealy includes a reference to Quine's essay *"On What There Is"* (Quine, 1953).

The fields of data and information modeling have been a fruitful ground for the applications of ontological theories, either implicitly or explicitly. In the 1970s, the so-called three schema architecture has been proposed in the database field (Jardine, 1976). The architecture suggests the distinction between *implementation schemas* (describing physical ways of storing data and procedures), *presentation schemas* (concerned with external interfaces to the user), and *conceptual schemas* (focusing of the description of the characteristics of the elements in the universe of discourse). In order to address the problem of how to create these conceptual schemas, different set of modeling concepts have been proposed.

First, the so-called *logical models* were proposed in early 1970's. Examples include the relational and network models. These models offered a set of abstract modeling primitives that were independent of physical modeling concepts but which, unfortunately, were deemed flat (i.e., lacking structure[9]) and unintuitive as how they should be used for modeling purposes (Mylopoulos, 1998).

Soon after logical models were proposed, different meta-conceptualizations were proposed which offered more expressive facilities

[9] See discussion on section 3.4.2 of this chapter.

for modeling applications and structuring information bases. The *semantic model* proposed by Jean-Raymond Abrial in 1974 (Abrial, 1974) and Peter Chen's *entity-relationship model* (Chen, 1976) are included in the category of languages that are now known as conceptual modeling (or semantic data modeling) languages.

The creation of both logical and conceptual models by the database and information modeling community was solely motivated by the search for better concepts that could be used for creating representations of a certain portion of reality. Both the Semantic model and the ER model were committed to a world view and based on the *ontological assumption* that the structural aspects of the world could be articulated by using the concepts of *entity* and *relationship*. However, none of these efforts took ontology seriously, in the sense that the choices of categories that are part of the conceptualization underlying these languages were not based on Ontology in the philosophical sense.

As pointed by (Smith & Welty, 2001), the *ad hoc* and inconsistent modeling that marked the early days of conceptual modeling led to many of the practical database integration problems that we face today. In order to tackle some of these problems and to provide a sound basis for the selection of modeling concepts that should underpin information system grammars, several researchers started to found their work on philosophical ontologies. Examples include the use of philosophical ontologies for analysis and evaluation of: (i) *information systems grammars* (Milton & Kamierczak, 2004; Shanks & Tansley & Weber, 2003; Evermann and Wand, 2001a,b; Gemino, 1999; Green and Rosemann, 2000; Opdahl and Henderson-Sellers, 2001; Opdahl & Henderson-Sellers & Barbier, 1999; Parsons and Wand, 1991; Wand and Weber, 1989, 1993; Wand & Storey & Weber, 1999); (ii) *reference models* (Fettke and Loos, 2003); (iii) *data quality* (Wand and Wang, 1996); (iv) *off-the-shelf system (COTS)* (Soffer et. al, 2001).

Along the years, there has been an increasing interest in the use of philosophical ontology in the conceptual modeling and information systems community as a foundation for their discipline. Research results following this idea are strongly related to the objectives of this work and will be thoroughly discussed in the development of this dissertation.

3.2.2 Ontology in Domain Engineering

Independently of these developments in the information systems community, yet another sub-field of computer science, namely software engineering, began to recognize the importance of what came to be known as *domain engineering* (Arango & Williams & Iscoe, 1991). This was mainly motivated by the need to reduce the disproportional costs in software maintenance and the need to reinforce software reuse in a higher level of abstraction than merely programming code (Arango, 1994).

In general, a domain engineering process is composed of the following subactivities: *domain analysis* and *domain design*, the latter being further decomposed in *infrastructure specification*, *infrastructure implementation*. Intuitively, domain engineering can be considered analogous to software engineering (software application engineering), however, operating in a meta-level (see table 3.1), i.e., instead of uncovering requirements, designing and implementing a specific application, the target is on a family of applications in a given domain (Arango & Prieto-Diaz, 1994).

Table 3-1 A comparison between Domain Engineering and Application Engineering activities (based Arango & Prieto-Diaz, 1994)

Application Engineering	Domain Engineering
Requirements Analysis	Domain Analysis
Application Design	Infrastructure Specification
Application Implementation	Infrastructure Implementation

The term *Domain Analysis* appears for the first time in (Neighbors, 1981) with the following definition: *"Domain Analysis is an attempt to identify objects, operations and relations that domain experts perceive as important in a given domain"*.

The product of a domain analysis phase is a *domain model*. A domain model defines objects, events and relations that capture similarities and regularities in a given domain of discourse. The resulting model is an architecture comprised of conceptual components that are common to a family of applications (*reuse*). It can be used to identify, explain and predict facts in a given domain, which can hardly be observed directly (*problem-solving*). Moreover, it serves the purposes of a unified reference model to be used when ambiguities arise in discussions about the domain (*communication*) and a source of knowledge that can be used in a learning process about that domain. In summary, the specification produced by the domain modelling activity is *"a shared representation of entities that domain experts deem relevant in a universe of discourse, which can be used to promote problem-solving, communication, learning and reuse in a higher level of abstraction"* (Arango, 1994).

Actually, more than a mere analogy, domain and application engineering are complementary disciplines and can be interrelated in a process that contemplates both development *to reuse* and development *with reuse.* Figure 3.3 depicts schematically how these disciplines are integrated in the so-called *two-level life cycle* (Falbo et. al, 2002a). Some of these relations are briefly elaborated below.

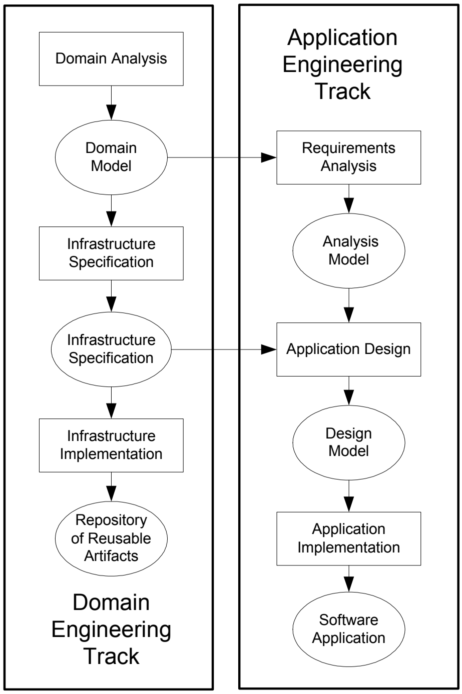


Figure 3-3 Domain Engineering and *Application Engineering* (based on Falbo et al., 2002a)

The result of a domain engineering process is a reusable infrastructure or *framework* (development for reuse). A framework can be (re)utilized in instances of software engineering processes for the construction of a several specific applications (development with reuse) whose requirements have been defined during the requirements analysis phase of each specific application.

In addition to being the basis for the development of framework, a domain model can also be used in the application requirements analysis phase to improve communication and understanding of the domain and to help in the requirements elicitation process (Falbo & Menezes & Rocha, 1998).

For a framework to be representative and, therefore, useful for potential applications in a domain, it must embed a correct conceptualization of the entities perceived as important by domain experts. The same must be true for a domain model to be useful as a shared reference for communication and problem-solving in application requirements analysis. Analogous to the problem faced by conceptual modelers in the database community, the challenge in *domain modeling* is again *finding the best concepts that can used to create representations of phenomena* *in a universe of discourse that are both as reusable as possible and still truthful to reality*.

This field, too, was severely debilitated by a lack of concrete and consistent formal bases for making modeling decisions (Smith & Welty, 2001). Despite the strong correspondence between domain models and what is named *domain ontologies* in AI (see section 3.2.3), only very recently ontologies started being used as a foundation for domain engineering. For instance, (Falbo et. al, 2002a) presents an ontology-based approach for software reuse and discusses how ontologies can support several tasks of a reuse-based software process. In (Falbo & Guizzardi & Duarte, 2002), an ontological approach for domain engineering (named ODE) is advocated and a systematic approach for ontological engineering is proposed. (Guizzardi & Falbo & Pereira Filho, 2001a), (Guizzardi & Falbo & Pereira Filho, 2001b) and (Guizzardi & Falbo & Pereira Filho, 2002) propose a modeling language for building ontology-based domain models and a systematic approach for deriving object-oriented frameworks from them. The framework derivation methodology proposed comprises a spectrum of techniques, namely, mapping directives, design patterns and formal translation rules. In particular, (Guizzardi & Falbo & Pereira Filho, 2001b) introduces a design pattern for guaranteeing the preservation of some ontological properties of part-whole relations (irreflexivity, asymmetry, transitivity and shareability) in object-oriented implementations.

(Falbo & Guizzardi & Duarte, 2002) and (Guizzardi & Falbo & Pereira Filho, 2002) exemplify the approaches proposed by constructing ontologies and deriving the corresponding object-oriented frameworks for the domains of software process and software quality, respectively. Fragments of these ontologies are depicted in figure 3.4 and 3.5, in that order. The representation language LINGO used in the picture is a set-theoretical language proposed in (Falbo & Menezes & Rocha, 1998) and (Guizzardi & Falbo & Pereira Filho, 2001a). The complete boxes represent concepts, boxes with open sides represent (material) relations, arrows represent subsumption relations (the arrow head pointing to the most general concept) and (hollow) circles represent shareable part-whole relations (the circle being connected to the whole. The cardinality constraints adjacent to a concept constrain the opposed navigational end of the relation. For example, in figure 3.4, a *Measurable Quality Characteristic can be measured by one-to-many Metrics*. Moreover, whenever omitted, cardinality constraints should be interpreted as zero-to-many. Finally, grey boxes represent concepts imported from other ontologies. For example, in the ontology of figure 3.5, the concept *Software Process* is imported from the Software Process ontology of figure 3.4. This picture also shows an axiom in this ontology which states that: *the resources allocated to a compound software activity*

A are those (and only those) which are allocated to the (sub)activities that are proper-parts of A.

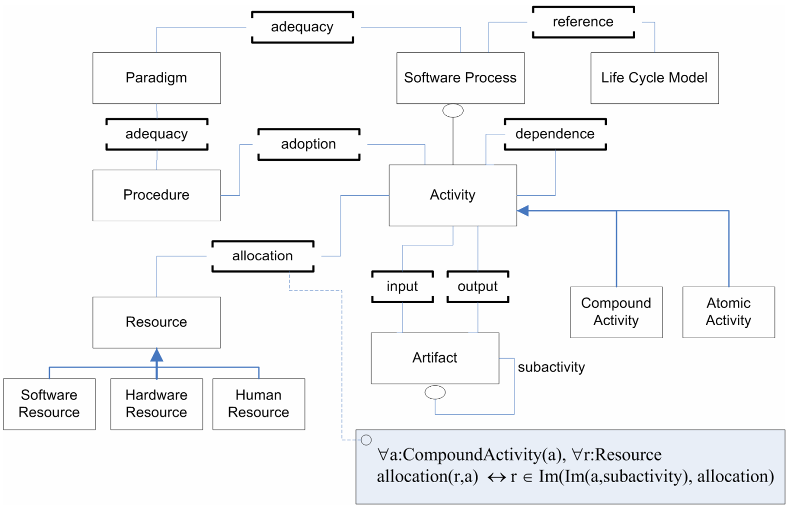


Figure 3-4 Excerpts of a Software Process Ontology developed using ODE and LINGO

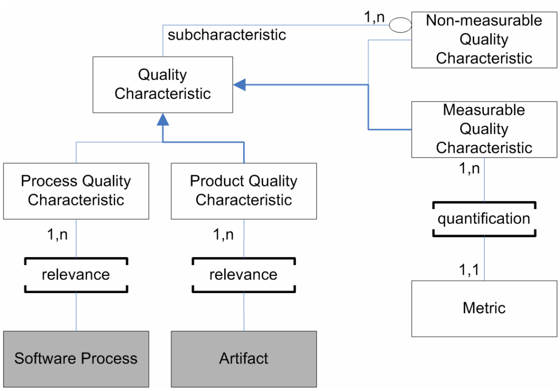


Figure 3-5 Excerpt of a Software Quality Ontology developed using ODE and LINGO

The ontology of figure 3.4 is used to generate the framework shown in figure 3.6. One can observe that, by using the transformation rules proposed by the methodology, the axioms in the domain ontology are systematically and explicitly mapped onto methods in the target framework. In the case exemplified, an invocation of the method *usage* in an *Activity* object returns the solution set of the corresponding axiom in the ontology, i.e., the set of resources used in by that specific *Activity*. This solution increases the reusability of the produced framework by representing explicitly the methods that address the ontology *competence questions*

(Gruninger & Fox, 1994a), as opposed to have domain knowledge hidden inside the code.

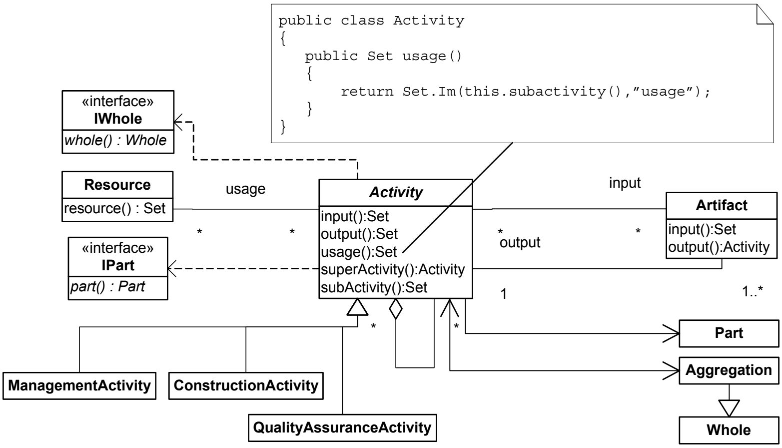


Figure 3-6 Fragment of a *Software Process Framework* derived from a *Software Process Ontology*

This framework can be reused and extended in a software engineering *development with reuse* process to address the needs of specific applications. For example, in (Falbo et. al, 2002b), the frameworks derived from the ontologies in figure 3.4 and 3.5 are used in the development of tools that are integrated in a software engineering environment. In particular, the *process framework* is used in the development of *process definition* (figure 3.7) and *project tracking* tools and the *quality framework* in the development of a *quality control application* (figure 3.8).

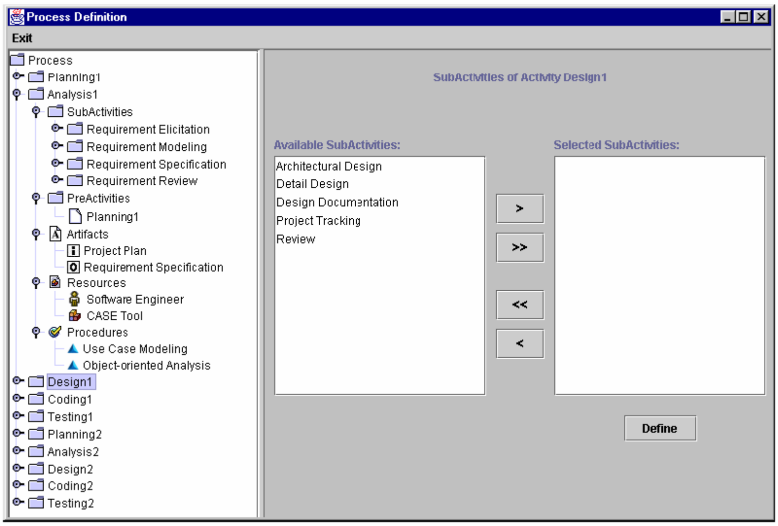


Figure 3-7 Process Definition Tool developed by (re)using a *Software Process Framework*

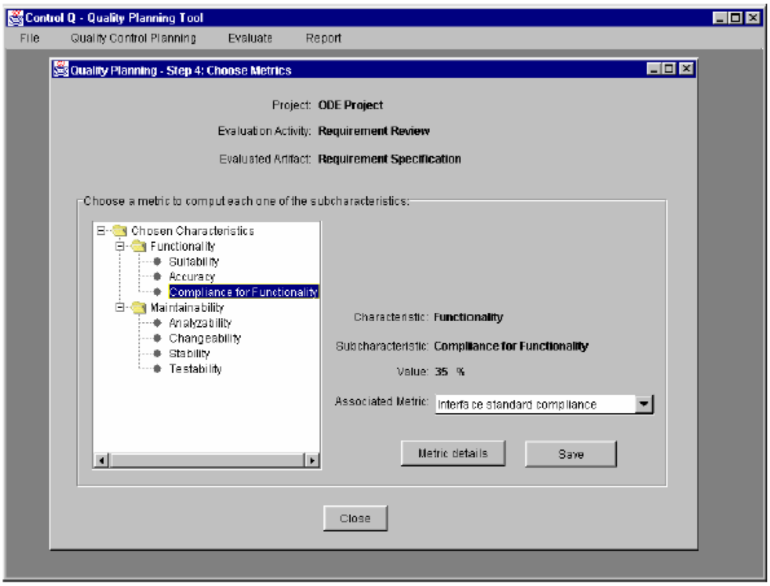


Figure 3-8 Quality Control Tool (*Control Q*) developed by (re)using a *Software Quality Framework*

A number of other domain ontologies and frameworks have been developed using ODE and LINGO in domains such as *resource allocation* (Guizzardi & Falbo & Pereira Filho, 2001a), *Software Risk* (Falbo et. al, 2004), *Knowledge Management* (Natali & Falbo, 2002), *Organizational Modeling* (Cota & Menezes & Falbo, 2004), *Steel Factoring* (Mian et. al, 2002), among others. Moreover, the domain engineering methodology proposed has been used as a foundation for the construction of an ontology editor (Mian & Falbo, 2002) and a domain-oriented software engineering environment (Mian & Falbo, 2003). An extension of this approach for enterprise engineering has been proposed as part of the AGILA project (Caplinkas, 2003).

LINGO was designed with the specific objective of achieving a positive trade-off between expression power of the language and the ability to facilitate bridging the gap between the conceptual and implementation levels (a preoccupation that also seem to be present in Peter Chen's original proposal for ER diagrams). The language succeeds in offering abstractions that conform to the object-oriented paradigm, which, hence, enable a systematic translation approach. Moreover, it contains some important ontological distinctions, for example, the distinction between sortal and non-sortal concepts (see chapter 4 of this work). Nonetheless, its purely extensional semantics and ontological incompleteness (in the technical sense proposed in chapter 2) make it inappropriate as a general conceptual modeling language.

An important point that should be emphasized is the difference in the senses of the word ontology used by the information systems and domain engineering communities. In information systems, the term ontology has been used in ways that conform to its definitions in philosophy (in both senses D1 and D2). As a system of categories, an ontology is independent of language: Aristotle's ontology is the same whether it is represented in English, Greek or First-Order Logic. In contrast, in most of other areas of computer science (including domain engineering and *artificial intelligence*), the term ontology is, in general, used as a concrete engineering artifact designed for a specific purpose. In section 3.4, we provide a precise account for this latter use of the term and elaborate on its relation to conceptualization and language as discussed in chapter 2.

Finally, as a concrete artifact, an ontology should be constructed in a systematic process analogous to those of traditional software engineering. An ontological engineering process model typically comprises activities such as: *Purpose Identification and Requirements Specification, Ontology Capture, Formalization, Reuse and Integration, Evaluation and Documentation* (Falbo & Guizzardi & Duarte, 2002). For a more elaborated discussion on ontological engineering methodologies one should refer to (Gruninger & Fox, 1994b; Falbo & Menezes & Rocha, 1998; Fernández-López et. al, 1999; Gomez-Perez & Corcho & Fernandez-Lopez, 2002; Devedžić, 2002).

3.2.3 Ontology and Artificial Intelligence

The work of (Clancy, 1993) was of great importance for laying the groundwork for the establishment of ontology in artificial intelligence. In the tradition of AI, up to that moment, knowledge in artificial systems used to be defined in a strictly *functional* way aiming to incorporate in a knowledge base the steps that domain experts typically use in the solution of a given problem. Clancy proposed a shift in such a perspective, arguing that *"the primary concern of knowledge engineering is modeling systems in the world, not replicating how people think"* (ibid.). Clancy is a proponent of the *modeling view* of knowledge acquisition, according to which a knowledge base is not a repository of knowledge extracted from an expert's mind (as in the so-called *transfer view*), but it refers to an objective reality that is much more related to the classical notion of truth intended as a correspondence to the real-world[10]. More exactly, the modeling activity must establish a

[10] In the correspondence theory of the truth advocated by philosophers such as (Russel, 1918) and (Wittgenstein, 1922), the truth of a proposition is determined by its correspondence to the existence of an entity or fact in reality, the so-called *Truthmaker* of a proposition (Mulligan & Simons & Smith, 1984). Here, by objective reality, we mean a task-

correspondence between a knowledge base and two separate subsystems: the behavior of the intelligent system (i.e., the problem-solving expertise) and its environment (the problem domain) (Guarino, 1995).

Guarino strongly defends the view that the modeling of domain knowledge should be pursued in a way that is as independent as possible of the problem-solving task. The argument is based on two important points:

1. If the represented knowledge is not considered as *part* of the objective reality of the domain, the very basic assumptions of the modeling view are contradicted: if a domain theory does not describe (partially) *"an inherent structure in the domain"*, what is it supposed to represent? Arguably, the agent's mind, which was exactly what the modeling view aimed to avoid (Guarino, 1995, p.2). This reasoning can also be applied when directed to domain models and schemas used in domain engineering and information systems, respectively. How can we make systems with different conceptual models but overlapping semantics work together, if not by referring to the common world to which they all relate?

2. Knowledge acquisition (domain analysis, requirements engineering, etc.) is a notoriously expensive process and, hence, reuse of domain knowledge and domain representations should be maximized across different applications. With such a perspective, knowledge specifications can acquire a value *per se*, and as much as we approximate to the truth as classically conceived the potential for reuse should increase. As put in (Smith, 1995), truth in classical sense is a sort of infinite reusability.

For these reasons, it becomes clear how the study of ontology (in sense D1) can be of great benefit to the knowledge-construction process of intelligent systems, which is a position that finds strong support in (Guarino, 1995, 1997, 1998; Smith, 2004). Nonetheless, as in the case of information systems and domain engineering, the initial approaches to tackle domain modeling in AI suffered from a relative narrow perspective, concentrating on the immediate needs of the AI practice, and refusing to take into account the philosophical achievements coming from the study of common-sense reality (Guarino, 1995).

Based on these considerations, in (Guarino, 1998), the author proposes a classification of ontology kinds based on their level of dependence on a particular task or point of view:

---

independent conceptualization that is *truthful to reality* in the sense discussed in depth in section 3.4.1.

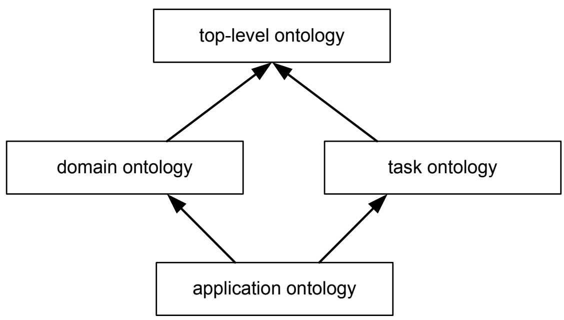
– *Top-level ontologies* describe very general concepts like space, time, matter, object, event, action, etc., which are independent of a particular problem or domain;

– *Domain ontologies* and *task ontologies* describe, respectively, the vocabulary related to a generic domain (like medicine, or automobiles) or a generic task or activity (like diagnosing or selling), by specializing the terms introduced in a top-level ontology;

– *Application ontologies* describe concepts that depend both on a particular domain and task, and often combine specializations of *both* the corresponding domain and task ontologies. These concepts often correspond to *roles* played by domain entities while performing a certain task, like *replaceable unit* or *spare component*.

In this definition, an *application ontology* is not considered as a synonym to a *knowledge base*. An ontology can be considered as a *particular* knowledge base, describing facts assumed, by a community of users, to hold *necessarily*, in virtue of the agreed-upon meaning of the vocabulary used. A generic knowledge base, instead, may also describe facts and assertions related to particular state of affairs or particular epistemic state. Consequently, within a generic knowledge base two components can be distinguished: the ontology (containing situation-independent information) and the "core" knowledge base (containing situation-dependent information) (Guarino & Giaretta, 1995). The idea of application ontologies is analogous to that of *specializations of knowledge frameworks* (Falbo et. al, 2002a,b; Falbo & Guizzardi & Duarte, 2002; Falbo & Menezes & Rocha, 1999). Specializations of knowledge frameworks are created to incorporate task and application knowledge that are dependent of a particular purpose and functionality that should be performed by a specific system or narrower class of systems.

Figure 3-9 A classification of different types of ontology. Arrows represent specialization relations (from Guarino, 1998)



The instances of all these different types of ontologies depicted in figure 3.9 are concrete engineering artifacts. Since the first time the word ontology was used in AI by Hayes (Hayes, 1978) and since the development of his naïve physics ontology of liguids (Hayes, 1985), a large amount of ontologies (mostly domain ontologies) have been developed. In the sequel, we briefly discuss some examples of domain ontologies that have been constructed over the years:

Engineering and Technical Applications

The *YMIR* ontology (Alberts, 1994) is a domain independent, sharable ontology for the formal representation of engineering design knowledge, based on systems theory. *EngMath* (Gruber & Olsen, 1994) is an ontology for mathematical modeling in engineering. It includes representations for scalar, vector, and tensor quantities, physical dimensions, units of measure, functions of quantities, and dimension quantities. *PHYSSYS* ontology (Borst, 1997) is an ontology for modeling, simulating and designing physical systems. It consists of three engineering ontologies that formalize the three viewpoints on physical devices, namely, system layout, physical process behavior and descriptive mathematical relations. *DORPA* (Varejão et. al, 2000) is an ontology for the design of reprographic machines developed in collaboration with the XEROX Palo Alto Research Center.

Enterprise Modeling and Manufacturing

The *TOVE* (*Toronto Virtual Enterprise*) ontology (Gruninger & Atefi & Fox, 2000), formalizes knowledge about production/communication processes, activities, causality, resources, quality and cost in business enterprises. Another example includes the *Enterprise Ontology* developed in a project supported by the UK's department of Trade and Industry and led by the AI Applications Institute at the University of Edinburgh (see Uschold et. al, 1998). It defines a collection of terms (e.g., activities and processes, organization, strategy, marketing) that are considered relevant for business enterprises. Once represented, these enterprise ontologies have been used as a basis for the development of methods and tools for enterprise modeling. The *Process Specification Language* (*PSL*) (Schlenoff et al., 2000; Ciocoiu & Gruninger, 2000) defines a neutral representation for manufacturing processes, which can be used to integrate modeling languages and tools that are used throughout the life cycle of a product, from early design of manufacturing process, through process planning, validation, production scheduling and control.

Chemistry, Biology and Ceramic Materials

CHEMICALS (Fernández-López et. al, 1999) is an ontology that contains knowledge within the domain of chemical elements and crystalline structures. The *Gene Ontology* (GO)[11] project is a collaborative effort to address the need for consistent descriptions of gene products among several of the world's largest repositories for plant, animal and microbial genomes. It is composed of three structured, controlled vocabularies that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. The *Fishery Core Ontology* (Gangemi et. al, 2002) was developed to support semantic interoperability among existing fishery information systems. Other examples include the *PLINIUS* (van der Vet & Mars, 1994) ontology of ceramic materials and the *ontology of pure substances* (van der Vet & Mars, 1995).

Medicine

The *GALEN* project aims at developing a terminological server for medical concepts (Rector at. al, 1995). It is based on a semantically sound model of clinical terminology: the *GALEN Coding reference* (*CORE*) *model*. This model comprises elementary clinical concepts (e.g., fracture, bone, and humerus), and relationships (e.g., *fractures can occur in bones*) that control how these may be combined. Moreover, it includes complex concepts (such as *fracture of the left humerus*) composed from simpler ones. The *ON9* is a large-scale ontology library for medical terminology developed in the context of the *ONIONS* (*ONtological Integration of Naïve Sources*) project (Gangemi & Pisanelli & Steve, 1999). One of terminological sources that are formalized and integrated in ON9 is the *UMLS* Metathesaurus developed by the American National Library of Medicine (Pisanelli & Gangemi & Steve, 1998). Additionally, (Pisanelli et. al, 2003) describes an ontology representing the concepts involved in evidence-based medicine and meta-analysis

Law

The *Core Legal Ontology* (*CLO*) (Gangemi et. al, 2003) is used to organize juridical concepts and to represent the assessment of legal regulatory compliance across different legal systems or between norms and cases. In (Sagri & Tiscornia & Gangemi, 2004) CLO is used as a foundation for the development of *JurWordNet* (a semantic lexicon to support legal information

[11] http://www.geneontology.org/

searching) (Sagri, 2003), and for the representation of click-on licenses in the field of Intellectual Property Right (IPR).

3.2.4 Ontology and the Semantic Web

The World Wide Web has been made possible due to the availability of a set of well established standards that guarantee interoperability at various levels, like, for instance, the transport and application level (with the TCP[12] and HTTP[13] standards, respectively), and the presentation level (with the HTML standard[14]). After Tim Berners-Lee's seminal paper (Berners-Lee, 1989-1990), the web started to be collectively created as a hypermedia network of information nodes. Along these years, it suffered a transformation from a medium for *information exchange* to a medium for *service deployment*. Nonetheless, its current structure is still directed for human processing and interpretation. Regardless if the user is booking a flight, searching for news or analysing catalogues of Volvo parts made by different manufactures, the interpretation of the content of each web node relies totally on the user's ability to give real-world semantics to symbols written in a certain language.

The next developmental step for the web has been characterized by the so-called *Semantic Web* vision (Berners-Lee, Hendler, Lassila, 2001), which stipulates a change in the web from being *machine-readable* (but only human-understandable) to *machine-understandable*. By machine-understandability, Berners-Lee and colleagues mean the following: web resources (information nodes and computational services) are annotated by meta-data written in a formal knowledge representation language, i.e., in a language with precisely defined formal semantics and with associated highly optimized inference procedures and engines. As a consequence, techniques developed by the knowledge representation community over the years could be exploited in the development of intelligent services, such as intelligent search engines (Mayfield & Finin, 2003), information brokering and filtering (Klien et. al, 2004; Vögele & Hübner & Schuster, 2003), web service annotation and automatic service composition/orchestration (Majithia & Walker & Gray, 2004; Paolucci & Sycara & Kawamura, 2003), knowledge management architectures (Davies & Fensel & van Harmelen, 2003; Guizzardi & Aroyo & Wagner, 2003), among many others.

In this scenario, ontologies are expected to play a fundamental role by interweaving human understanding of symbols with machine processability. The idea is that, firstly, ontologies representing shared conceptualizations of

[12] Transmission Control Protocol (http://www.faqs.org/rfcs/rfc793.html)
[13] HyperText Transfer Protocol (http://www.w3.org/Protocols/)
[14] http://www.w3.org/MarkUp/

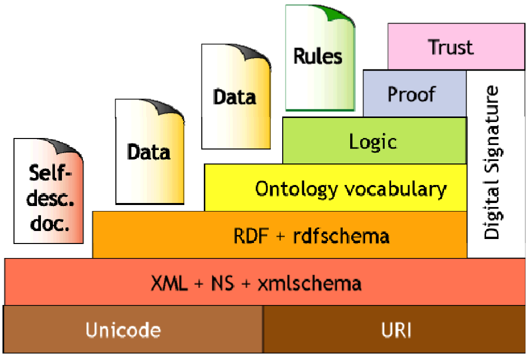
reality are constructed and specified in a *machine-understandable* language. Then, as formal specifications, they can be used as semantic domains for the definition of formal and real-world semantics for syntactic symbols present in web resources.

In the sequel, we discussed the most relevant semantic web technologies from the point of view of domain representation.

3.2.4.1 An architecture for the Semantic Web

One of the main architectural premises of the Semantic web is the stack of languages, often drawn in a figure firstly presented by Berners-Lee in his XML 2000 keynote address[15] and replicated in figure 3.10. In the sequel we briefly discuss some layers in this stack leading up to the ontology representation languages. For a more detailed discussion one should refer to (Koivunen & Miller, 2001).

Figure 3-10 The Semantic Web Layered Architecture



In the bottom language layer of the stack we have the *eXtensible Markup Language* (*XML*)[16]. As opposed to HTML documents, which have a predefined syntactic structure, XML was designed as a markup-language for arbitrary document structures. By using *XML Schema Definitions* (*XSD*'s), one can define a grammar (an abstract syntax definition) for a class of XML documents, defining vocabulary and syntactic rules that are specific to a class of applications. For this reason, it can be used as serialization syntax for other markup languages. For example, the *Synchronized Multimedia Integration Language* (*SMIL*)[17] is syntactically just a particular XSD.

For a processing application, any symbol in a HTML document that is not a member of the predefined markup primitives is seen as a meaningless

[15] Available at http://www.w3.org/2000/talks/1206-xml2k-tbl/.
[16] http://www.w3.org/XML/
[17] http://www.w3.org/AudioVideo/

string of characters. In contrast, XML provides some structuring and a standard way for defining the abstract syntax for a class of documents that can be shared by other applications. Nonetheless, a XSD only specifies syntactic rules, and any intended semantics for the syntactic elements is totally left outside the realm of the XML specification. Hence, for a processing application, the tags used to structure a XML document are as devoid of meaning as a variable label in a Java program, regardless of the natural language term that we give to it.

On top of the XML layer, the W3C[18] defines the *Resource Description Framework* (*RDF*)[19], which standardizes the definition and use of meta-data descriptions about web resources. A web resource, from a RDF point of view, is anything that can be given a *Uniform Resource Identifier* (*URI*)[20], such as a web page, a computer device, a multimedia file, etc. Examples of applications of metadata descriptions are:

- Website map (the metadata structure can describe the content of the web pages as well as their interrelationships);
- Descriptions of resources privacy policies;
- Description of advisory rating for multimedia content;
- Description of device capabilities;
- Description of user's preferences and characteristics;
- Digital Signatures;
- Content classification systems (in particular, its ability to describe taxonomic relationships can be exploited, for example, for books, films, articles, etc.).

RDF data is made up of statements, where each statement expresses the values of the properties of a resource. The basic RDF data model consists of an object-attribute-value triple commonly written as *Attribute(Object,Value)*. For example, in figure 3.11, we present a RDF description in the notation of labelled directed graphs. In this figure, the property title (predicate) relates the http://www.example.org/ resource (subject) with a literal (object).

---

[18] W3C stands for *World Wide Web Consortium* which is the entity responsible for the standardization of WWW-related technologies (http://www.w3.org/).

[19] http://www.w3.org/RDF/

[20] http://www.gbiv.com/protocols/uri/rfc/rfc3986.html/

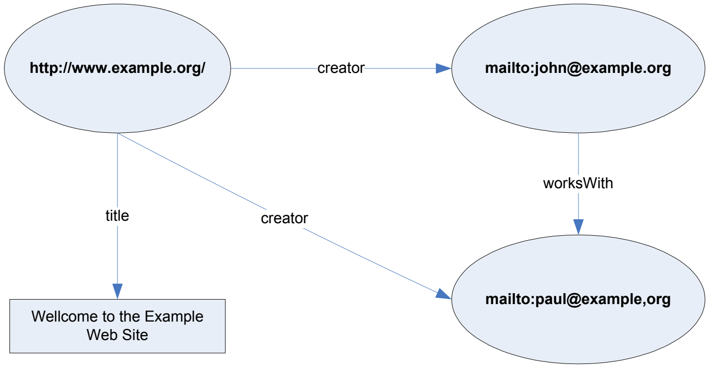


Figure 3-11 Example of a simple RDF description

In principle, RDF could also be used to represent any ordinary data model. However, it comprises only a minimalist model containing primitives such as *description*, *resource*, *property*, and does not provide any means for the definition of the terms used in resource annotation, such as *creator*, *workswith* and *title* in figure 3.11. For this reason, a schema language named *RDF Schema* (*RDFS*)[21] has been proposed. In contrast with XML, the RDF/RDFS model is able to define the semantics of its specifications (in case, for instance, a semantics such as the one proposed in (Pan & Horrocks, 2003)[22] is adopted).

In the semantic web community, RDFS is regarded as a simple ontology representation language, since it incorporates the simplest parts of frame-based languages such as *OKBC* (Grosso, 1999) (i.e., *classes, properties, domain and range restrictions, instance-of, subclass-of* and *subproperty-of* relationships). Nonetheless, there are many useful types of formulas that cannot be expressed in the language. A few examples are:

- In a RDFS specification, one cannot state that two subconcepts of a common concept form a partition, i.e., that they are disjoint and complete;
- although restrictions can be posed on the type of resources that form the domain and range of a property, cardinality constraints cannot be represented in a RDFS specification.

In order to overcome the deficiencies of RDFS as an ontology representation language, two different languages were proposed, offering a

[21] http://www.w3.org/TR/rdf-schema/

[22] An interesting aspect of the non-standard model theoretical semantics defined in this paper is its ability to solve some ambiguity problems with the RDF model as well as its compatibility with layering of a description logic based ontology modelling language (e.g. OWL) on top of it.

richer set of modelling primitives: *DAML-ONT* (McGuiness et al., 2002a) and *OIL* (Fensel et. al, 2001). Later, these languages converged in a single proposal named *DAML+OIL* (McGuiness et al., 2002b), which was further refined to become the W3C recommendation *OWL* (*Ontology Web Language*) (Bechhofer et. al, 2004). These (logical layer) ontology modelling languages have been carefully designed for the best possible trade-off between expressiveness and computational efficiency, *the latter property having precedence*. The language inherits from a specific type of family of *description logics* (*SHIQ*) (Baader & Horrocks & Sattler, 2003; Horrocks & Patel-Schneider & van Harmelen, 2003) its formal semantics and reasoning support, which ensure logical completeness, correctness and efficiency.

Another design requirement for the language was related to the need for maintaining compatibility with other languages in the stack of figure 3.10. Consequently, OWL has been provided with both XML and RDF serializations. On one hand, an OWL specification is a valid XML document whose syntax is property defined in XSD. On the other hand, OWL reuses many of the RDFS primitives (e.g., *class, domain, range, property*), which makes its specifications partially available to RDFS-only software.

Finally, the language was also designed aiming at being intuitive to the human user. For this reason its modelling primitives are based on the popular paradigm of frame-based and object-based languages. A related (but nonetheless diverse) point, which has not been considered by the designers of the language, is the pragmatic effectiveness of the language's concrete syntax. Although mainly targeted for machine-processing (as opposed to human problem-solving and communication), OWL specifications are, in general, created and manipulated by humans. In order to tackle this problem some proposals towards a UML syntax for OWL have been pursued (see, for example, Baclawski et. al, 2001).

3.2.4.2 Context-Aware Services: An application Scenario

An application area in which Semantic web ontologies have been employed successfully is the subfield of ubiquitous computing named context-aware computing (Ríos, et. al, 2003; Chen & Finin & Joshi, 2003; Strang & Linnhoff-Popien & Korbinian, 2003). Context-aware computing is a new computing paradigm that has brought the possibility of exploring the dynamic context of the user in order to provide more adaptable, complex and personalized services. A context is a situation involving the user or its environment, which can be considered of interest for an application. Examples of contexts include:

– Physical contexts (such as location and time, etc.);

– Environmental contexts (weather, altitude, velocity, humidity, light, etc.);
– Informational contexts (stock quotes, sports scores, etc.);
– Personal contexts (health, mood, schedule, activity, etc.);
– Social contexts (group activity, social relationships, vicinity of people, etc.);
– Application contexts (email received, websites visited, etc.);
– System contexts (network traffic, status of printers, device battery charge load, etc.).

In (Ríos et. al, 2003; Ríos, 2003), ontologies are used for improving the modeling and handling of contextual information in a context-aware services platform named *WASP* (*Web Architecture for Services Platform*) (Costa et al., 2004a). This approach, illustrated in figure 3.12, is summarized in the sequel.

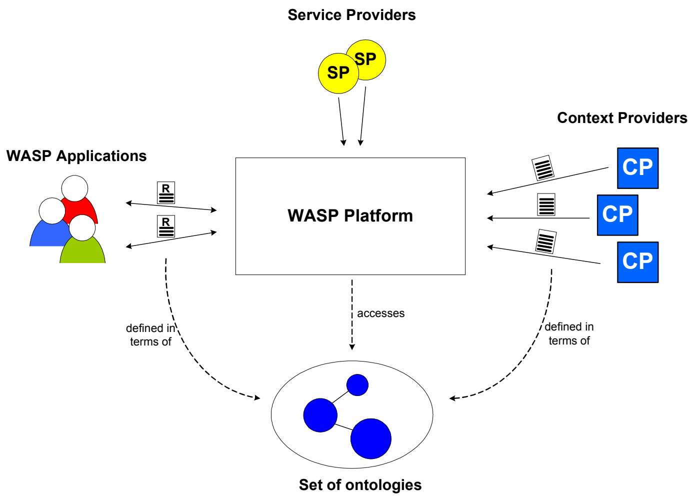


Figure 3-12 The ontology-based version of the WASP platform (from Ríos, 2003)

The objective of the WASP platform is to serve as broker between four different types of entities, namely, *context providers*, *service providers*, *WASP applications* and *semantic web ontologies.*

Context Providers (CP) are responsible for making contextual information available to the other entities interaction with the platform. Examples of Context Providers include physical sensors and software agents. The architecture must be open to new kinds of third-party providers. These providers may supply information using different protocols and/or languages using different syntaxes. An example of a message sent to the platform by a CP would be: *"The visitor John Smith is inside the Chiaroscuro* *gallery of the Rijksmuseum"*. This message can be considered a description relating the resource *John Smith* (identified via a URI as a record in some database) to another resource (the *Chiaroscuro* gallery) via the property *inside*.

The platform is open to third-party *Service Providers (SP)* interested in offering services to the users of the platform. Depending on the user requirements and context, the platform must support discovery and publishing of services. An example of a simple service is an instant messenger/SMS service provided by a third party SP.

Another objective of the platform is to provide to WASP applications facilities for reacting to their dynamic environment. This is accomplished by allowing the applications to describe to the platform what actions should be taken (e.g., execute a service) in case a situation associated with a context holds. An example of a WASP subscription would be *"**if** a visitor enters the Chiaroscuro gallery in the Rijksmuseum **then** he should be notified via a SMS service about the free Rembrandt Calendar Gift"*, or *"**if** a visitor enters a museum, cinema or theatre **then** a silent mode instruction must be sent to his mobile phone via a mobile phone control service"*.

Ontologies in this case are logical theories that play the role of semantic domains. In this way, context providers and WASP applications can interoperate because they use the same set of interrelated ontologies to define the semantics of both contextual information messages and application subscriptions. For instance, the meaning of *visitor*, *museum*, *museum gallery*, and of the predicate *inside* can be defined in terms of: (a) more general ontologies that, for instance, define the properties of physical objects and places; (b) domain-specific ontologies that define characteristics of museums, galleries and works of art. By reasoning on these ontologies, the platform can detect inconsistencies (e.g., a person cannot be in two disjoint places at the same time), and exhibit a more intelligent behavior, by deriving knowledge from the factual knowledge available (e.g., a visitor is in *a place if he is in any part of that place).*

In (Ríos, 2003), some applications of context-aware services using the ontology-based version of the WASP platform are presented. These include: (i) an *airport information application* offering services related to flight information; (ii) an *event advisors* which notifies users about upcoming events that match their personal interests; (iii) *a friend finder* application that notifies an user when he is close to or inside the same place as one of his friends. Both (ii) and (iii) make use of a *Tourism Ontology*, depicted in figure 3.13[23].

[23] This picture is an exactly copy from (Rios, 2003), in which a case tool is used to generate a UML syntax for the original OWL representation. The notation *p:X* in this specification symbolizes that X is a *property* in the OWL sense of the term.

The sense of the term ontology adopted in the Semantic web vision is similar to the one adopted by the AI and domain engineering communities, i.e., ontology as an engineering artifact consisting of a formal structure of concepts and relations among concepts, and a set of axioms that both constrains the interpretation of this structure and affords the derivation of knowledge from the factual knowledge represented in the structure.

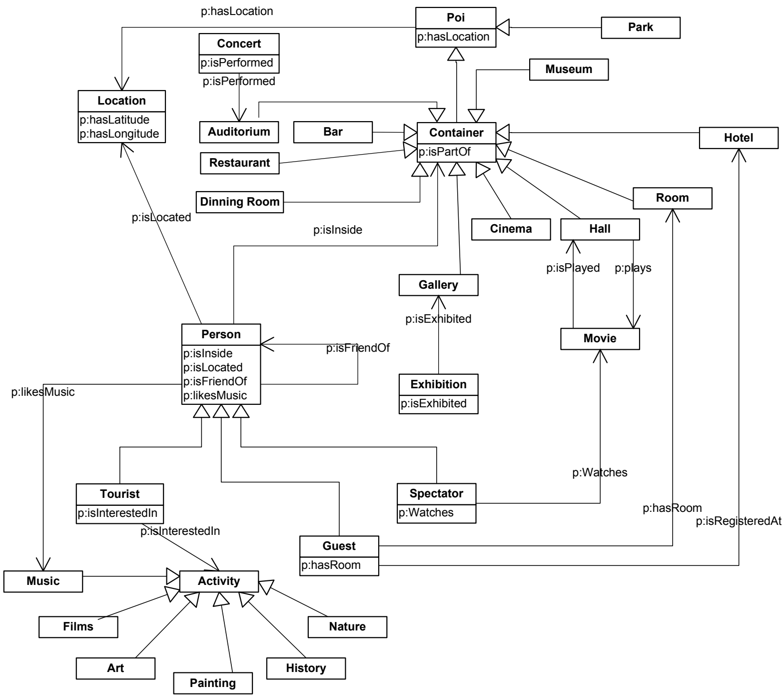


Figure 3-13 A Tourism Ontology referenced by context-aware applications in the ontology-based version of the WASP platform (exactly copy from Ríos, 2003)

3.3 Terminological Clarifications and Formal Characterizations

In the information systems community, the term ontology has a clear meaning, which is akin to the original definition (D2) given in philosophy. In the works of (Wand & Storey & Weber, 1999; Opdahl & Henderson-Sellers, 2001), for example, ontology is clearly used as a system of categories, which is independent of language and state of affairs. In contrast, in the majority of the works in other areas of computer science, the term is used to denote a class of artifacts with a spectrum of divergent characteristics. To illustrate this terminological confusion, figure 3.14 from

(Smith & Welty, 2001), depicts a wide range of different artifacts that have been termed ontologies in the literature. One can observe that these specifications range from simple catalogs containing, for instance, the products that a company sells, to a lexicon of terms with natural language definitions (thesaurus), to formal logical theories.

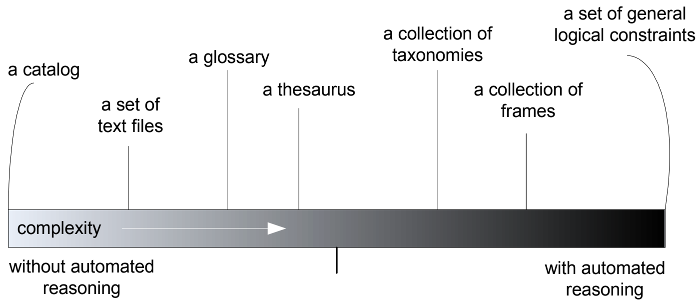


Figure 3-14 Different sorts of specifications classified as ontologies in the computer science literature (from Smith & Welty, 2001)

In (Guarino & Giaretta, 1995), the authors discuss seven different interpretations of the word ontology as used in AI and, after a careful analysis, restrict its possible use to three sound interpretations:

(a). ontology as a representation of a conceptual system that is characterized by specific logical properties (special type of logical theory containing only necessarily true formulas);
(b). ontology as a specification of an ontological commitment;
(c). ontology as a synonym of conceptualization.

Many of the specifications accounted in the classification of figure 3.14 are indeed ontologies in sense (a). However, another share of them complies only with a sense of the term, which was deemed in Guarino & Giaretta's analysis insufficient to qualify as an ontology, namely, ontology as a representation of a conceptual system that is *characterized by specific purposes*. These different senses, as well as their interrelations are discussed in depth in the remaining of this section, in which we establish a precise definition for these terms as they are used in the remaining of this work.

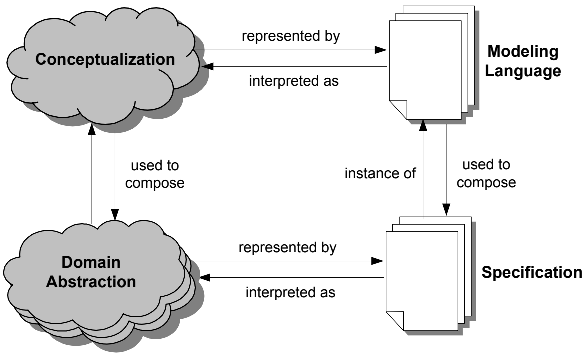
3.3.1 Ontology and Conceptualization

One of the most cited definitions for the term ontology in the Computer Science literature is the one given by Tom Gruber in (Gruber, 1995): *"An Ontology is a explicit representation of a conceptualization"*.

As discussed in (Guarino & Giaretta, 1995; Guarino, 1995), the way this definition was originally proposed by Gruber, it is a problematic one as it relies on a notion of conceptualization that is purely extensional, namely that of (Genesereth & Nilsson, 1987). Moreover, as argued in (Smith & Welty, 2001), it is too open for interpretations as all those different types artifacts in figure 3.14 comply with this definition. Nonetheless, it captures an intuitive idea, which remains true for the sense of ontology which is employed in the great majority of the works referring to ontology in artificial intelligence, domain engineering and the semantic web.

In chapter 2, we have used figure 3.15 to illustrate the relationships between a conceptualization, a language, a domain abstraction[24] (representing part of some state of affairs in reality articulated according to the conceptualization) and a specification in the language which represents this domain abstraction.

Figure 3-15 Relations between conceptualization, domain abstraction, modeling language and specification



In (Guarino, 1998), the author points out that the sense in which ontology is used in philosophy as a system of categories accounting for a certain vision of the world is akin to what we name a *conceptualization* in figure 3.15. In the philosophical reading, an ontology is independent of language and of particular epistemic state or state of affairs. On one hand, the same ontology could be represented in different languages, For example, the words *orange*, *arancia*, *laranja* and *sinasappel* refer to exactly the same ontological entity, namely the natural kind denoted by these terms. On the other hand, an ontology is neutral w.r.t. the actual existence of a particular orange *a*, but in contrast, it stipulates that if there is a situation in which *a* exists as an orange then *a* also exists as a fruit in this situation. Finally, although we can have an epistemic state of an agent expressing his uncertainty whether *a* is an *orange OR a lemon*, there is no such a thing as an

[24] In chapter 2, we have used the term *model* instead of *domain abstraction* since it is the most common term in conceptual modeling. In this session, exclusively, we adopt the latter in order to avoid confusion with the term (logical) model as used in logics and tarskian semantics. The term logical model here, in turn, bears no relation to the term *logical model* in databases as used in section 3.2.1 of this chapter.

ontological optional property, or something in reality that has the property of being *either an orange or a lemon*.

However, in order to reason about the characteristics of a conceptualization, we must have it captured in some concrete form. In this section, we use a minimum concrete representation of a conceptualization, which is still precise and useful for the purpose of the discussion. The idea is to characterize a conceptualization as an *intensional structure* which, hence, encompasses all state of affairs considered, and which is independent of a particular language vocabulary. This notion of a conceptualization has been proposed in (Guarino, 1998) and can be formally defined as follows:

Definition 3.1 (conceptualization): a conceptualization $\mathcal{C}$ is an intensional structure $\langle \mathcal{W}, \mathcal{D}, \Re \rangle$ such that $\mathcal{W}$ is a (non-empty) set of possible worlds, $\mathcal{D}$ is the domain of individuals and $\Re$ is the set of n-ary relations (concepts) that are considered in $\mathcal{C}$. The elements $\rho \in \Re$ are intensional (or conceptual) relations with signatures such as $\rho^n : \mathcal{W} \rightarrow \wp(\mathcal{D}^n)$, so that each n-ary relation is a function from possible worlds to n-tuples of individuals in the domain.

■

For instance, we can have ρ accounting for the meaning of the natural kind apple. In this case, the meaning of apple is captured by the intensional function ρ, which refers to all instances of apples in every possible world.

Definition 3.2 (intended world structure): For every world $w \in \mathcal{W}$, according to $\mathcal{C}$ we have an *intended world structure* $\mathcal{S}_w\mathcal{C}$ as a structure $\langle \mathcal{D}, \mathcal{R}_w\mathcal{C} \rangle$ such that $\mathcal{R}_w\mathcal{C} = \{\rho(w) \mid \rho \in \Re\}$.

■

More informally, we can say that every intended world structure $\mathcal{S}_w\mathcal{C}$ is the characterization of some state of affairs in world w deemed admissible by conceptualization $\mathcal{C}$. From a complementary perspective, $\mathcal{C}$ defines all the admissible state of affairs in that domain, which will be represented by the set $\mathcal{S}_\mathcal{C} = \{\mathcal{S}_w\mathcal{C} \mid w \in \mathcal{W}\}$.

Let us consider now a language $\mathcal{L}$ with a vocabulary $\mathcal{V}$ that contains terms to represent every concept in $\mathcal{C}$.

Definition 3.3 (logical model): A logical model for $\mathcal{L}$ can be defined as a structure $\langle \mathcal{S}, \mathcal{I} \rangle$: $\mathcal{S}$ is the structure $\langle \mathcal{D}, \mathcal{R} \rangle$, where $\mathcal{D}$ is the domain of individuals and $\mathcal{R}$ is a set of extensional relations; $I{:}\mathcal{V}\rightarrow\mathcal{D} \cup \mathcal{R}$ is an interpretation function assigning elements of $\mathcal{D}$ to constant symbols in $\mathcal{V}$, and elements of $\mathcal{R}$ to predicate symbols of $\mathcal{V}$. A model, such as this one, fixes a particular extensional interpretation of language $\mathcal{L}$.

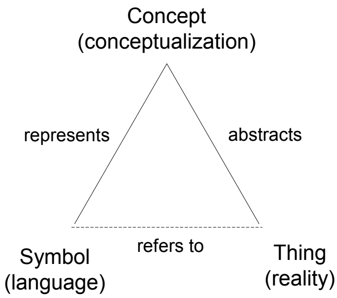
■

Definition 3.4 (intensional interpretation): Analogously, we can define an intensional interpretation by means of the structure $\langle \mathcal{C},\Im \rangle$, where $\mathcal{C} = \langle \mathcal{W},\mathcal{D},\Re \rangle$ is a conceptualization and $\Im{:}\mathcal{V} \rightarrow \mathcal{D} \cup \Re$ is an intensional interpretation function which assigns elements of $\mathcal{D}$ to constant symbols in $\mathcal{V}$, and elements of $\Re$ to predicate symbols in $\mathcal{V}$.

■

In (Guarino, 1998), this intensional structure is named the *ontological commitment* of language $\mathcal{L}$ to a conceptualization $\mathcal{C}$. We therefore consider this intensional relation as corresponding to the *represents* relation depicted in Ullmann's triangle in figure 3.16 depicted below (see discussion in section 2.1.4).

Figure 3-16 Ullmann's Triangle: the relations between a thing in reality, its conceptualization, and a symbolic representation of this conceptualization



Definition 3.5 (ontological commitment): Given a logical language $\mathcal{L}$ with vocabulary $\mathcal{V}$, an *ontological commitment* $\mathcal{K} = \langle \mathcal{C},\Im \rangle$, a model $\langle \mathcal{S},I \rangle$ of $\mathcal{L}$ is said to be compatible with $\mathcal{K}$ if: (i) $\mathcal{S} \in \mathcal{S}_{\mathcal{C}}$; (ii) for each constant c, $I(c) = \Im(c)$; (iii) there exists a world w such that for every predicate symbol p, I maps such a predicate to an admissible extension of $\Im(p)$, i.e. there is a conceptual relation ρ such that $\Im(p) = \rho$ and $\rho(w) = I(p)$. In accordance with (Guarino, 1998), the set $I_k(\mathcal{L})$ of all models of $\mathcal{L}$ that are compatible with $\mathcal{K}$ is named the set of *intended models* of $\mathcal{L}$ according to $\mathcal{K}$.

■

Definition 3.6 (logical rendering): Given a specification $\mathcal{X}$ in a specification language $\mathcal{L}$, we define as the logical rendering of $\mathcal{X}$, the logical theory $\mathcal{T}$ that is the first-order logic description of that specification (Ciocoiu & Nau, 2000).

■

In order to exemplify these ideas let us take the example of a very simple conceptualization C such that $\mathcal{W} = \{w,w'\}$, $\mathcal{D} = \{a,b,c\}$ and $\Re = \{person, father\}$. Moreover, we have that $person(w) = \{a,b,c\}$, $father(w) = \{a\}$, $person(w') = \{a,b,c\}$ and $father(w') = \{a,b\}$. This conceptualization accepts two possible state of affairs, which are represented by the world structures $S_wC = \{\{a,b,c\}, \{\{a,b,c\},\{a\}\}$ and $S_{w'}C = \{\{a,b,c\}, \{\{a,b,c\},\{a,b\}\}$. Now, let us take a language $\mathcal{L}$ whose vocabulary is comprised of the terms `Person` and `Father` with an underlying metamodel specification that poses no restrictions on the use of these primitives. In other words, the metamodel specification of $\mathcal{L}$ has the following logical rendering $\mathcal{T}_1$:

($\mathcal{T}_1$)

```
∃x Person(x)
∃x Father(x)
```

Clearly, we can produce a logical model of $\mathcal{L}$ (i.e., an interpretation that validates the logical rendering of $\mathcal{L}$) but that is not an intended world structure of C. For instance, the model $\mathcal{D}'=\{a,b,c\}$, $person = \{a,b\}$, $father = \{c\}$, and I(`Person`) = *person* and I(`Father`) = *father*. This means that we can produce a specification using $\mathcal{L}$ whose model is not an *intended model* according to C.

However, we can update the metamodel specification of language $\mathcal{L}$ by adding the following axiom:

($\mathcal{T}_2$)

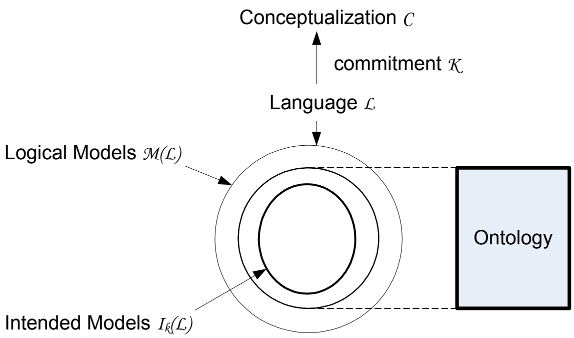
```
∃x Person(x)
∃x Father(x)
∀x Father(x)→ Person(x)
```

Contrary to $\mathcal{L}$, the resulting language $\mathcal{L}'$ with the amended underlying metamodel specification $\mathcal{T}_2$ has the desirable property that all its valid specifications have logical models that are *intended world structures of* C.

We can summarize the discussion so far as follows. A domain conceptualization C can be understood as describing the set of all possible state of affairs, which are considered admissible in a given universe of discourse U. Let $\mathcal{V}$ be a vocabulary whose terms directly correspond to the intensional relations in $\mathcal{C}$. Now, let $\mathcal{X}$ be a *conceptual specification* (i.e., a concrete representation) of universe of discourse U and let $\mathcal{T}_{\mathcal{X}}$ be a logical rendering of $\mathcal{X}$, such that its axiomatization constrains the possible interpretations of the members of $\mathcal{V}$. We call $\mathcal{X}$ (and $\mathcal{T}_{\mathcal{X}}$) an *ideal ontology* of U according to $\mathcal{C}$ iff the logical models of $\mathcal{T}_{\mathcal{X}}$ describe all and only state of affairs which are admitted by $\mathcal{C}$.

The relationships between language vocabulary, conceptualization, ontological commitment and ontology are depicted in figure 3.17 below. This use of the term ontology is strongly related to the third sense (D3) in which the term is used in philosophy, i.e. as *"a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system"* (section 3.1).

Figure 3-17 Relations between language (vocabulary), conceptualization, ontological commitment and ontology



The logical theory ($\mathcal{T}_2$) described above is, thus, an example of an ontology for the person/father toy conceptualization. The same would apply to the models depicted in figures 2.17, 3.4, 3.5 and 3.13.

As pointed out in (Guarino, 1998), ontologies cannot always be ideal and, hence, a general definition for an (non-ideal) ontology must be given: *An ontology is a conceptual specification that describes knowledge about a domain in a manner that is independent of epistemic states and state of affairs. Moreover, it intends to constrain the possible interpretations of a language's vocabulary so that its logical models approximate as well as possible the set of intended world structures of a conceptualization* $\mathcal{C}$ *of that domain.*

According to criteria of accuracy, we can therefore give a precise account for the quality of a given ontology. Given an ontology $O_{\mathcal{L}}$ and an ideal ontology $O_{\mathcal{C}}$, the quality of $O_{\mathcal{L}}$ can be measured as the distance between the set of logical models of $O_{\mathcal{L}}$ and $O_{\mathcal{C}}$. In the best case, the two ontologies share the same set of logical models. In particular, if $O_{\mathcal{L}}$ is the specification of the ontological metamodel of modeling language $\mathcal{L}$, we can state that if $O_{\mathcal{L}}$ and O_{C} are isomorphic then they also share the same set of possible models. It is important to emphasize the relation between the possible models of $O_{\mathcal{L}}$ and the completeness of language $\mathcal{L}$ (in the technical sense discussed in chapter 2). There are two ways in which incompleteness can impact the quality of $O_{\mathcal{L}}$: firstly, if $O_{\mathcal{L}}$ (and thus $\mathcal{L}$) does not contain concepts to fully characterize a state of affairs, it is possible that the logical models of $O_{\mathcal{L}}$ describe situations that are present in several world structures of C. In this case, $O_{\mathcal{L}}$ is said to *weakly* characterize C (Guarino, 1998), since it cannot guarantee the reconstruction of the relation between worlds and extensional relations established by C; secondly, if the representation of a concept in $O_{\mathcal{L}}$ is underspecified, it will not contain the axiomatization necessary to exclude unintended logical models. As an example of the latter, we can mention the incompleteness of UML class diagrams w.r.t. classifiers and part-whole relations discussed in chapters 4 and 5 of this work, respectively. In summary, we can state that an ideal ontology O_{C} for a conceptualization C of universe of discourse U can be seen as the specification of the ontological metamodel for an ideal language to represent U according to C. For this reason, the adequacy of a language $\mathcal{L}$ to represent phenomena in U can be systematically evaluated by comparing $\mathcal{L}$'s metamodel specification with O_{C}.

By including the third axiom, ($\mathcal{T}_1$) is transformed into an ideal ontology ($\mathcal{T}_2$) of C. One question that comes to the mind is: How can one know that? In other words, how can we systematically design an ontology O that is a better characterization of C. There are two important points that should be called to attention. The first point concerns the language that is used in the representation of these specifications, namely, that of standard predicate calculus. The axiom added to ($\mathcal{T}_1$) to create ($\mathcal{T}_2$) represents a subsumption relation between Person and Father. Subsumption is a basic primitive in the group of the so-called epistemological languages (see 3.4.2), which includes languages such as EER (Elmasri & Weeldreyer & Hevner, 1985) and OWL. It is, in contrast, absent in ontological neutral logical languages such as predicate calculus. By using a language such as OWL to represent a conceptualization of this domain, a specification such as the one in figure 3.18 should be produced. In this model, the third axiom would be automatically included through the semantics of the metamodeling language. Therefore, if a suitable ontology modeling language is chosen, its primitives incorporate an axiomatization, such that the specifications (ontologies) produced using this language will better approximate the intended models of a conceptualization C. Additionally, as one can notice, subsumption is not a relation which is specific to the represented domain. In contrast, it is a *formal* relation that appears in several different universes of discourse. This feature is compatible with those of primitives that should figure in a general conceptual modeling language.

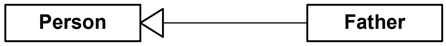


Figure 3-18 Example of a subsumption relation in UML

The second point that should be emphasized is related to the question: *how are the world structures that are admissible to $\mathcal{C}$ determined?* The rationale that we use to decide that are far from arbitrary, but motivated by the laws that govern the domain in reality. In (Bunge, 1977), the philosopher of science Mario Bunge defines the concepts of a *state space* of a thing[25], and a subset of it, which he names a *nomological state space*. The idea is that among all the (theoretically) possible states a thing can assume, only a subset of it is lawful and, thus, is actually possible. Additionally, he defends that the only really possible *facts* involving a thing are those that abide by laws, i.e., those delimited by the nomological state space of thing. As a generalization, if an actual state of affairs consists of facts (Armstrong, 1997), then the set of possible state of affairs is determined by a *domain nomological state space*. In sum, possibility is not by any means defined arbitrarily, but should be constrained by the set of laws that constitute reality. For example, it is law of the domain (in reality) that every Father is a Person. The specification ($\mathcal{T}_2$) is an ideal ontology for $\mathcal{C}$ because it includes the representation of this law of this domain via the subsumption relation between the corresponding representations of father and person. Conversely, if $\mathcal{C}$ included a world structure in which this law would be broken, the conceptualization itself would not be truthful to reality. To refer once more to Ullmann's triangle (figure 3.16), the relation between $\mathcal{C}$ and the *domain nomological state space* is that relation of *abstracts* between a conceptualization and reality.

Now, to raise the level of abstraction, we can also consider the existence of a meta-conceptualization $\mathcal{C}'$, which defines the set of all domain conceptualizations such as $\mathcal{C}$ that are truthful to reality. Our main objective is to define a general conceptual modeling language $\mathcal{L}'$ that can be used to produce domain ontologies such as $O_{\mathcal{C}}$, i.e., a language whose primitives include theories that help in the formal characterization of a domain-specific language $\mathcal{L}$, restricting its logical models to those deemed

[25] The word Thing is used by Bunge in a technical sense, which is synonymous to the notion of *substantial individual* defined in Chapter 6.

admissible by $\mathcal{C}$. In order to do this, we have to include in $\mathcal{L}'$ primitives that represent the laws that are used to define the nomological world space of meta-conceptualization $\mathcal{C}'$. In this case, these are the general laws that describe reality, and describing these laws is the very business of *formal ontology* in philosophy.

In summary, we defend that the ontology underlying a general conceptual modeling language $\mathcal{L}'$ should be a meta-ontology that describes a set of real-world categories that can be used to talk about reality. Likewise, the axiomatization of this foundational ontology must represent the laws that define that nomological world space of reality. This meta-ontology, when constructed using the theories developed by *formal ontology* in philosophy, is named a *foundational ontology*.

3.3.2 The Ontological Level

When a general conceptual modeling language (or ontology representation language) is constrained in such a way that its intended models are made explicit, it can be classified as belonging to the *ontological level*. This notion has been proposed by Nicola Guarino in (Guarino, 1994), in which he revisits Brachman's classification of knowledge representation formalisms (Brachman, 1979).

In Brachman's original proposal, the modeling primitives offered by knowledge representation formalisms are classified in four different levels, namely: *implementation*, *logical*, *conceptual* and *linguistic* levels.

In the logical level, we are concerned with the predicates necessary to represent the concepts of a domain and with evaluating the truth of these predicates for certain individuals. The basic primitives are propositions, predicates, functions and logical operators, which are extremely general and ontologically neutral. For instance, suppose we want to state that a red apple exists. In predicate calculus we would write down a logical formula such as (F_1) $\exists x(apple(x) \wedge red(x))$.

Although this formula has a precise semantics, the real-world interpretation of a predicate occurring in it is completely arbitrary, since one could use it to represent a property of a thing, the kind the thing belongs to, a role played by the thing, among other possibilities. In this example, the predicates apple and red are put in the same logical footing, regardless of the nature of the concept they represent and the importance of this concept for the qualification of predicated individual. Logical level languages are neutral w.r.t. ontological commitments and it is exactly this neutrality that makes logic interesting to be used in the development of scientific theories. However, it should be used with care and not directly in the development of ontologies, since one can write perfectly correct logical formulas, but which are devoid of ontological interpretation. For example, the entailment relation has no ontic correlation. Moreover, while one can negate a predicate or construct a formula by a disjuction of two predicates, in reality, there are neither negative nor alternative entities (Bunge, 1977).

In order to improve the "flatness" of logical languages, Brachman proposes the introduction of an *epistemological level* on top of it, i.e., between the logical and conceptual levels in the original classification.

Epistemology is the branch of philosophy that studies "*the nature and sources of knowledge*". The interpretation taken by Brachman and many other of the logicist tradition in AI is that knowledge consists of propositions, whose formal structure is the source of new knowledge. Examples of representation languages belonging to this level include Brachman's own KL-ONE (Brachman & Schmolze, 1985) and its derivatives (including the semantic web languages OIL, DAML, DAML+OIL, RDFS, OWL) as well as object-based and frame-based modeling languages such as EER, LINGO and UML.

The rationale behind the design of epistemological languages is the following: (i) the languages should be designed to capture interrelations between pieces of knowledge that cannot be smoothly captured in logical languages; (ii) they should offer *structuring* mechanisms that facilitate understanding and maintenance, they should also allow for economy in representation, and have a greater computational efficiency than their logical counterparts; (iii) finally, modeling primitives in these languages should represent structural connections in our knowledge needed to justify conceptual inferences in a way that is independent of the meaning of the concepts themselves.

Indeed languages such as UML and OWL offer powerful structuring mechanisms such as classes, relationships (attributes) and subclassing relations. However, if we want to impose a certain *structure* in the representation of formula (F_1), in a language such as UML, we would have to face the following structuring choices: (a) consider that there are instances of apples that can posses the property of being red or, (b) consider that there are instances of red things that can have the property of being apples. Formally we can state either that (F_2) `∃x:Apple.red(x)` as well as (F_3) `∃x:Red.apple(x)`, and both these many-sorted logic formalizations are equivalent to the previous one-sorted axiom. However, each one contains an implicit structuring choice for the sort of the things we are talking about.

The design of epistemological languages puts a strong emphasis on the inferential process, and the study of knowledge is limited to its form, i.e., it is "*independent of the meaning of the concepts themselves*". Therefore, the focus of these languages is more on formal reasoning than on (formal) representation. Returning to our example, although the representation choice (b) seems to be intuitively odd, there is nothing in the semantics of a UML class or an OWL concept that prohibits any unary predicate such as red or tall to be modeled as such. In other words, since in epistemological languages the semantics of the primitive “sort” is the same as its corresponding unary predicate, the choice of which predicates correspond to sorts is completely left to the user.

In (Guarino, 1994), Guarino points out that structuring decisions, such as this one, should not result from heuristic considerations but instead should be motivated and explained in the basis of suitable *ontological distinctions*. For instance, in this case, the choice of Apple as the sort (a) can be justified by the meta-properties that we are ascribed to the term by the *intended meaning* that we give to it. The ontological difference between the two predicates is that Apple corresponds to a *Natural Kind* whereas Red corresponds to an *Attribution*. Whilst the former applies necessarily to its instances (an apple cannot cease to be an apple without ceasing to exist), the latter only applies contingently. Moreover, whilst the former supplies a *principle of identity* for its instances, i.e., a principle through which we judge if two apples are numerically the same, the latter cannot supply one. However, it is not the case that an object could subexist without obeying a principle of identity (van Leewen, 1991; Gupta, 1980), an idea which is defended both in philosophical ontology (e.g., Quine's dicto "*no entity without idendity*" (Quine, 1969)), and in conceptual modeling (e.g., Chen's design rational for ER (Chen, 1976)). Consequently, the structuring choice expressed in (F_3) cannot be justified.

For an extensive discussion on *kinds*, *attributions* and *principles of identity* as well as their importance for the practice of conceptual modeling we refer to chapter 4 of this thesis.

In addition to supporting the justified choice for structuring decisions, the ontological level has important practical implications from a computational point of view. For Instance, one can exploit the knowledge of which predicates hold necessarily (and which are susceptible to change) in the design and implementation of more efficient update mechanisms.

Finally, there are senses in which the term Red can be said to hold necessarily (e.g., "*scarlet is a type of red*" referring to a particular shade of color), and senses in which it carries a principle of identity to its instances (e.g., "*John is a red*" - meaning that "*John is a communist*"). The choice of representing Red as an attribution makes explicit the intended meaning of this predicate, ruling out these two possible interpretations. In epistemological and logical languages, conversely, the intended meaning of a predicate relies on our understanding of the natural language label used.

The position defended in this work is that, in general conceptual modeling languages, the meaning of structuring primitives should be characterized in terms of meta-level conditions that make explicit the ontological commitment made by the modeler when choosing a particular structuring primitive to represent a domain element. For example, in the profile used to create the ontology of figure 2.17, the modeling primitives are tagged with the ontological categories they represent. When choosing to represent Person as a Kind, the user explicitly commits to the axiomatization implied by this representation choice. There are alternative representations in which Person can be considered to hold only contingently, for example, if one considers Person as a synomym for LivingPerson and, let us say, Corpse as synonym for deceasedPerson (figure 2.17). However, when contrasting the two representations we notice that the two usages of the term, although syntactically identical, are semantically and ontologically diverse, since they possess incompatible meta-properties.

In this sense, the approach taken in this thesis departs from the one followed by top-level ontologies such as CyC (Lenat & Guha, 1990) and the IEEE Standard Upper-Level Ontology (SUO)[26]. In these cases, the idea is to create a specification that defines a complete inventory of reality in a way that all domain ontologies could commit to. CyC, for instance, has been under development since 1990 and, only its open source version (OpenCyC[27]), contains currently 6000 concepts and 60.000 assertions about these concepts. Conversely, the position taken here is that the choice of how to represent a certain domain of reality is the responsibility of a community of users. However, the intended meaning embedded in these representation choices should be made explicit via an ontological commitment to a system of meta-level categories, or a *foundational ontology*. In this way, agreement is reinforced on the meta-level and not on the level of individual domain theories. This is far from promoting a scenario where a unique meta-ontology exists. On the contrary, the position defended here is equivalent to the one of (Masolo et. al, 2003a), namely, that there should be a library of *foundational ontologies*, where each of these ontologies (explicitly) commits to different philosophical choices (see, for instance, section 3.4.3), and, consequently, is suitable for different classes of applications. Nonetheless, the stability and universality of these foundational ontologies should be supported by the theoretical developments in formal ontology, and afforded by empirical evidence, either in natural sciences or in the sciences of cognition.

---

[26] http://suo.ieee.org/

[27] http://www.opencyc.org/

3.3.3 On a suitable Meta-Conceptualization

Ontologies vary in the way they manage to represent an associated conceptualization. An ontology such as the one in figure 2.17 is more accurate than if it were represented in ER, OWL, pUML (Evans & Kent, 1999) or LINGO. This is because the modeling profile which is used commits to a much richer meta-ontology than the ones underlying these other languages. As a consequence, to formally characterize its ontological distinctions, a formal language with higher expressiveness is needed. When the stereotyped modeling primitives of the profile in figure 2.17 are used, an axiomatization in the language of *intensional modal logics* is incorporated in the resulting specification, constraining the intepretation of its terms.

Intensional modal logics are considerably more expressive than *SHIQ* or standard set theory. However, *SHIQ* has interesting properties such as computational tractability and decidability, which are properties that are in general absent in more expressive languages. Likewise, LINGO was designed to facilitate the translation to Object-Oriented implementations. Therefore, in the design of a general conceptual modeling language, the following tradeoff must be recognized. On one side we need a language that commits to a rich foundational ontology. This meta-ontology, however, will require the use of highly expressive formal languages for its characterization, which in general, are not interesting from a computational point of view. On the other side, languages that are efficient computationally, in general, cannot commit to a suitable meta-conceptualization. The obvious question is then: how can we design a general conceptual modeling language according to these conflicting requirements?

The position advocated here is analogous to the one defended in (Masolo et. al, 2003a), namely, that we actually need two classes of languages. On one hand, highly-expressive languages should be used to create strongly axiomatized ontologies that approximate as well as possible to the ideal ontology of the domain. The focus on these languages is on representation adequacy, since the resulting specifications are intended to be used by humans in tasks such as communication, domain analysis and problem-solving. The resulting domain ontologies, named *reference ontologies* in (Guarino, 1998), should be used in an *off-line* manner to assist humans in tasks such as meaning negotiation and consensus establishment. On the other hand, once users have already agreed on a common conceptualization, versions of a reference ontology can be created. These versions are named here *lightweight ontologies.* Contrary to reference ontologies, lightweight ontologies are not focused on representation adequacy but are designed with the focus on guaranteeing desirable computational properties. Examples of languages suitable for lightweight ontologies include OWL and LINGO. An example of a conceptual modeling language that is suitable for reference ontologies is the one developed throughout this work.

The importance of reference ontologies has been acknowledged in many cases in practice. For instance, (Ríos, 2003) illustrates examples of semantic interoperability problems that can pass undetected when interoperating lightweight ontologies. Likewise, (Fielding et. al, 2004) discusses how a principled foundational ontology can be used to spot inconsistencies and provide solutions for problems in lightweight biomedical ontologies. As a final example, the need for methodological support in establishing precise meaning agreements is recognized in the *Harvard Business Review* report of October 2001, which claims that *"one of the main reasons that so many online market makers have foundered [is that] the transactions they had viewed as simple and routine actually involved many subtle distinctions in terminology and meaning"*.

Once we have made clear that the meta-ontology developed throughout this work should be a foundational one, an important issue still remains to be addressed, namely, which kind of foundational ontology one should commit to. In (Strawson, 1959), the philosopher Peter Strawson draws a distinction between two different kinds of ontological investigation, namely, *descriptive* and *revisionary methaphysics*. Descriptive metaphysics aims to lay bare the most general features of the conceptual scheme that are in fact employed in human activities, which is roughly that of common sense. The goal is to capture the ontological distinctions underlying natural language and human cognition. As a consequence, the categories refer to cognitive artifacts more or less depending on human perception, cultural imprints and social conventions (Masolo et. al, 2003a), and do not have necessarily to agree on the principles advocated by the natural sciences. Nonetheless, the very existence of these categories can be empirically uncovered by research in cognitive sciences (Keil, 1979; Xu & Carey, 1996; Mcnamara, 1986) in a manner that is analogous to the way philosophers of science have attempted to elicit the ontological commitments of the natural sciences

Revisionary metaphysics, conversely, is prepared to make departures from common sense in light of developments in science, and considers linguistic and cognitive issues of secondary importance (if considered at all). The following example, from (Masolo et. al, ibid.), exemplifies these different approaches. Common sense distinguishes between *things* (*spatial objects* like a car, a city and the moon) and *events* (*temporal objects* like business processes, birthday parties and football games). According to the relativity theory, however, time is viewed as another dimension of objects on a par with the spatial dimensions. As a consequence, revisionist researchers propose that the common sense distinction between things and events should be regarded as an (ontologically irrelevant) historical and cognitive accident, and that it should be abandoned in favor of an ontology of processes (Sowa, 2000; Whitehead, 1978).

In summary, whilst a descriptive ontology aims at giving a correct account of the categories underlying human common sense, a revisionary ontology is committed to capture the intrinsic nature of the world in a way that is independent of conceptualizing agents. Nonetheless, the taxonomies of objects produced by both approaches can be shown to be in large degree compatible with each other, if only we are careful to take into account the different granularities at which each operates (Smith & Brogaard, 2002).

In order to motivate a choice for a foundational ontology that a general conceptual modeling language should commit to, we first must investigate the definition and purposes of conceptual modeling and conceptual specifications. In a seminal paper, John Mylopoulos (Mylopoulos, 1992) provides the following definition, which highlights some aspects of the discipline and the produced representations that are of great relevance for the discussion carried out here. First, he defines conceptual modeling as *"the activity of formally describing some aspects of the* ***physical*** *and* ***social*** *world around us for purposes of* ***understanding*** *and* ***communication****."* This passage highlights that conceptual modeling is about the modeling of reality and not about modeling a computational system. In another part of the text he states that *"conceptual modelling supports structuring and inferencial facilities that are* ***psychologically grounded****. After all, the descriptions that arise from conceptual modelling activities are intended to be used* ***by humans, not machines"***. This is an important point that also emphasizes the precedence of truthfulness to reality over features such as computational efficiency and tractability of the produced representations. Moreover, it draws attention to the importance of considering human-oriented issues, such as the need to produce representations that are pragmatically efficient and are psychologically grounded and amenable to human cognition. Finally, he summarizes these aspects in the following adequacy requirement for conceptual modeling languages:*"The adequacy of a conceptual modelling notation rests on its contribution to the construction of models of reality that promote a common understanding of that reality among their human users."*

The main objective of conceptual specifications (in particular domain ontologies) in computer science is to support tasks such as communication, domain learning and problem solving in disciplines such as domain engineering and database schema integration. Moreover, in cases such as inter-organizational service interoperability, information brokering and intelligent search in the Semantic Web, the ontologies produced have an intrinsic social nature. In the majority of cases, these ontologies represent conceptualizations that are afforded by our common sense view of reality. For this reason, in this work we defend the idea that a general conceptual modeling language should commit to a *descriptive foundational ontology*. Consequently, the foundational ontology presented here should be understood as a descriptive theory of *a priori* distinctions, focused on entities of the so-called *mesoscopic* level, i.e., the level of human experience. That is to say that the categories in our foundational ontology focus on meta-properties of everyday objects such as apples, people, cars, chairs and insurance claims but neither on microscopic or macroscopic entities. For instance, it is outside the scope of this thesis to consider ontological problems such as the ones discussed in (Lowe, 2001), which arise when characteristics of entities of the atomic and subatomic levels are considered.

3.4 Final Considerations

In the course of this chapter we have discussed the reasons that have historically motivated the use of philosophical ontology in computer science disciplines such as information systems, domain engineering and knowledge representation. A common aspect of these disciplines is the need to:

(i) promote reuse in a higher level of abstraction aiming at the maximization of the reuse of domain models;
(ii) produce domain specifications that are truthful to reality.

The main contribution of philosophical ontology in the accomplishment of these goals is the set of conceptual tools that have been developed along the years for constructing these general systems of categories. In philosophy, an ontology is committed only to the truth in the classical sense, i.e., it is meant to represent knowledge of reality in a way that is independent of a particular use one makes of it. Since the very task of semantic interoperation (e.g., database schema and framework integration, service interoperability) is to find a common conceptualization that different representations agree on, the potential benefit from philosophical ontology becomes evident or, how could this common conceptualization be achieved if not through a task-independent model of the underlying reality that different representations refer to?

The relations between the various senses in which the term ontology has been used in philosophy and computer science can be summarized as follows:

1. *Formal Ontology*, as conceived by Husserl, is part of the discipline of Ontology in philosophy (sense D1), which is, in turn, the most important branch of metaphysics;

2. Formal Ontology aims at developing general theories that accounts for aspects of reality that are not specific to any field of science, be it physics or conceptual modeling (sense D2);

3. These theories describe knowledge about reality in a way, which is independent of language, of particular states of affairs (states of the world), and of epistemic states of knowledgeable agents. In this thesis, these language independent theories are named (meta) *conceptualizations*. The representation of these theories in a concrete artifact is a *foundational ontology*;

4. A foundational ontology tries to characterize as accurately as possible the conceptualization it commits to. Moreover, it focuses on representation adequacy regardless of the consequent computational costs, which is not actually a problem since the resulting model is targeted at human users.

5. A foundational ontology can be used to provide real-word semantics for general *conceptual modeling languages*, and to constrain the possible interpretations of their modeling primitives. An ontology can be seem as the metamodel specification for an ideal language to represent phenomena in a given domain in reality, i.e., a language which only admits specifications representing possible state of affairs in reality (related to sense D3).

6. Finally, suitable general conceptual modeling languages can be used in the development of reference *domain ontologies*, which, in turn, among many other purposes, can be used to characterize the vocabulary of *domain-specific languages*.

Chapter 4

Universals and Taxonomic Structures

A central concern of this thesis is to construct a philosophical and cognitive ontology that can be used as a foundation for conceptual modeling languages. Moreover, we aim at formally characterizing the elements constituting this ontology. Finally, we intend to demonstrate the usefulness of the ontological categories and theories proposed in addressing recurrent problems in the practice of conceptual modeling.

The construction of the foundation ontology proposed in this thesis is organized in four complementary chapters, namely, chapters 4 to 7. In this chapter, we aim at providing ontological foundations for the philosophical categories of *universals* and *individuals*, which are represented in conceptual modeling by the constructs of *Types (classes, classifiers)* and their *instances*, respectively.

Types are fundamental for conceptual modeling, being represented in all major conceptual modeling languages (e.g., OO classes, EER Entity types, LINGO and OWL concepts). In general, monadic types used in structural conceptual models stand for universals whose instances are *substantials*. The precise notions of *substantial* adopted in this thesis will be formally defined in chapter 6. However, for now, an intuitive understanding of this term will suffice. The term, as used here, is akin to what is sometimes named *thing* (Bunge, 1977), *endurant* (Masolo et al., 2003a), or *continuant* (van Leeuwen, 1991) in the philosophical literature. Intuitively, it is similar to what is termed *object*, in the colloquial use of the latter term. Substantials are entities that persist in time while keeping their identity (as opposed to *events* such as a kiss, a business process or a birthday party). Examples include physical and social persisting entities of everyday experience such as balls, rocks, students, the North Sea and Queen Beatrix.

In the practice of conceptual modeling, a set of primitives is often used to represent distinctions in different types of substantial universals (Type, Role, State, Mixin, among others). However, there is still a lack of methodological support for helping the user of the language to decide how to represent elements that denote universal properties in a given domain (viz. Person, Student, Red Thing, Physical Thing, Deceased Person, Customer) and, hence, modeling choices are often made in an ad hoc manner. Likewise is the judgment of what are the admissible relations between these modeling constructs. Finally, an inspection of the literature shows that there is still much debate on the meaning of these categories (Wieringa & de Jong & Spruit, 1995; Bock & Odell, 1998; Steimann, 2000b; Evermann & Wand, 2001b).

In this chapter, we propose a philosophically and psychologically well-founded theory of substantial universals for conceptual modeling. The ontological distinctions and postulates proposed by this theory are presented in section 4.1.

In section 4.2, the ontological distinctions countenanced by the theory are organized in a typology of universals, together with a number of constraints on how the elements in this typology can be combined to form taxonomic structures. This typology and associated constraints are further used to derive a *modeling profile* for conceptual modeling, along with a set of methodological guidelines that govern its use. Still in section 4.2, we demonstrate the usefulness of the theory and derived profile proposed to evaluate and improve the conceptual quality of class hierarchies and concept taxonomies, and to solve some recurrent problems in the practice of conceptual modeling.

In section 4.3, we present a number of empirical research efforts carried out in cognitive psychology that provide evidence supporting the proposed theory of universals.

In section 4.4, we elaborate on two different (albeit complementary) systems of modal logics designed to formally characterize the distinctions and constraints proposed by the theory. The section also discusses how these systems address the limitations of classical (unrestricted extensional) modal logics in that respect.

Section 4.5 discusses related work and demonstrates how the ontological distinctions proposed in this chapter are compatible but richer than those found in the conceptual modeling literature hitherto.

Finally, section 4.6 elaborates on some final considerations.

4.1 A Theory of Universal Types: Philosophical and Psychological Foundations

In (van Leeuwen, 1991), Jacques van Leeuwen shows an important grammatical difference occurring in natural languages between common nouns (CNs) on one side and arbitrary general terms (adjectives, verbs, mass nouns, etc…) on the other. Common nouns have the singular feature that they can be combined with determiners and serve as argument for predication in sentences such as:

(i) (exactly) five mice were in the kitchen last night;
(ii) the mouse which has eaten the cheese, has been in turn eaten by the cat.

In other words, if we have the patterns *(exactly) five X…* and *the Y which is Z…*, only the substitution of X, Y, Z by CNs will produce sentences that are grammatical. To verify that, we can try the substitution by the adjective Red in the sentence (i): *(exactly) five red were in the kitchen last night.* A request to "count the red in this room" cannot receive a definite answer: Should a red shirt be counted as one or should the shirt, the two sleeves, and two pockets be counted separately so that we have five reds? The problem in this case is not that one would not know how to finish the counting but that one would not know how to start, since arbitrarily many *subparts of a red thing are still red.*

It is important to emphasize that red here is not used as a CN, i.e., as a synonym for red color, which is a nominalization of an adjective and denotes a particular shade of red. This reading would make a sentence such as *"exactly 256 greys exist in the Windows color palette"* grammatically viable. In fact, in order to play the same role as a CN, general terms must be nominalized, which implies a shift to the category of common nouns (e.g., whiteness, the fall of Jack or a Bucket of water).

The explanation for this feature unique of CNs lies on the function that determinates (demonstratives and quantifiers) play in noun phrases, which is to determine a certain range on individuals. Both reference and quantification requires that the thing (or things) which are referred or which form the domain of quantification are determinate individuals, i.e., their conditions for individuation and identity must be determinate. In other words, if it is not determinate how to count Xs or how to identify the X that is the same as Y, the sentences in the patterns (i) and (ii) do not express determinate propositions, i.e., propositions with definite truth values.

According to (van Leeuwen, 1991), this syntactic distinction between the two linguistic categories reflects a semantical and ontological one, and the distinction between the grammatical categories of CNs and arbitrary general terms can be explained in terms of the ontological categories of *Sortal* and *Characterizing universals* (Strawson, 1959), which are roughly their ontological counterparts. Whilst the latter supply only a *principle of application* for the individuals they collect, the former supply both a principle of application and a *principle of identity*. A principle of application is that in accordance with which we judge whether a general term applies to a particular (e.g., whether something is a Person, a Dog, a Chair or a Student). A principle of identity supports the judgment whether two particulars are the same, i.e., in which circumstances the identity relation holds.

In (Mcnamara, 1986), cognitive psychologist John Macnamara, investigates the role of sortal concepts in cognition and provides a comprehensive theory for explaining the process that a child undergoes when learning proper nouns and common nouns. He proposes the following example: suppose a little boy (Tom), who is about to learn the meaning of a proper name for his puppy. When presented to the word "Spot", Tom has to decide what it refers to. A demonstrative such as "that" will not suffice to determinate the bearer of the proper name. How to decide that "that", which changes all its perceptual properties is still *Spot*? In other words, which changes can Spot suffer and still be the same? As Macnamara (among others) shows, answers to these questions are only possible if *Spot* is taken to be a proper name for an individual, which is an instance of a Sortal universal. The principles of identity supplied by the Sortals are essential to judge the validity of all identity statements. For example, if for an instance of the sortal *Statue* loosing a piece will not alter the identity of the object, the same does not hold for an instance of *Lump of Clay*.

The statement that we can only make identity and quantification statements in relation to a Sortal amounts to one of the best-supported theories in the philosophy of language, namely, that the identity of an individual can only be traced in connection with a Sortal Universal, which provides a *principle of individuation* and *identity* to the particulars it collects (Mcnamara, 1986, 1994; Gupta, 1980; Lowe, 1989; van Leeuwen, 1991). The position advocated in this chapter affirms an equivalent stance for a theory of conceptual modeling. We defend that among the conceptual modeling counterparts of general terms (*types*) only constructs that represent sortals can provide a principle of identity and individuation for its instances. As a consequence, a principle that represents the junction of

Quine's dicto *"no entity without identity"* (Quine, 1969) with the position defended in this section *"no identity without a Sortal"* can be postulated:

Postulate 4.1: Every individual in a conceptual model (CM) of the domain must be an instance of a conceptual modeling type (CM-Type) representing a sortal.

As argued by Kripke (Kripke, 1982), a proper name is a rigid designator, i.e. it refers to the same individual in all possible situations, factual or counterfactual. For instance, it refers to the individual Mick Jagger both now (when he is the lead singer of Rolling Stones and 62 years old) and in the past (when he was the boy Mike Philip living in Kent, England). Moreover, it refers to the same individual in counterfactual situations such as the one in which he decided to continue in the London School of Economics and has never pursued a musical career. We would like to say that the boy Mike Philip is identical with the man Mick Jagger that he latter became. However, as pointed out by Wiggins (Wiggins, 2001) and Perry (Perry, 1970), statements of identity only make sense if both referents are of the same type. Thus, we could not say that a certain Boy is the same Boy as a certain Man since the latter is not a Boy (and vice-versa). However, as Putnam put it, when a man *x* points to a boy in a picture and says "I am that boy", the pronoun "I" in question is typed not by Man but by a supertype of Man and Boy (namely, Person), which embraces *x*'s entire existence (Putnam, 1994). A generalization of this idea amounts to a thesis, proposed by Wiggins, named thesis D (Wiggins, 2001): If an individual falls under two sortals in the course of its history there must be exactly one ultimate sortal of which both sortals are specializations. (Griffin, 1977) elaborates Wiggins' thesis D in terms of two correlated principles:

a) The Restriction Principle: if an individual falls under two distinct sortals F and F' in the course of its history then there is at least one sortal of which F and F' are both specializations.

b) The Uniqueness Principle: if an individual falls under two distinct sortals F and F' in the course of its history then there is at most one *ultimate sortal* of which F and F' are both specializations. A sortal F is ultimate if there is no other sortal F' distinct from F which F specializes.

It is not the case that two incompatible principles of identity could apply to the same individual *x*, otherwise *x* will not be a viable entity (determinate particular) (van Leeuwen, 1991). For instance, suppose an individual *x* which is an instance of both *Statue* and *Lump of clay*. Now, the answer to the question whether loosing a piece will alter the identity of *x* is indeterminate, since each of the two principles of identity that *x* obeys imply a different answer. As a consequence, we can say that if two sortals F and F' intersect (i.e., have common individuals in their extension), the principles of identity contained in them must be equivalent. Moreover, F and F' cannot supply a principle of identity for *x*, since both sortals apply to x only contingently, and a principle of identity must be used to identify *x* in all possible worlds. Therefore, there must be a sortal G that supplies the principle of identity carried by F and F'. This proves the restriction principle. The uniqueness of the ultimate sortal G can be argued as follows: (i) G is a sortal, since it supplies a principle of identity for all the things in its extension; (ii) if it restricts a sortal H then, since H cannot supply an incompatible principle of identity, H either is: equivalent to G (i.e., does supply the same principle of identity) and therefore should be ultimate, or does not supply a principle of identity for the particulars in its extension (see text on dispersive classifiers below). This proves the uniqueness principle. The unique ultimate sortal G that supplies the principle of identity for its instances is named a *substance sortal*.

As a consequence of thesis D, we derive a second postulate:

Postulate 4.2: An individual represented in a conceptual model of the domain must instantiate exactly one CM-Type representing an ultimate Substance Sortal.

In the example above, the sortal Person is the *unique* substance sortal that defines the validity of the claim that Mick Jagger is the same as Mike Philip or, in other words, that Mike Philip persists through changes in height, weight, age, residence, etc., as the same individual. Person can only be the sortal that supports the proper name Mick Jagger in all possible situations because it applies necessarily to the individual referred by the proper name, i.e., instances of Person cannot cease to be so without ceasing to exist. As a consequence, the extension of a substance sortal is world invariant. This meta-property of universals is named *Modal Constancy* (Gupta, 1980) or *rigidity* (Guarino & Welty, 2002b) and is formally stated as follows:

Definition 4.1 (Extension functions): Let *W* be a non-empty set of possible worlds and let $w \in W$ be a specific world. The extension function $ext_w(G)$ maps a universal G to the set of its instances in world *w*. The extension function ext(G) provides a mapping to the set of instances of the universal G that exist in all possible worlds, such that

1. $ext(G) = \bigcup_{w \in W} ext_w(G)$

■

Definition 4.2 (Specialization relation): Let F and G be two universals such that F is a specialization of G. Then, for all w ∈ W we have that

2. $ext_w(F) \subseteq ext_w(G)$

■

Definition 4.3 (Rigid Universal): A universal G is rigid (or modally constant) iff for any w,w' ∈ W

3. $ext_w(G) = ext_{w'}(G)$

■

Putting definitions 4.1 and 4.3 together, we have that for any rigid universal G the following is true

4. $ext(G) = ext_w(G)$, for all w ∈ W

A rigid universal is one that applies to its instances necessarily, i.e., in every possible world. Every substance sortal G is a rigid universal. Examples of *non-rigid* Sortals include universals such as *Boy* and *Adult Man* in the example discussed above, but also *Student*, *Employee*, *Caterpillar* and *Butterfly*, *Philosopher*, *Writer*, *Alive* and *Deceased*. Actually, these examples of sortals are not only non-rigid, but they are *anti-rigid*. Non-rigidity is the simple logical negation of rigidity, i.e., a universal is non-rigid if it does not apply necessarily to at least one of its instances. In contrast, a universal is anti-rigid if it does not apply necessarily to all its instances. An example of a non-rigid universal is *Seatable*. Suppose the instances of Seatable include a particular chair *x* and a particular crate *c*. In this case, whilst *x* instantiates Seatable necessarily, this is not the case for *c*: a crate can cease to be steady to afford sitting but still be the same crate.

Non-rigidity and anti-rigidity of universals are defined formally in the sequel:

Definition 4.4 (Non-Rigid Universal): A universal G is non-rigid iff for a w ∈ W

5. There is an x such that $x \in ext_w(G)$, and there is a $w' \in W$ such that $x \notin ext_{w'}(G)$

■

Definition 4.5 (Anti-Rigid Universal): A universal G is anti-rigid iff for any $w \in W$

6. For every x such that $x \in ext_w(G)$, there is a $w' \in W$ such that $x \notin ext_{w'}(G)$

■

Notice that non-rigidity constitutes a much weaker constraint than what is imposed by anti-rigidity, i.e. anti-rigidity is a sort of non-rigidity. In (Guarino & Welty, 2002b), a universal is named *semi-rigid* iff it is non-rigid but not anti-rigid.

Sortals that possibly apply to an individual only during a certain phase of its existence are named *phased-sortals* in (Wiggins, 2001). Contrary to substance sortals, phased-sortals are anti-rigid universals. For example, for an individual John instance of Student, we can easily imagine John moving in and out of the Student type, while being the same individual, i.e. without loosing his identity. Moreover, for every instance x of Student in a world w, there is another world w' in which x is not an instance of Student. Finally, as a consequence of the Restriction Principle, we have that for every phased-sortal PS that applies to an individual, there is a substance sortal S of which PS is a specialization. Formally a phased-sortal can be defined as follows:

Definition 4.6 (Phased-Sortal): Let PS be a universal and let S be a *substance sortal* specialized (restricted by) PS. Now, let

7. $ext_w(\sim PS) = ext_w(S) \setminus ext_w(PS)$

be the complement of the extension of PS in world w. In this formula, the symbol \ represents the set theoretical operation of set difference. The universal PS is a *phased-sortal* iff for all worlds $w \in W$, there is a $w' \in W$ such that

8. $ext_w(PS) \cap ext_{w'}(\sim PS) \neq \emptyset$

■

Putting (2), (4) and (6) together we can derive another postulate:

<u>Postulate 4.3:</u> A CM-Type representing a rigid universal cannot specialize (restrict) a CM-Type representing an anti-rigid one.

<u>Proof:</u> Take an arbitrary rigid universal G which specializes an anti-rigid universal F. Let $\{a,b,c,d\}$ and $\{a,b\}$ be the extension of F and G in world w, respectively. By (6), there is a world w' in which $a \notin ext_{w'}(F)$. By (4), however, $ext_w(G) = ext_{w'}(G)$ and, thus, $a \in ext_{w'}(G)$. By (2), $ext_{w'}(G) \subseteq ext_{w'}(F)$ and, ergo, $a \in ext_{w'}(F)$. We then have that $a \notin ext_{w'}(F)$ and $a \in ext_{w'}(F)$, which is a contradiction. We therefore conclude that a rigid universal cannot specialize an anti-rigid one.

□

If PS is a phased-sortal and S is the substance sortal specialized by PS, there is a specialization condition φ such that *x* is an instance of PS iff *x* is an instance S that satisfies φ (van Leeuwen, 1991). A further clarification on the different types of specialization conditions allows us to distinguish between two different types of phased-sortals which are of great importance to the practice of conceptual modeling, namely, *phases* and *roles*.

Phases (also named *dynamic subclasses* in Wieringa & de Jonge & Spruit, 1995) or states (Bock & Odell, 1998) constitute possible stages in the history of a substance sortal. Examples include: (a) Alive and Deceased: as possible stages of a Person; (b) Catterpillar and Butterfly of a Lepidopteran; (c) Town and Metropolis of a City; (d) Boy, Male Teenager and Adult Male of a Male Person. *Universals representing phases constitute a partition of the substance sortal they specialize*. For example, if ‹Alive, Deceased› is a *phase-partition* of a substance sortal Person then for every world w, every Person x is either and instance of Alive or of Deceased but not of both. Moreover, if x is an instance of Alive in world *w* then there is world *w*' such that x is not an instance of Alive in *w*', which in this case, implies that x is an instance of Deceased in w'.

This can be generalized as follows: Let $\langle P_1 \ldots P_n \rangle$ be a phase-partition that restricts the sortal S. Then we have that for all $w \in W$:

9. $$ext_w(S) = \bigcup_{P_i \in \langle p_1 \ldots p_n \rangle} ext_w(P_i)$$

and for all $P_i, P_j \in \langle P_1 \ldots P_n \rangle$ (with $i \neq j$) we have that

10. $$ext_w(P_i) \cap ext_w(P_j) = \emptyset$$

Finally, it is always possible (in the modal sense) for an instance *x* of S to become an instance of each P_i, i.e., for any $P_i \in \langle P_1 \ldots P_n \rangle$ which restricts S, and for any instance *x* such that $x \in ext_w(S)$, there is a world w' ∈ W such that $x \in ext_{w'}(P_i)$. This is equivalent of stating that for any $P_i \in \langle P_1 \ldots P_n \rangle$ the following holds

11. $ext(S) = ext(P_i)$

Contrary to phases, roles do not necessarily form a partition of substance sortals. Moreover, they differ from phases with respect to the specialization condition φ. For a phase P, φ represents a condition that depends solely on intrinsic properties of P. For instance, one might say that if Mick Jagger is a Living Person then he is a Person who has the property of being alive or, if Spot is a Puppy then it is a Dog who has the property of being less than an year old. For a role R, conversely, φ depends on extrinsic (relational) properties of R. For example, one might say that if John is a Student then John is a Person who is enrolled in some educational institution or that, if Peter is a Customer then Peter is a Person who buys a Product y from a Supplier z. In other words, an entity plays a role in a certain context, demarcated by its relation with other entities. In general, we can state the following: Let R be a role that specializes a sortal S (named the *allowed type* for R (Bock & Odell, 1998)) and φr be a binary relation defined between R and the universal D on which R is *externally dependent of* (Welty & Guarino, 2001). For instance, $\varphi_{enrollment} \subseteq$ Student × School, $\varphi_{purchase\text{-}from} \subseteq$ Customer × Supplier or $\varphi_{Marriage} \subseteq$ Husband × Wife. Moreover, let the domain of the relation φr in world *w* (Dom_w) be defined as follows: $Dom_w(\varphi_r) = \{x \mid <x,y> \in ext_w(\varphi_r)\}$. Then for all worlds $w \in W$ we have that

12. $ext_w(R) = Dom_w(\varphi_r)$

Relational properties are represented in this chapter in terms of plain (extensional) relations for simplicity only. In chapter 6, we present the complete treatment of relations and relational properties which is adopted in our foundational ontology. That is, in turn, used in chapter 7 to elaborate on a fuller characterization of roles. For the same reason of simplicity, we will thus not extend the representation of relations in the example above to the case of n-ary relations.

Although Frege argued at length that *"one cannot count without knowing what to count"* (Frege, 1934), in artificial logical languages inspired by him, natural language general terms such as common nouns, adjectives and verbs are treated uniformly as predicates. For instance, if we want to represent the sentence “there are tall men”, in the fregean approach of classical logic we would write $\exists x\ (Man(x) \wedge Tall(x))$. This reading puts the count noun Man (which denotes a Sortal) on an equal logical footing with the predicate Tall. Moreover, in this formula, the variable x is interpreted into a “supposedly” universal kind Thing (or Entity). So, the natural language reading of the formula should be “there are things that have the property of being a man and the property of being tall”. Since, by postulate 4.1, all individuals must be instances of a substance sortal we must conclude that Thing is a unique universal ultimate sortal that is able to supply a principle of identity for all elements that we consider in our universe of discourse. Moreover, by postulate 4.2, this principle of identity must be unique. Can that be the case?

In (Hirsch, 1982), Hirsch argues that concepts such as Thing, (Entity, Element, among others) are *dispersive*, i.e., they cover many concepts with different principles of identity. For instance, in the extension of Thing we might encounter an individual x that is a cow and an individual y that is a watch. Since the principles of identity for Cows and Watches are not the same, we conclude that Thing cannot supply a principle of identity for its instances. Otherwise, x and y would obey incompatible principles of identity and, thus, would not be determinate individuals. Therefore, as defended in (van Leeuwen, 1991; Gupta, 1980; Mcnamara, 1994; Hirsch, 1982), dispersive concepts do not denote sortals, despite that they are considered CNs in natural languages, and *therefore cannot have direct instances.* More than that, a principle of identity supplied by a substance sortal G is inherited by all universals that specialize G or, to put in another way, all subtypes of G carry the principle of identity supplied by G. Thus, all specializations of a sortal are themselves sortals, ergo,

Postulate 4.4: A CM-Type representing a dispersive universal cannot specialize a CM-Type representing a Sortal.

4.2 An Ontologically Well-Founded Profile for Conceptual Modeling Universals

We start this section by compiling the ontological distinctions proposed by the theory of section 4.1 in a *typology of substantial universals*. The elements constituting this typology are depicted in figure 4.1 below.

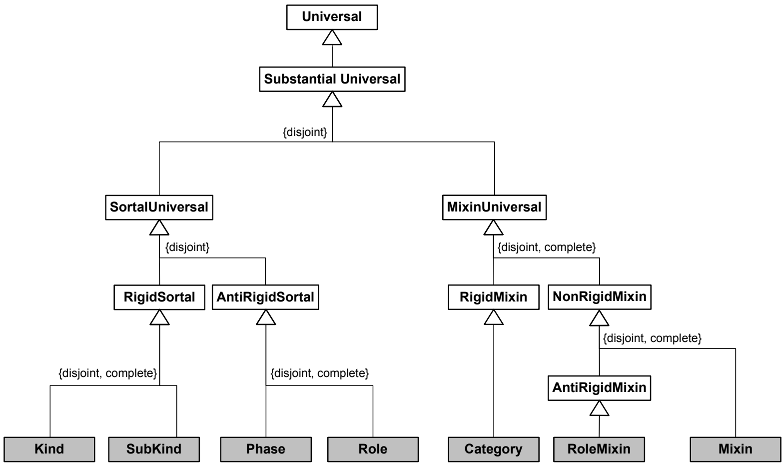


Figure 4-1 Ontological Distinctions in a Typology of Substantial Universals

The idea here is to use the ontology of universals depicted above to design a fragment of a conceptual (ontology) modeling language. This language should contain (as modeling primitives) constructs that represent the bottom-most specialization of the ontological categories proposed by this theory, i.e., the leaf nodes of the generalization tree of figure 4.1 (highlighted in grey).

We use the extension mechanisms of the Unified Modeling Language (UML) to illustrate these ideas by proposing a *profile* whose modeling elements represent each of the relevant distinctions made by the theory. It is important to emphasize, however, that UML is used here solely with the purpose of exemplification, due to its acceptance and practical relevance, and to the convenience offered by the built-in extension mechanisms of the language. A similar result could be achieved by extending, for instance, LINGO via its *theory inclusion* extension mechanisms (Falbo & Menezes & Rocha, 1998), or simply by proposing a new modeling language.

The UML built-in extension mechanisms allow one to modify the language elements to suite certain modeling needs. Extensions to the language can be performed in two different ways: (i) by specializing the UML metamodel (layer 2) to add new semantics to UML modeling elements; or (ii) by changing the so-called MOF model (layer 3) to add new elements to the UML metamodel. The former mechanism is named *lightweight extension* and the latter *heavyweight extension*. A coherent set of such extensions, defined accordingly to a specific purpose or domain, constitutes a *UML profile* (OMG, 2003b).

In a *representation mapping* (see chapter 2) from the ontology of figure 4.1 to the UML metamodel, we can map the category of *universal* to the UML meta-construct of a *Class*. In UML, a *"Class describes a set of Objects* *sharing a collection of Features, including Operations, Attributes and Methods, that are common to the set of Objects."* (OMG, 2001, p.32). *"The model is concerned with describing the intension of the class, that is, the rules that define it. The run-time execution provides its extension, that is, its instances."* (ibid., p.202). The concept underlying this construct, namely, the one of a *Type*, is one of the most common modeling concepts in conceptual modeling. For instance, it can be found practically in all object-modeling languages (e.g., OMT, Objectory, OML), semantic web languages (e.g., OWL, DAML+OIL), ontology modeling languages (e.g., LINGO) and information systems grammars (e.g., ER). For this reason, the principles and distinctions laid out in section apply to any of these conceptual modeling languages in which the substantial universal concept is represented.

If we continue this representation mapping, we realize that in UML (but also in all of the languages aforementioned) there are no modeling constructs that represent the ontological categories specializing *Substantial Universal* in figure 4.1. In other words, there are ontological concepts prescribed by the theory of section 4.1 that are not represented by any modeling construct in the language. A case of ***ontological incompleteness*** at the modeling language level (see section 2.2.4).

In order to remedy this problem, in the sequel, we propose a lightweight extension to UML that represents finer-grained distinctions between different types of classes. The proposed *profile* contains a set of *stereotyped* classes that support the design of ontologically well-founded conceptual models according to the theory proposed in this chapter. Moreover, the profile also contains a number of constraints (derived from the postulates of the theory) that restrict the way the modeling constructs can be related. The goal is to have a metamodel such that all syntactically correct specifications using the profile have logical models that are *intended world structures* of the conceptualizations they are supposed to represent (see definition 3.2 in chapter 3).

In summary, according to the ontological semantics given to the profile, the ontological interpretation of the class meta-construct is that of a *Universal*. Each of the stereotyped classes comprising the profile, thus, represents one of the leaf ontological categories specializing substantial universal in figure 4.1. In terms of the profile, this also means that all stereotyped classes have as the *base class* the UML Class meta-construct (OMG, 2003b).

In the subsections that follow, we elaborate in each of these distinctions. It is important to emphasize that the particular classes chosen to exemplify each of the proposed categories are used for illustration purposes only. For example, when stereotyping class *Person* as a *Kind* we are not advocating that Person should be in general considered as a kind in conceptual modeling.

Conversely, the intention is to make explicit the consequences of this modeling choice. The choice itself, however, is always left to the model designer.

4.2.1 Kinds and Subkinds

A UML class in this profile stereotyped as a « kind » represents a *substance sortal* that *supplies* a principle of identity for its instances. Kinds can be specialized in other *rigid* subtypes that inherit their supplied principle of identity named *subkinds*. For instance, if we take Person to be a kind then some of its subkinds could be Man and Woman. In general, the stereotype « subkind » can be omitted in conceptual specifications without loss of clarity.

Every object in a conceptual specification using this profile must be an instance of a Kind, directly or indirectly (postulate 4.1). Moreover, it cannot be an instance of more than one ultimate Kind (postulate 4.2). Figure 4.2 shows an excerpt of a conceptual specification that violates the second postulate (extracted from the CYC[28] upper-level ontology). Here, we assume that the kinds *Social Being* and *Group* supply different principles of identity. Moreover, it is considered that *Group* supplies an extensional principle of identity, i.e. two groups are the same iff they have the same members. This is generally incompatible with a principle supplied by *Social Being*: we can change the members of a company, football team or music band and still have the same social being. Moreover, the same group can form different social beings with different purposes. One should notice that if the particular referred by the proper name *The Beatles* would be an instance of both Kinds, it would not be a determinate object (an answer to the question whether it was still the same thing when Ringo Star replaced Pete Best, is both affirmative and negative!). Figure 4.3 shows a version of the specification of figure 4.2 that obeys the constraints of this profile. In this revised representation, we have explicit modeled the *substantial* The Beatles and the group of people that compose this individual in a given circumstance as distinct individuals.

---

[28] http://www.opencyc.org/

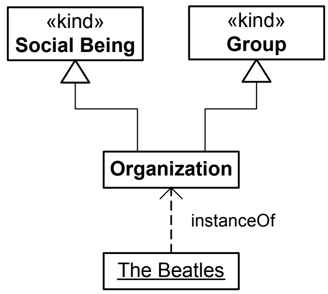


Figure 4-2 Example of an instance with conflicting principles of identity

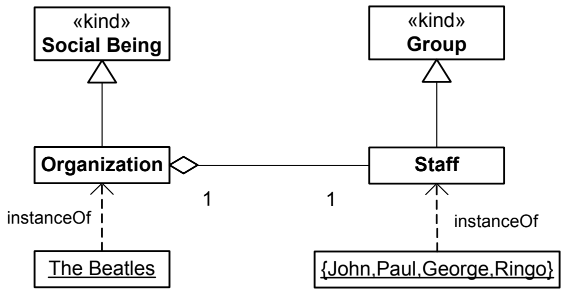


Figure 4-3 An ontologically correct version of the specification of figure 4-2

By postulate 4.3, rigid classes cannot be supertyped by anti-rigid ones. Therefore, kinds cannot appear in conceptual models as subtypes of phases, roles (see section 4.2.3), and role mixins (section 4.2.4).

4.2.2 Phases

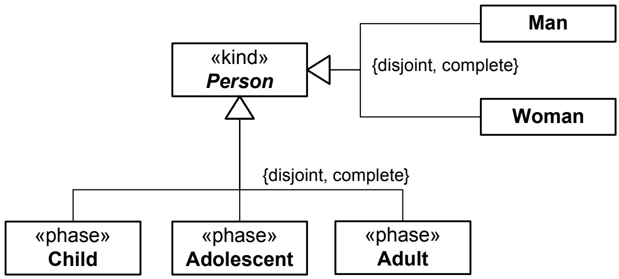
UML classes stereotyped as « phase » in this profile represent the phased-sortals *phase*. Figure 4.4 depicts an example with the kind Person, its subkinds Man and Woman and the phases Child, Adolescent and Adult.

The classes connected to one single hollow arrowhead symbol in UML (concrete syntax for the subtyping relation) define a *generalization set* (OMG, 2003b). In the UML 2.0 metamodel, the classes that are member of a generalization are necessarily disjoint. However, by default, they do not form a partition (in the mathematical sense) of the restricted subclass. In order to represent that a set of subclasses form a partition of their direct common superclass, the corresponding generalization set must be decorated with the constraints {*disjoint, complete*}. In other words, the concept of a *type partition* is represented as a decorated generalization set in UML.

The kind person in figure 4.4 represents the substance sortal restricted by a *phase-partition* ⟨Child, Adolescent, Adult⟩. According to formulas (9) and (10) in section 4.1, the generalization set representing a phase-partition must indeed by a partition in the mathematical sense and, hence, the restricted superclass must always be defined as an *abstract class* (i.e., a class that cannot have direct instances). In the specification of figure 4.4, the subkinds Man and Woman also define a partition for the substance sortal Person. However, contrary to phases, subkinds do not have to be necessarily defined in a partition.

In UML, an abstract class is represented by a class with its name italicized.

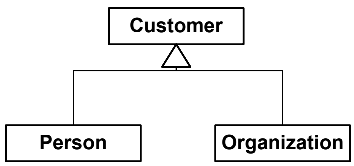
Figure 4-4 Two partitions of the same Kind Person: a subkind partition (Man, Woman) and a phase partition (Child, Adolescent, Adult)



4.2.3 Roles

UML classes stereotyped as « role » represent the phased-sortals *role*. Roles and Phases are anti-rigid universals and cannot appear in a conceptual model as a supertype of a Kind (postulate 4.3). However, sometimes subtyping is wrongly used in conceptual modeling to represent alternative *allowed types* that can fulfill a role. For instance, in figure 4.5, the intention of the model is to represent that customers are either persons or organizations. An analogous example is shown in figure 4.6. In general, being a customer is assumed to be a contingent property of person, i.e., there are possible worlds in which a Person is not a customer but still the same person. Likewise, a participant can stop participating in a Forum without ceasing to exist. In summary, if the universals represented in figures 4.5 and 4.6 are ascribed the semantics just discussed, these specifications are ontologically incorrect representations according to the theory postulated in section 4.1.

Figure 4-5 Problems on modeling Roles and their *allowed types*



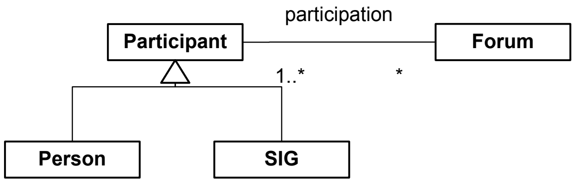


Figure 4-6 Mistaken cardinality specification for Roles

In a series of papers (Steimann, 2000a, 2000b, 2001), Steimann discusses the difficulties of specifying allowed types for Roles that can be played by instances of disjoint types such as the roles *Customer* and *Participant* in figures 4.5 and 4.6, respectively. As a conclusion, the author claims that the solution to this problem lies in the separation of role and type hierarchies. This solution not only departs from the traditional use of the concepts of role and type in the practice of conceptual modeling but, in particular, it leads to a radical revision of the UML metamodel (a heavyweight extension). In chapter 7 of this thesis, while discussing several problems w.r.t. role modeling found in the literature of conceptual modeling, we show that this claim is not warranted. To support our argument and propose a solution to the problem of role modeling with disjoint allowed types, we propose a design pattern based on the profile presented in this section. The solution presented has a smaller impact to UML than the one proposed by the author, since it does not demand heavyweight extensions to the language. The *roles with disjoint allowed types design pattern* proposed in chapter 7 can then be used as an ontologically correct solution to this recurrent modeling problem.

Figure 4.7 below depicts an ontologically correct version of figure 4.5 generated by the application of the design pattern aforementioned. We return to the discussion of this issue and, specifically, of this particular example in chapter 7. The notion of role mixin in figure 4.7 is discussed in section 4.2.4 below.

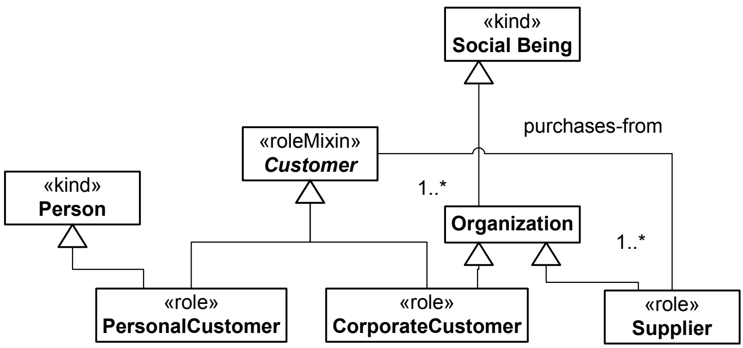


Figure 4-7 An ontologically correct version of the specification of figure 4-5 by using the profile and the *roles with disjoint allowed types* design patterns

Figure 4.6 contains yet another conceptual problem. In this specification, a participant can take part in zero-to-many forums. It is common in Database and Object-Oriented Design to use a minimum cardinality equal to zero to express that in a certain state of the system, for example, an object of type Participant is not related to any object of type Forum. However, from a conceptual viewpoint, the involvement in this relation is part of the very definition of the role type. In this example, the association *participation* is a specialization (restriction) condition (see section 4.1), which is part of the content of the concept Participant. In other words, a *Participant* is a *Person* or *SIG* that takes part in a *Forum*. As a consequence of formula (7), the following constraint must hold for classes stereotyped as « role »:

Let X be a class stereotyped as « role » and r be an association representing X's restriction condition. Then the minimum cardinality of X.r must be at least 1 (#X.r ≥ 1).

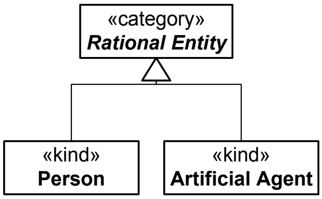
This constraint is elaborated in chapter 8 of this thesis after our treatment of relations is presented and formally characterized in chapter 6.

4.2.4 Mixins

Conceptual modeling types classified as Mixins represent the dispersive universals discussed in section 4.1. Mixin types are perceived to be of great importance in structuring the specification of conceptual models (Jacobson & Booch & Rumbaugh, 1998; Booch, 1994; Welty & Guarino, 2001). They can represent top-most types such as Thing, Entity, Element (discussed in section 4.1) but also concepts such as *RationalEntity*, which represent an abstraction of properties that are common to multiple disjoint types (figure 4.8). In this case, the mixin *RationalEntity* can be judged to represent an essential property that is common to all its instances and it is itself a rigid type. We use the stereotype «category» to represent a rigid mixin that subsumes different kinds.

In contrast, some mixins are anti-rigid and represent abstractions of common properties of roles. These types are stereotyped as «roleMixin» and represent dependent anti-rigid non-sortals. Examples of role mixins include the so-called *formal roles* (Welty & Guarino, ibid.) such as *whole* and *part* and *initiatior* and *responder*. However, role mixins are more general than formal roles, representing any abstraction of common contingent properties of multiple disjoint roles. An example of a role mixin is depicted in figure 4.7. Further examples are discussed in chapter 7 of this thesis.

Figure 4-8 Example of a *category*, i.e., a rigid mixin.



Moreover, some mixins represent properties that are essential to some of its instances and accidental to others. As discussed in section 4.1, this meta-property is named non-rigidity (a weaker constraint than anti-rigidity). An example is the mixin *Seatable* (figure 4.9), which represents a property that can be considered essential to the kinds Chair and Stool but accidental to Crate, Paper Box or Rock. We use the stereotype « mixin » (without further qualification) to represent *non-rigid* non-sortals.

Finally, by postulate 4.4, we have that mixins cannot appear in a conceptual model as subclasses of kinds, phases or roles. Moreover, due to postulate 4.3, rigid mixins (categories) cannot be subsumed by anti-rigid ones, i.e., by role mixins. Finally, since they cannot have direct instances, a mixin must always be depicted as an abstract class in a UML conceptual specification.

Figure 4-9 Example of a semi-rigid mixin

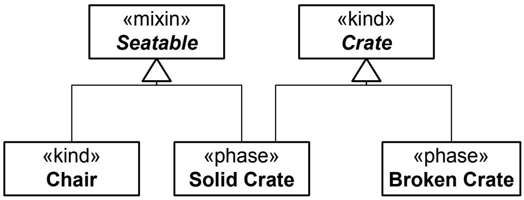


Table 4.1 below summarizes the profile proposed in this section.

Table 4-1 Summary of the proposed modeling profile for the conceptual modeling representation of substantial universals

Stereotype	Constraints
	RIGID SORTALS
« kind »	supertype is **not** a member of {« subkind », « phase », « role », « roleMixin »}
« subkind »	supertype is **not** a member of {« phase », « role », « roleMixin »} For every subkind SK there is a unique kind K such that K is a supertype of SK
	ANTI-RIGID SORTALS
« phase »	Always defined as part of partition. For every Phase P there is a unique Kind K such that K is a supertype of P
« role »	Let X be a class stereotyped as « role » and R be an association representing X's restriction condition. Then, $\#X.R \geq 1$ For every Role X there is a unique Kind K such that K is a supertype of X

	NON-SORTALS
« category »	supertype is **not** a member of {« kind », « subkind », « phase », « role », « roleMixin »} Always defined as an abstract class
« roleMixin »	supertype is **not** a member of {« kind », « subkind », « phase », « role »}. Let X be a class stereotyped as « roleMixin » and R be an association representing X's restriction condition. Then, #X.R ≥ 1 Always defined as an abstract class
« mixin »	supertype is **not** a member of {« kind », « subkind », « phase », « role », « roleMixin »} Always defined as an abstract class

4.3 Psychological Evidence

The postulates presented in section 4.1 are represented in a list of psychological claims proposed by cognitive psychologist John Mcnamara in (Mcnamara, 1994). The proposed psychological claims are related to the cognitive interpretation of linguistic expressions. In that article, Mcnamara defends the position that there is a logic underlying the fact that we can understand certain propositions, and the proposed set of psychological claims points to a class of logics that take cognizance of this fact. This position, which is also supported by, for instance, (Mcnamara, 1986; Xu, 2004; Lidz & Waxman & Freedman, 2003) is analogous to the one advocated by Chomsky in defense of his notion of a Universal Grammar (Chomsky, 1965, 1980, 1986). Chomsky is the proponent of the theory that states that the reason why we can learn a natural language is due to the existence of a *mental language*, a *linguistic competence* that is nature-supplied, uniform across individuals and complete in each one. According to this view, there is a close fit between the mind's linguistic properties and properties of natural languages and, hence, natural languages have the properties they do because they can be recognized and manipulated by infants without the meta-linguistic support that is available to second-language learners. Therefore, for Chomsky, a grammar for a particular language is *descriptively adequate* if it correctly describes its object, namely the linguistic intuition of the native speaker.

In the same spirit, a number of cognitive scientists (see, for example, Mcnamara 1986, 1994), have proposed a theory of *logical competence* based on the notion of a *Language of Though* (Fodor, 1975). For him, the reason we can learn the *meaning* of natural language symbols for categories, individuals and their properties without the meta-linguistic support available to second language learners lies on the mapping between the language specific symbols for these categories onto a system of categories already existing in the language of though, i.e., a *cognitive ontology*.

In this section, we discuss some empirical evidence that support many of the points defended throughout this chapter. This evidence results from a number of psychological experiments in the areas of cognitive psychology, which has been developed with the aim of investigating categorical development and logical competence in infants since a pre-language age of 3-4 months.

Firstly, many laboratory results provide evidence that infants in the early age of 3-4 months are already able to form categories (e.g., Eimas & Quinn, 1994). (Cohen & Younger, 1983; Mcnamara, 1982), for instance, provide evidence that children are able to classify objects into categories for which they have no natural-language symbols. The reason that categorization appears so early in cognitive development is related to its fundamental relevance to cognition. As (Markman 1989, p. 11) puts it: *"Categorization . . . is a means of simplifying the environment, of reducing the load on memory, and of helping us to store and retrieve information efficiently".* Without concepts, mental life would be chaotic. If we perceived each entity as unique, we would be overwhelmed by the sheer diversity of what we experience and unable to remember more than a minute fraction of what we encounter. Furthermore, if each individual entity needed a distinct name, our language would be staggeringly complex and communication virtually impossible. In contrast, if you know nothing about a novel object but are told it is an instance of X, you can infer that the object has all or many properties that Xs have (Smith & Medin, 1981).

Nonetheless, in order to able the learn what are the properties that we can expect instances of X to have, another ability, namely the ability to (re)identity instances of X must be present. If all of an object's properties were immediately manifest to the child upon every encounter there would be no need to learn and remember what these properties were. However, carrying knowledge of substances becomes necessary since most of a substance's properties are not manifest but hidden from us most of the time (Milikan, 1998). For example, different encounters with Felix will reveal different properties about the individual Felix and about the kind Cat. However, for this to happen, the child must be able to: (i) recognize that Felix *is a Cat*; (ii) recognize that the individual that can jump from the table to the TV set *is the same* as the one that can be fed milk; (iii) recognize that Tom *is also a Cat*, thus, he can also be fed milk. In summary, both principles of application and identity are fundamental in our construal of the environment.

A number of experiments provide evidence for the existence of an individuation and indentity system in infants by the age they begin to form categories. Researchers such as (Xu & Carey, 1996; Spelke et. al., 1995; Moore, Borton, & Darby, 1978) provide evidence that by the early age 3-4 months old infants have criteria for deciding whether an object is the same one as a previously seen object. (Xu & Carey, ibid.), for instance, provide evidence that until about nine months of age, infants rely on a unique principle of individuation and identity for all objects, which is supplied by the type *Physical Object*. This notion of physical object is synonymous to *maximally connected physical object whose parts move along together in a spatiotemporal continuous path*. As defended by the authors, in this sense, physical object is indeed a sortal since any identity and individuation statements involving two physical objects is determinate. They term this system of individuation an *object-based system*. These results show that not only does spatiotemporal discontinuity lead to a representation of two distinct objects, but also that spatiotemporal continuity leads to a representation of a single, persisting object. Other laboratories have also replicated this basic finding using somewhat different procedures (e.g., Aguiar & Baillargeon, 1999).

Thus, even young infants have some criteria for establishing representations of distinct objects. These first criteria are spatiotemporal in nature, including generalizations such as: (i) objects travel on spatiotemporally connected paths; (ii) two objects cannot occupy the same space at the same time, and (iii) one object cannot be in two places at the same time (Xu, 2004).

Xu and Baker (Xu & Baker, 2003) address the question whether infants can use perceptual property information for object individuation, i.e., if non-sortal categories can support the judgment of individuation statements. The results show that, for 10-month infants, perceptual properties are at best only used to confirm the application of the principle of identity which is first supplied by the sortal *Physical Object*. Moreover, these results show that *spatiotemporal evidence for a single object changing properties overrides perceptual property information* (see also Xu & Carey, 1996). (Xu & Carey & Quint, 2004) also shows that 12-months infants are not able to use non-sortal categories alone to support the judgment of individuation statements. This view is also supported by (Milikan, 1998) who calls the attention for the fact that children come to appreciate separable dimensions, such as color, shape, and size only after a considerable period in which "holistic similarities" dominate their attention.

Results from (Xu & Carey & Quint, 2004) show that between 9 to 12 months of age a second system of individuation emerges in infant's cognition. This is named a *kind-based individuation system* and operates independently of the object-based attention system. In general, by the early age of 12 months, infants have already developed the multiple-sortal system which is used by adults to judge individuation and identity statements. Moreover, (Xu, 2004) shows that: (i) representations of object kinds can override strong spatiotemporal evidence for a single object; (ii) perceptual property information is always treated as kind-relative. As remarked by the author, *"during this period, infants' worldview undergoes fundamental changes: They begin with a world populated with objects...By the end of the first year of life, they begin to conceptualize a world populated with sortal-kinds... In this new world, objects are thought of not as 'qua object' but rather 'qua dog' or 'qua table'"*. Other experiments such as those conducted by Bonatti et al. (2002), Waxman and Markow (1995) and Booth and Waxman (2003) corroborate with these findings and support the claim that around 12-months infants are sensitive to the distinction between sortals and arbitrary general terms, which are represented differently and used differently in individuation tasks.

(Xu, 2004) also suggests that it is not a coincidence that along with acquiring their first words, infants begin to develop their kind-based system of individuation. The bulk of a child's first words are concrete nouns, including proper names and names for sortal universals (Milikan, 1998). For example, (Mcnamara, 1972) shows that children learn common nouns (the linguistic counterparts of sortal universals) before they learn predicates such as verbs and adjectives (counterparts of characterizing universals). (Gentner, 1982) presents evidence that this finding holds cross-culturally for children learning German, Kaluli, Japanese, Mandarin Chinese, Turkish and English. In summary, in early stages of language learning, children are more likely to pick words for sortals than of other kinds. Although perceived, attributes and events are construed, individuated and conceptualized under the dependence of a sortal (Mcnamara, 1986, p.145).

(Katz & Baker & Mcnamara, 1974) and (Mcnamara, 1982) provide evidence that children younger than 17-month-old are able to distinguish proper names by coordinating the notions of individual and kind. According to these findings, children are able to judge that individuals of some kinds, but not of others, are likely to be the bearers of a proper name. Together with the results from (Gelman & Taylor, 1984), these findings provide strong indications that under certain circumstances children are led to take some words as proper names (namely, when applied to individuals of familiar kinds) and in others to take them as sortals (when applied to individuals of unfamiliar kinds). (Mcnamara, 1986) advocates that this evidence is an obvious suggestion that children have the appropriate sortals to support the learning of proper names in some language. In addition to that, when reporting on a number of observations extracted from the linguistic record he kept of his son, Mcnamara remarks that his son's use of proper names behaved like rigid designators from the start (Mcnamara, 1986).

According to (Rosch et. al, 1976; Markman & Hutchinson, 1984), when learning a word for a kind of object, basic-level sortals (*substance sortals*) are the ones that occur to children. For example, when creating categories, children attend to similarities among dogs before subclassifying them and before they attend to properties dogs share with other animals. (Mcnamara, 1986, p.147) strongly argues that substance sortals hold a psychologically salient and privileged position compared to other types of rigid sortals, and that children's perceptual systems seem to be especially tuned to identify substance sortals. This position finds evidence in the results of (Anglin, 1977; Mervis & Crisafi, 1982).

In summary, a substantial number of psychological experiments confirm the philosophy of language thesis that individuation and identity judgment can only take place with the support of a sortal universal. Both systems of individuation and identity that are employed by human cognition are sortal-based. Humans start with a principle of identity afforded by the unique sortal physical object and, in a later developmental phase, undergo a cognitive shift to a system that employs a multitude of principles of identity supplied by different substance sortals. In both systems, perceptual property information is secondary and can be overridden. (Xu, 2004) defends the position that this developmental process makes good sense in terms of learnability, since it starts the child on the solid ground of a concept of object that seems to be inate (Spelke, 1990) and it allows the child to work with these individuated objects and with the help of language, ultimately develop a new ontology of sortal-kinds. In addition, as defended by (Milikan, 1998), the primacy of substance sortals also makes good sense in evolutionary terms since, from the standpoint of an organism that wishes to learn, the focus should be on constructing categorizations that are *essential*, since they are the ones that supply knowledge that affords the most useful and reliable inferences.

4.4 Formal Characterization

In this section we present two complementary systems of *intensional sortal modal logics* (intensional modal logics with quantification restricted by sortal terms). These systems differ mainly in the nature of the entities they admit to the domain of quantification. However, more importantly, they share the objective of making explicit the distinction between sortals and mixin universals (neglected in classical modal logics) in the sense that only sortals can carry a principle of persistence and trans-world identity for their instances. Moreover, they also incorporate in their semantics, the constraints that qualify the different types of sortals universals recognized by the theory presented in section 4.1.

4.4.1 Quantifying over Momentary States

In the use of the term *individuals* that we have been employing in this chapter, individuals are special types of endurants, i.e., entities that exist in time while keeping their identity. Examples including ordinary *objects* of everyday experience such as: a person, a student, a house, a dog, among others. However, in a UML Object Diagram (Instance Specification), an UML instance *"represents an entity at a point in time (a snapshot)"* (OMG, 2003c, p.59). In the language presented in this section, we explore this idea by taking the primitive elements of quantification in the system of intensional modal logics proposed to represent states of instances in given time boundaries, or to put it simply, snapshots of ordinary substantial individuals.

The following example illustrates some of these ideas. Suppose that there is an individual person referred by the proper name John. As discussed in section 4.1, proper names for substantials refer rigidly and, hence, if we say that John weights 80kg at t_1 but 68kg at t_2 we are in the two cases referring to the same individual, namely the particular John. Now, let x_1 and x_2 be snapshots representing the projection of John at time boundaries t_1 and t_2, respectively. The truth of the statements *overweight(John,*t_1*)* and *overweight(John,*t_2*)* depends only on whether overweight applies to the states x_1 or x_2, respectively. In other words, the judgment if an individual *i* is an instance of a universal G (e.g., overweight) in world w depends only whether the principle of application carried by G applies to the state of *i* in w.

In a computational system, the identity of x_1 and x_2 is guaranteed by the presence of some artificial identification scheme (oids, primary keys, surrogates). However, from an ontological perspective, how can one determine that, despite of possibly significant dissimilarities, x_1 and x_2 are states of the same particular John? As argued throughout the chapter, this is done via a principle of trans-world identity and, in particular, a principle of persistence supplied by the substance sortal Person, of which John is an instance.

In classical (extensional) modal logics, no distinction is made between different types of universals. Universals are represented as predicates in the language that divide the world (at each circumstance) into two classes of elements: those that fall under them and those that do not. This principle determines the extension of each universal at each circumstance. Classical (one-place) predicates, i.e., as functions from worlds to sets of individuals, properly represent the *principles of application* that are carried by all universals but fail to represent the *principles of identity* which are unique of sortals. Equivalently, they treat all objects as obeying the same principle of identity.

This difference is made explicit in the language L_1 defined in this section in the following way:

– The intention of the proper name John is represented by an *individual concept J*, i.e., a function that maps to a snapshop x_i of John in each possible world w. The notion of individual concepts, first introduced by Leibniz, refers to a singleton property that only holds for one individual. For instance, the property of being Mick Jagger has a single instance, namely the individual person Mick Jagger;

– Sortal universals, such as Person, are represented as *intensional properties*, which are functions from possible worlds to sets of individual concepts. For instance, for the sortal Person there is a function ℓ that maps every world w to a set of individual concepts (including *J*). An individual *x* is a Person in world w iff there is an individual concept $k \in \ell(w)$ such that $k(w) = x$;

– Individual Concepts represent the principle of identity supplied by the universal Person such that if $J(w) = x_1$ and $J(w') = x_2$ then we say that x_1 in w is the same Person as x_2 in w', or in general: for all individuals x,y representing snapshots of an individual C of type T we say that x in w is the same T as y in w' iff C is in the extension of T and $C(w) = x$ and $C(w') = y$;

– Whilst the principle of identity is represented by sortal determined individual concepts that trace individuals from world to world, the principle of application considers individuals only at a specific world. For instance, John is overweight in word w iff *overweight(J(w),w)* is true.

In summary, in the language L_1 presented below, the primitive elements in the domains of quantification are momentary states of substantial individuals, not the individuals themselves. Ordinary substantial individuals are instead represented by individual concepts.

The idea of representing objects of ordinary experience by individual concepts is similar to the solution presented in (Heller & Herre, 2004) in which individual concepts for substantial individuals are named *abstract substances* or *persistents*. The notion of momentary state of individuals adopted here is similar to that of *presentials* in (Heller & Herre, ibid.).

In the sequel, we formally define the syntax and semantics of L_1.

Syntax of L_1

Let L_1 be a language of modal logics with identity with a vocabulary V = (K,B,A,P,T) where: (a) T is a set of individual constants; (b) P is a non-empty set n-ary predicates; (c) A is a set of anti-rigid sortal types; (d) B is a set of subkinds; (e) K is a non-empty set of kinds (substance sortal type); (f) $R = K \cup B$ is named the set of rigid sortal types and the set $C = R \cup A$, the set of sortal types; The alphabet of L_1 contains the traditional operators: = (equality), $\neg$ (negation), $\rightarrow$ (implication), $\forall$(universal quantification), $\Box$ (necessity). The notions of term, sortal and formula are define as follows:

Definition 4.7

(1) all individual constants and variables are *terms*;

(2) All sortal types belong to the category of *Sortal Types;*

(3) If s and t are terms, then $s = t$ is an *atomic formula*;

(4) If P is a n-place predicate and $t_1 \ldots t_n$ are terms, then $P(t_1,..,t_n)$ is an *atomic formula*;

(5) If A and B are formulas, then so are $\neg A$, $\Box A$, $(A \rightarrow B)$;

(6) If S is sortal classifier, x is a variable and A is a formula, then $(\forall S,x)A$ is a formula.

■

The symbols $\exists$ (existential quantification), $\wedge$ (conjunction), $\vee$ (disjunction), $\Diamond$ (possibility) and $\leftrightarrow$ are defined as usual:

Definition 4.8

(7) $(A \wedge B) =_{def} \neg(A \rightarrow \neg B)$;

(8) $(A \vee B) =_{def} ((A \rightarrow B) \rightarrow B)$;

(9) $(A \leftrightarrow B) =_{def} (A \rightarrow B) \wedge (B \rightarrow A)$

(10) $\Diamond A =_{def} \neg\Box\neg A$

(11) $(\exists S,x)\ A =_{def} \neg(\forall S,x)\ \neg A$

(12) $(\exists! S,x)\ A =_{def} (\exists S,y)(\forall S,x)\ (A \leftrightarrow (x = y))$

■

In L_1, all quantification is restricted by Sortals. The quantification restricted in this way makes explicit what is only implicit in standard predicate logics. As previously discussed, suppose we want to state the following proposition: (a) *There are red tasty apples*. In classical predicate logic we would write down a *logical* formula such as (b) ∃x (apple(x) ∧ tasty(x) ∧ red(x)). In an ontological reading, (b) states that "there are things which are red, tasty and apple". The theory proposed section 4.1 rejects that we can conceptually grasp an individual under a general concept such as Thing or Entity or, what is almost the same, that a logic (or conceptual modeling language) should presupposed the notion of a *bare particular*. Moreover, it states that only a sortal (e.g., Apple) can carry a principle of identity for the individuals it collects, a property that is absent in attributions such as Red and Tasty. For this reason, a logical system, when used to represent a formalization of conceptual models, should not presuppose that the representations of natural general terms such as Apple, Tasty and Red stand in the same logical footing. For this reason, (a) should be represented as (∃Apple,x) (tasty(x) ∧ red(x)) in which the sortal binding the variable x is the one responsible for carrying its principle of identity.

In L_1, sortal classifiers are never used in a predicative position. Therefore, if S ∈ C is a sortal type, the predicate s(x) (in lowercase) is a meta-linguistic abbreviation according to the following definition.

Definition 4.9

$$s(t) =_{def} (\exists S,x)\ (x = t)$$

■

According to this definition, the sentence "John is a Man" is actually better rendered as "John is identical to a Man". In opposition, in the sentence "John is Tall", the copula represents the "is" of predication, which denotes a relation of mere equivalence.

Semantics of L_1

Definition 4.10 (model structure)

A model structure for L_1 is defined as an ordered couple $\langle \mathcal{W}, \mathcal{D} \rangle$ where: (i) W is a non-empty set of possible worlds; (ii) L_1 adopts a *varying domain frame* (Fitting & Mendelsonh, 1998) and, thus, instead of a set, $\mathcal{D}$ is a function that assigns to each member of $\mathcal{W}$ a non-empty set of elements.

■

Given a model structure $\mathcal{M}$ $(= \langle \mathcal{W}, \mathcal{D} \rangle)$, the intention of individual constant can be represented by an *individual concept*, i.e., a function i that assigns to each world $w \in \mathcal{W}$, an individual in $\mathcal{D}(w)$. Formally we have that

Definition 4.11 (individual concept)

Let $\mathcal{M} = \langle \mathcal{W}, \mathcal{D} \rangle$ and $\mathcal{U} = \bigcup_{w \in W} D(w)$.

An individual concept i in $\mathcal{M}$ is function from $\mathcal{W}$ into $\mathcal{U}$, such that $i(w) \in \mathcal{D}(w)$ in all worlds. For a given model structure $\mathcal{M}$ we define I as a set of individual concepts defined for that structure.

■

The intention of an n-place predicate is defined (as usual) as an n-ary property, i.e., a function that assigns to each world $w \in \mathcal{W}$ a set of n-tuples. If a tuple $\langle d_1 \ldots d_n \rangle$ belongs to the representation of a predicate at world w, then $d_1 \ldots d_n$ stand in w in the relation expressed by the predicate.

Definition 4,12 (property)

An n-ary property (n>0) in $\mathcal{M}$ is a function $\mathcal{P}$ from $\mathcal{W}$ into $\wp(\mathcal{D}(w))^n$, i.e., if $\langle d_1 \ldots d_n \rangle \in \mathcal{P}$(w), then $d_1 \ldots d_n \in \mathcal{D}(w)$.

■

The intension of sortal classifiers is defined in such a way that both the principles of application and identity are represented. This is done by what is named in (Gupta, 1980) *sorts*, i.e., *separated intensional properties*.

Definition 4.13 (sort)

Let $\mathcal{M} = \langle \mathcal{W}, \mathcal{D} \rangle$ be a model structure. An *intensional property* in $\mathcal{M}$ is a function ℓ from $\mathcal{W}$ into the powerset of individual concepts in $\mathcal{M}$ (i.e., $\wp(I)$).

An intensional property assigns to each world a set of individual concepts, and it can be used to represent the intention of a sortal type in the following way. Suppose that ℓ represents the intension of the sortal type $\mathcal{S}$ and that the individual concept i belongs to ℓ at world w, i.e., $i \in \ell(w)$. Then $i(w)$ is a $\mathcal{S}$ in w, and $i(w')$ is identical to $i(w)$ in w.

Let ℓ be an intensional property in $\mathcal{M}$, and let $\mathcal{L} = \bigcup_{w \in W} \ell(w)$.

Now, let i,j be two individual concepts such that $i,j \in \mathcal{L}$. We say that the intensional property ℓ is *separated* iff: if there is a world $w \in \mathcal{W}$ such that $i(w) = j(w)$ then, for all $w' \in \mathcal{W}$, $i(w') = j(w')$, i.e., $i = j$.

Finally, a *sort* in a model structure $\mathcal{M}$ is an intensional property which is separated.

■

The requirement of separation proposed in (Gupta, 1980) states, for example, that if two individual concepts for Person, say 007 and James Bond, apply to the same object in a world w then they apply necessarily to the same object. This prevents unlawful conceptualizations in which a substantial individual splits or in which two individuals can become one while maintaining the same identity.

Given a sort ℓ in $\mathcal{M}$ we designate by $\ell[w]$ the set of objects that fall under ℓ in w. Formally,

Definition 4.14:

$\ell[w] = \{d: d \in \mathcal{D}(w)$ and there is an individual concept $i \in \ell(w)$ such that $i(w) = d\}$.

■

Moreover, we define the set of objects in w that are *possibly* ℓ, i.e.,

Definition 4.15

$\ell\,|[w]| = \{d: d \in \mathcal{D}(w)$ and there is an individual concept $i \in \ell(w')$ such that $i(w') = d\}$.

■

We therefore are able to define the notion of *counterpart* relative to a sort ℓ.

Definition 4.16 (counterpart)

We say that d *in world* w *is the same* ℓ *as* d' *in* w' iff there is an individual concept i that belongs to ℓ at some world (i.e., there is a w'' such that $i \in \ell(w'')$) and $i(w) = d$ and $i(w') = d'$. The ℓ *counterpart* in w' of the individual d in w is the unique individual d' such that d' *in world* w' *is the same* ℓ *as* d *in* w.

■

Definition 4.17 (model)

A model in L_1 can then be defined as a triple $\langle \mathcal{W}, \mathcal{D}, \delta \rangle$ such that:

1. $\langle \mathcal{W}, \mathcal{D} \rangle$ is a model structure for L_1;

2. δ is an interpretation function assigning values to the non-logical constants of the language such that: it assigns an individual concept to each individual constant c ∈ T of L_1; an n-ary property to each n-place predicate p ∈ P of L_1; a sort to each sortal type S ∈ C of L_1.

The interpretation function δ must also satisfy the following constraints.

3. If S ∈ R then the sort ℓ assigned to S by δ must be such that: for all $w,w' \in \mathcal{W}$, $\ell(w) = \ell(w')$, i.e., all rigid sortals are world invariant (modally constant);

4. Let S ∈ (B ∪A) be a subkind or an anti-rigid sortal type. Then, there is a kind S' ∈ K such that, for all $w \in \mathcal{W}$, $\delta(S)(w) \subseteq \delta(S')(w)$;

5. Let S, S'∈ K be two Kinds and let ℓ and ℓ' be the two sorts assigned to S and S' by δ, respectively. Then we have that: there is a $w \in \mathcal{W}$ such that $\ell(w) \cap \ell'(w) \neq \emptyset$ iff $\ell = \ell'$, i.e., sorts representing kinds do not intersect unless they are identical. In other words, this restriction states that individuals belong to one single substance sortal, i.e., they obey one single principle of identity;

6. Let S ∈ A be an anti-rigid sortal type. The sort ℓ assigned to S by δ must be such that: for all $w \in \mathcal{W}$, and for all individual concepts $i \in \ell(w)$, there is a world $w' \in \mathcal{W}$ such that $i \notin \ell(w')$;

7. Let S, S'∈ K be two Kinds and let ℓ and ℓ' be the two sorts assigned to S and S' by δ, respectively. Then we have that: there is a $w \in \mathcal{W}$ such that $\ell[w] \cap \ell'[w] \neq \emptyset$ iff $\ell = \ell'$. Differently from (5) above this restriction states that individual states of objects can only be referred by individual concepts of the same kind;

■

Finally, we are now in position to define an assignment for L_1.

Definition 4.18 (assignment)

An assignment for L_1 relative to a model $\langle \mathcal{W}, \mathcal{D}, \delta \rangle$ is a function that assigns to each variable of L_1 an ordered pair $\langle \ell, d \rangle$, where ℓ is a sort relative to the modal structure $\langle \mathcal{W}, \mathcal{D} \rangle$ and $d \in \mathcal{U} = \bigcup_{w \in W} D(w)$.

If a is an L_1 assignment then $a_o(x)$ is the object assigned to variable x by a and $a_S(x)$ is the sortal to which x is bound. Moreover, it is always the case that $a_o(x) \in a_S(x)|[w]|$ for all variables.

■

Definition 4.19

An assignment a' for L_1 is an ℓ variant of a at x in w iff:

1. a' is just like a except perhaps at x (abbreviated as $a' \sim_X a$),
2. $a'_s(x) = \ell$,
3. $a'_o(x) \in \ell[w]$.

■

Definition 4.20

The w' variant of an assignment a relative to w (abbreviated as $f(w',a,w)$) is the unique assignment a' that meets the following conditions:

(i) $a'_S(x) = a_S(x)$ at all variables x,

(ii) $a'_o(x)$ in w' is the $a_S(x)$ *counterpart* of $a_o(x)$ in w relative, at all variables x.

■

Definition 4.21

Finally, let α be an expression in L_1 and, let *the semantic value of* α *at world* w *in model* $\mathcal{M}$ *and relative to assignment* a be the value of the valuation function $\mathcal{V}^{w}_{M,a}$.

With these definitions, we are able to define the semantics of L_1 as follows:

(a). If α is an individual constant or a sortal type, then $\mathcal{V}^{w}_{M,a}(\alpha) = \delta(\alpha)(w)$.

(b). If α is variable, then $\mathcal{V}^{w}_{M,a}(\alpha) = a_o(\alpha)$

(c). If α is an atomic formula $t_1=t_2$, then $\mathcal{V}^{w}_{M,a}(\alpha) = T$ if $\mathcal{V}^{w}_{M,a}(t_1) = \mathcal{V}^{w}_{M,a}(t_2)$. Otherwise $\mathcal{V}^{w}_{M,a}(\alpha) = F$.

(d). If α is an atomic formula $P(t_1 \ldots t_n)$, then $v^{w}_{M,a}(\alpha)$ = T if $\langle v^{w}_{M,a}(t_1) \ldots v^{w}_{M,a}(t_n) \rangle \in \delta(P)(w)$. Otherwise $v^{w}_{M,a}(\alpha)$ = F.

(e). If α is the formula $\neg$A, then $v^{w}_{M,a}(\alpha)$ = T if $v^{w}_{M,a}(A)$ = F. Otherwise $v^{w}_{M,a}(\alpha)$ = F.

(f). If α is the formula (A$\rightarrow$B), then $v^{w}_{M,a}(\alpha)$ = T if $v^{w}_{M,a}(A)$ = F or $v^{w}_{M,a}(B)$ = T. Otherwise $v^{w}_{M,a}(\alpha)$ = F.

(g). If α is the formula $(\forall S,x)A$, then $v^{w}_{M,a}(\alpha)$ = T if $v^{w}_{M,a'}(A)$ = T for all assignments *a'* which are $\delta(S)$ variants of *a* at x in *w*. Otherwise $v^{w}_{M,a}(\alpha)$ = F.

(h). If α is the formula $\Box$A, then $v^{w}_{M,a}(\alpha)$ = T if $v^{w'}_{M,f(w',a,w)}(A)$ = T for all $w' \in \mathcal{W}$. Otherwise $v^{w}_{M,a}(\alpha)$ = F.

■

The language L_1 has been proposed based on the first of the four systems introduced in (Gupta, 1980). Gupta, however, does not elaborate of different types of sortals. Consequently, restrictions from (3) to (7) posed to δ in definition 4.17 are simply not defined in his system. Restriction (7), in particular, would be rejected by Gupta, as a consequence of his contingent (or relative) view of identity. Note that restriction (7) implies (5) but not vice-versa.

It is widely accepted that any relation of identity must comply with the so-called Leibniz's law: if two individuals are identical then they are necessarily identical (van Leeuwen, 1991). Relativists, however, adopt the thesis that it is possible for two individuals to be identical in one circumstance but different in another. A familiar example, cited by Gupta, is that of a statue and a lump of clay. The argument proceeds as follows: Suppose that in world *w* we have a statue *st* of the Dalai Lama which is identical to the lump of clay *loc* this statue is made of. In w, *st* and *loc* have exactly the same properties (e.g., same shape, weight, color, temperature, etc.). Suppose now that in world w', a piece (e.g., the hand) is subtracted from *st*. If the subtracted piece is an inessential part of a statue then the statue *st'* that we have in *w'* is identical to *st*. In contrast, the lump of clay *loc'* which *st'* is made of is different from *loc*. In sum, we have in *w'* the same statue as in *w* but a different lump of clay.

In Gupta's system, without restriction (7), we have that for two individual concepts *i* and *j* such that *i(w) = j(w)*, it is possible that there is a world *w'* such that *i(w') ≠ j(w')*. In other words, the formula (α) $((\exists \text{Statue},x)(x = dl) \wedge (\exists \text{LoC},y)(y = dl) \wedge \Diamond(x \neq y))$ is satisfiable.

Here, we reject this possibility for two reasons. Firstly, because we support the view that Leibniz's rule must hold for a relation to be considered a relation of identity. Otherwise, any equivalence relation such as *being instance of the same class* must be considered a relation of identity. Secondly, if Gupta's primitive elements are thought as momentary states, then (α) does not actually qualify as a statement of relative identity. It actually expresses that two objects can coincide (i.e., share the same state) in a world *w* but not in a different world *w'* (van Leeuwen, 1991). Notice that if restriction (7) is assumed, formula (α) is no longer satisfiable.

Proof: (a) if (x = dl) is true then there is an individual concept st of Statue that refer in the actual world *w* to the same entity *d* as dl; (b) if (y = dl) is true then there is an individual concept loc of LoC that refer in the actual world *w* to the same entity *d* as dl; (c) by transitivity of equality, st and loc refer to the same d in world *w* and, consequently, d is then both of kind Statue and of kind LoC in w; (e) due to (7), the intentions of Statue and LoC are identical; (f) finally, due to separation, st and loc must coincide in every world.

□

In L_1, we take the *multiplicationist* (Masolo et al., 2003a) position that *st* and *loc* do not actually share the same state in *w* in the strong sense. In opposition, we consider that the states *st(w)* and *loc(w)* are numerically different albeit instantiating the same universals.

A simple way to modify L_1 in order to account for coincidence as manifested in Gupta's system can be achieved by:

(a). removing the constraint (7) in definition 4.17;

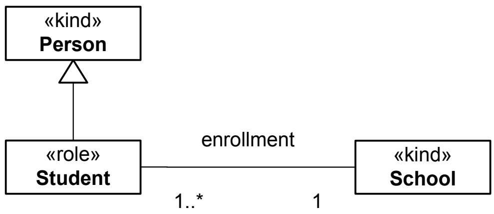
(b). including the operator ≈ for coincidence with the following semantics: If α is an atomic formula $t_1 \approx t_2$, then $v^{w}_{M,a}(\alpha) = T$ if $v^{w}_{M,a}(t_1) = v^{w}_{M,a}(t_2)$. Otherwise $v^{w}_{M,a}(\alpha) = F$;

(c). Defining the identity relation between individual constants as $(t_1 = t_2) =_{def} \Box(t_1 \approx t_2)$, i.e., two continuants are identical if they coincide in every possible world.

Example

The UML specification of figure 4.10 depicts and example of a conceptual specification produced using the profile defined in table 4.1.

Figure 4-10 Example depicting a phased-sortal role, its allowed type and a relational restriction condition



The L_1 rendering of this specification is presented below:

□(∀Student,x person(x))
□(∀Student,x ∃!School,y enrolled-in(x,y))
□(∀School,x ∃Student,y enrolled-in(y,x))

According to definition 4.9, □(∀Student,x person(x)) is actually an abbreviation for □((∀Student,x)(∃Person,y) (y = x)).

In this case we have that

K = R = C = {Person}, B = ∅, A = {Student},
P = {enrolled-in}, T = ∅

A L_1 model for this specification is presented below:

$\mathcal{W} = \{w,w'\}$
$\mathcal{D}(w) = \{\alpha,\beta,\gamma\}$ $\mathcal{D}(w') = \{\varepsilon,\eta,\iota\}$
$I = \{john, mary, \mathcal{UT}\}$

Individual Concepts

$john(w) = \alpha$ $john(w') = \varepsilon$
$mary(w) = \beta$ $mary(w') = \eta$
$\mathcal{UT}(w) = \gamma$ $\mathcal{UT}(w') = \iota$

Interpretation of Sortals

δ(Person)(w) = *{john,mary}* δ(Person)(w') = *{john,mary}*
δ(School)(w) = *{UT}* δ(School)(w') = *{UT}*
δ(Student)(w) = *{john}* δ(Student)(w') = *{mary}*

Interpretation of Predicates

δ(enrolled-in)(w) = $\{<\alpha,\gamma>\}$ δ(enrolled-in)(w') = $\{<\eta,\iota>\}$

4.4.2 Quantifying over substantial individuals

In this section, we consider a modification of language L_1 to arrive at a system L_2 whose elements of quantification are ordinary individuals.

The syntax of L_2 is exactly like that of L_1, i.e., every grammatically valid formula in the latter language is a valid formula of the former, and vice-versa. In L_2 all quantification is still restricted by sortals, and sortals and predicates still differ in the fact that whilst in the extension of the latter we have the primitive objects of quantification, in the extension of the former we have individual concepts.

From a semantic point of view, we make the following modifications in the L_2 in comparison to L_1.

Definition 4.22 (model structure)

A model structure for L_2 is defined as an ordered couple $\langle \mathcal{W}, \mathcal{D}\rangle$ where: (i) $\mathcal{W}$ is a non-empty set of possible worlds; (ii) In L_2, the domain $\mathcal{D}$ of quantification is that of *possibilia*, which includes all possible entities independent of their actual existence. Therefore we shall quantify over a constant domain in all possible worlds. Moreover, all worlds are equally accessible and therefore we omit the accessibility relation from the model structure (Fitting & Mendelsonh, 1998).

■

Given a model structure $\mathcal{M}$ $(= \langle \mathcal{W}, \mathcal{D}\rangle)$, the intention of individual constant can be represented by an individual concept. However, due to the nature of the domain D in L_2, we also have to change the definition of the individual concepts adopted in this language.

Definition 4.23 (individual concept)

Let $\mathcal{M} = \langle \mathcal{W}, \mathcal{D}\rangle$. An individual concept i in $\mathcal{M}$ is function from $\mathcal{W}$ into $\mathcal{D}$.

Let i be an individual concept in $\mathcal{M}$ such that i is a constant function. That is to say: for all $w,w' \in \mathcal{W}$, $i(w) = i(w')$. The individual concepts in $\mathcal{M}$ that have this property are named here L_2-individual concepts.

For a given model structure $\mathcal{M}$ we define I as a set of individual concepts defined for that structure and, $I_C \subseteq I$ as the subset of I that includes only L_2-individual concepts.

■

The intention of n-place predicate and of sortal classifier are defined exactly as in L_1, i.e., as an n-ary *property* and a *sort*, respectively. Notice that the following alternative definition for sorts in L_2 can be given.

Definition 4.24 (sort)

Let $\mathcal{M} = \langle W, D \rangle$ be a L_2 model structure. An *intensional property* in $\mathcal{M}$ is a function ℓ from $\mathcal{W}$ into the powerset of individual concepts in $\mathcal{M}$ (i.e., $\wp(I)$). A sort in $\mathcal{M}$ is an intensional property that has as co-domain the set I_C, i.e., a sort is a function from worlds to sets of L_2-individual concepts.

■

Notice that separation becomes a consequence of this definition for sorts.

Proof: Let ℓ be a sort in L_2, and let i,j be two individual concepts such that there are worlds $w,w' \in \mathcal{W}$, such that $i \in \ell(w)$, and $j \in \ell(w')$. Since i and j are L_2-individual concepts, they are constant functions. Thus, if i and j coincide at a world w'', then they coincide in every world.

□

Again, due to the nature of the domain $\mathcal{D}$ in L_2, we also have to change the definitions of $\ell[w]$ and $\ell|[w]|$.

Definition 4.25

$\ell[w] = \{d: d \in \mathcal{D}$ and there is an individual concept $i \in \ell(w)$ such that $i(w) = d\}$.

■

Definition 4.26

$\ell|[w]| = \{d: d \in \mathcal{D}$ and there is an individual concept $i \in \ell(w')$ for some $w \in \mathcal{W}$ such that $i(w') = d\}$.

■

The notion of counterpart becomes dispensable in L_2 for reasons that we shall see in the sequel. A model in L_2 is defined as:

Definition 4.27 (model)

A model in L_2 can then be defined as a triple $\langle \mathcal{W}, \mathcal{D}, \delta \rangle$ such that:

1. $\langle \mathcal{W}, \mathcal{D} \rangle$ is a model structure for L_2;
2. δ is an interpretation function assigning values to the non-logical constants of the language such that: it assigns an L_2-individual concept to each individual constant $c \in T$ of L_2; an n-ary property to each n-place predicate $p \in P$ of L_2; a sort to each sortal classifier $S \in C$ of L_2.
3. The interpretation function δ must also satisfy the following the constraints (3) to (7) from definition 4.17.

■

Suppose that $c \in T$ is an individual constant of L_2. According to definition 4.27 above, the interpretation function δ assigns to c a L_2-individual concept i of the model structure $\langle \mathcal{W}, \mathcal{D} \rangle$. Since i is a constant function, we have that the interpretation of an individual constant c (proper name) is world invariant. This amount to Kripke's thesis that proper names are rigid designators (Kripke, 1982) and conforms to Montague's meaning postulate for common nouns 1 (MP1) (Montague, 1974).

Notice that (even without restriction (7) of definition 4.17) statements of relative identity cannot be expressed in L_2. This is because individual constants are always interpreted as L_2-individual concepts. Thus, if two individual concepts i,j that represent the intention of L_2 constants coincide at a world, they coincide in every world (since they are constant functions), even if they do not belong to the same type. Moreover, notice that if $S \in K$ is a kind, and ℓ the sort assigned to S by δ, then we have that $\ell[w] = \ell|[w]|$, for any $w \in \mathcal{W}$. This is due to: (a) since S is a kind then ℓ is a constant function; (b) every individual concept in ℓ is a constant function. In other words, if S is a kind, then every object which is a possible S is actually an S.

In the sequel we define an L_2 assignment.

Definition 4.28 (L_2 assignment)

An assignment for L_2 relative to a model $\langle \mathcal{W}, \mathcal{D}, \delta \rangle$ is a function that assigns to each variable of L_2 an ordered pair $\langle \ell, d \rangle$, where ℓ is a sort relative to the model structure $\langle \mathcal{W}, \mathcal{D} \rangle$ and $d \in \mathcal{D}$. If a is an L_2 assignment then $a_o(x)$

is the object assigned to variable x by a and $a_S(x)$ is the sortal to which x is bound. Moreover, it is always the case that $a_o(x) \in a_S(x)|[w]|$ for all variables.

An L_2 is almost identical to an L_1 assignment, with the difference that in ordered pair $\langle \ell, d \rangle$ assigned to a L_2 variable by a, $d \in$ D.

■

An assignment a' for L_2 is an ℓ variant of a at x is defined exactly as in definition 4.19. Moreover, the notion of a w' variant of an assignment a relative to w (definition 4.20) becomes dispensable, since in L_2 we no longer have the notions of counterpart.

We can then finally define the truth for formulas α in L_2.

Definition 4.29

Finally, let α be an expression in L_2 and, let *the semantic value of* α *at world w in model M and relative to assignment a* be the value of the valuation function $v^{w}_{M,a}$.

With these definitions, we are able to define the semantics of L_2 as follows:

(a). For all cases described in (a) to (g) in definition 4.21, the valuation function in L_2 has exactly the same value as their respectives in L_1.

(b). If α is the formula □A, then $v^{w}_{M,a}(\alpha) = \text{T}$ iff for all $w' \in \mathcal{W}$ then $v^{w'}_{M,a}(\text{A}) = \text{T}$. Otherwise $v^{w}_{M,a}(\alpha) = \text{F}$.

■

The L_1 rendering of the specification of figure 4.10 presented in the previous section is still a syntactically valid L_2 specification. A L_2 model for this specification is presented below:

$\mathcal{W} = \{w, w'\}$
$\mathcal{D} = \{\Delta, \Phi, \Theta\}$
$I = \{john, mary, \mathcal{UT}\}$

Individual Concepts

$john(w) = \Delta$	$john(w') = \Delta$
$mary(w) = \Phi$	$mary(w') = \Phi$
$\mathcal{UT}(w) = \Theta$	$\mathcal{UT}(w') = \Theta$

Interpretation of Sortals

δ(Person)*(w) = {john,mary}*	δ(Person)*(w') = {john,mary}*
δ(School)*(w) = {UT}*	δ(School)*(w') = {UT}*
δ(Student)*(w) = {john}*	δ(Student)*(w') = {mary}*

Interpretation of Predicates

δ(enrolled-in)*(w) = {<Δ,Θ>}* δ(enrolled-in)*(w') = {<Φ,Θ>}*

4.5 Related Work

4.5.1 OntoClean

The work presented in this chapter has been influenced by the OntoClean methodology, which proposes a number of methodological guidelines to evaluate the conceptual correctness of specialization relationships (Guarino & Welty, 2004, 2002a, 2002b). Despite bearing a strong similarity with OntoClean's useful property types, the stereotypes presented in table 4.1 also hold some important differences. Firstly, rigid sortals in our typology are always considered to be independent. Examples of rigid sortals that are typically considered dependent are universals whose instances are *features* in the sense of (Gangemi et. al, 2003) (e.g., holes, bumps, stains). In the scope of this work, features are considered parts of their hosts (as opposed to *moments* inhering in them, see chapter 6) and therefore do not qualify as dependent entities according to the definition of Guarino and Welty. The relation between a feature and its host is, thus, considered one of *inseparable parthood*. We say, for instance, that a hole in a piece of cheese is an inseparable part of the cheese, not *externally dependent* on it (see chapter 6). Likewise, substantial individuals can have features as essential parts. For instance, the key whole of a locker may be considered an essential part of the locker. The distinctions between inseparable parts/mandatory wholes and essential/mandatory parts are explained in details in chapter 5 of this thesis. Another consequence of this choice is that *categories*, which typically represent abstractions of different kinds, are also always considered to be independent.

A second difference exists regarding the use of the term *mixin*. In OntoClean, mixin is used to denote a combination (conjunction or disjunction) of rigid and non-rigid properties which are subsumed by at least one sortal. Therefore, in this sense mixin is also a sortal. Another example is a combination of the kind CAT and the formal role WEAPON producing the semi-rigid CAT-or-WEAPON. We believe that the category denoted by this sense of the word mixin, although useful in structuring large ontologies, is of little use in conceptual modeling. Conversely, we use the term mixin here according to its widespread use in the object modeling community, i.e. as an abstract type that can classify instances of different classes but which has no direct instances (Booch, 1994). In this sense, mixins include *category*, *role mixin* but also what is called *attribution* in (Welty & Guarino, 2001). Moreover, the stereotype « roleMixin » used here posses the same meta-properties as the category of *formal roles* in OntoClean. As previously mentioned, role mixins include the category of formal roles but it also includes dispersive types which are abstractions of common properties of material roles.

In OntoClean, the notion of dependence applied to roles is a weaker constraint than the one proposed here. In their case, a role classifier is regarded as *notionally dependent* on another classifier. For instance, the roles *Offspring* and *Parent* are notionally dependent on each other and, thus, for an instance *x* of Offspring to exist another individual *y*, instance of Parent, must also exist. The notion of notional dependence used is the one of (Simons, 1997). Here, this constraint is strengthened: *x* and *y* must also be related via a relationship r, where r is the restriction condition for the two roles.

Finally, in Ontoclean, despite of having the same meta-properties, phases are not explicitly required to be defined as partitions of the supertypes they specialize.

4.5.2 The BWW (Bunge-Wand-Weber) Approach

An approach that shares the same objectives as the work presented here appears in (Evermann & Wand, 2001b; Parsons & Wand, 2000; Wand & Storey & Weber, 1999; Weber, 1997; Wand & Weber, 1989, 1993). In these articles, the authors report their results in mapping common conceptual modeling constructs to an upper level ontology. Their approach is based on the BWW ontology, a framework created by Yair Wand and Ron Weber on the basis of the original metaphysical theory developed by Mario Bunge in (Bunge, 1977). In (Evermann & Wand, 2001b), in particular, the authors propose that a UML class should be used to represent a BWW-natural kind (it should be equivalent to functional schema of a BWW-natural kind). A natural kind is defined by Bunge as *"a set of substantial things that share lawfully related properties"*. According to this definition, the authors' proposal for the interpretation of the UML class meta-construct is equivalent to that of kinds and subkinds (i.e. rigid sortals) defined in this chapter. A law is an essential property of its instances (by definition). Since a natural kind is a grouping of things that share these essential properties it is, also by definition, a rigid class. Equating a natural kind with the denotation of a substance sortal concept of substantials is in conformance with other works in the philosophical literature (Lowe, 1989; van Leeuwen, 1991).

As demonstrated throughout this work, kinds constitute a subset of the category types that are necessary to represent the phenomena available in cognition and language. Therefore, a modeling construct representing a kind is only one of a set of modeling constructs that should be available to the conceptual modeler. For this reason, the typology of classifiers presented in this chapter is not only compatible with the proposals of the BWW approach but extends it towards a much richer system of ontological distinctions.

4.5.3 The Approach of Wieringa, de Jong and Spruit

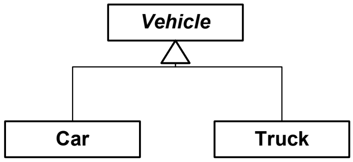
In (Wieringa & de Jong & Spruit, 1995), the authors propose three distinctions in types of taxonomic structures that should be employed to model class migration in object-oriented modeling, namely, *static* and *dynamic subclassing* and *role playing*.

An example of a static taxonomic structure is depicted in figure 4.11. A static structure is always defined as a partition of a superclass that is responsible for supplying a principle of identity for the instances of the subclass. The authors write: *"an instance of a subclass is identical to an instance of a superclass"*. In their approach, the superclass is responsible for generating identifiers for its instances and, hence, for the instances of its subclasses. The authors write that *"object identifiers have a 1-1 relationship with objects and that relation can never be changed"*, i.e., object identifiers are representation of proper names, as here conceived, and also comply with Kripke's requirement of rigidity for proper names. Moreover, instances of the subclass can never migrate to other subclasses in the partition. For instance, a Car can never become a Truck and vice-versa. Since Car and Truck define a partition, ergo, a Car (Truck) can never cease to be a Car (Truck) without ceasing to exist. We can conclude therefore that if we take a *possibilist*[29] view on modality, the extensions of both the superclass and

[29] In a possibilist approach of modal logics, quantifiers range over entities that possibly exist and, therefore, the domain of quantification is constant in every possible world. In actualism, in contrast, quantification considers what actually exist in each world (Fitting & Mendelsonh, 1998).

each of the subclasses in a static partition are *modally constant*. This is, however, not explicitly represented by the authors who take an *actualist* approach. In summary, the superclass and subclasses in a static partition are analogous to what we term here kind and subkind, respectively. Nonetheless, contrary to the authors, we do not require a subkind taxonomic structure to be always defined as a partition of the kind.

Figure 4-11 Example of a static subclassing partition

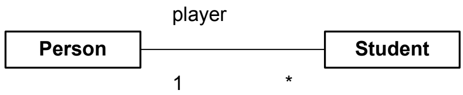


A *dynamic structure* in their approach is identical to what we term here a *phase partition*. Wieringa and colleagues also explicitly forbid the construction of models in which a static partition is followed upwards in the taxonomic structure by a dynamic partition, i.e., a subclass in a dynamic partition cannot be a superclass in a static one. This is equivalent to say, again if we take a possibilist interpretation of their approach, that a phased-sortal cannot subsume a rigid sortal. The justification for this constraint in Wieringa et al. is that it *enhances the intuitive structure of the models and simplifies their formalization*. Since this constraint is equivalent to our postulate 4.3, the philosophical justification provided for the latter in section 4.1 can also serve as a justification for the former.

One of the main differences between the approach of Wieringa et al. and the one proposed here is w.r.t. the category of role universals. In their approach a role universal is not a phased-sortal. Conversely, roles are rigid classifiers whose instances are said to be *played by* instances of ordinary (static and dynamic) types. The *played by* relation (also termed *inheritance by delegation* by the authors) between a role instance *r* and an object *o* implies that r is *specifically existentially dependent* on *o*. This means that r can only be played by o, and can only exist when being played by *o*. However, in contrast, *o* can possibly be associated via the *play by* relation to many instances of the role class (and to many different role classes). Unlike static and dynamic subtypes, role universals are not required to be defined in a partition. Moreover, role universals are responsible to supplying a principle of identity for its instances, which is different from the one supplied by the universals instantiated by their players.

Figure 4.12 depicts an example of ordinary and a role universal according to the Wieringa, de Jong, and Spruit.

Figure 4-12 Example of Role and Role Player Universals



Although mainly motivated by practical implementation issues, a sound philosophical interpretation can be given to role universals and role individuals as conceived by the authors. In chapter 7, we discuss in depth the relation between the notions of role as proposed by Wieringa, de Jong and Spruit and the one proposed in this chapter. We demonstrate that, albeit strongly related, the two notions of role refer to different ontological entities with incompatible meta-properties. Actually, an even stronger statement can be made: role instances in the two approaches do not even belong to the same meta-level ontological category. Our notion of role universal (as well as the other universals discussed in this chapter) refers to the ontological category of *substantial universals*. Role universals, as conceived by the authors, conversely, refer to universals whose instances are individualized properties, i.e., moment universals (see chapter 6).

We demonstrate throughout this thesis that both notions of role are important for the theory and practice of conceptual modeling. However, we have some remarks about the notion of role as formulated by the authors:

1. In discussing the relations between what they term *roles* and *ordinary objects*, Wieringa and colleagues refer to the philosophical logic relation of *coincidence* of objects (see subsection 4.4.2). For example, a student *s* and a person *p* that plays the role of a student in world w are said to coincide at w. In our perspective, since *s* is actually a moment, the relation between *s* and *p* in this case is a relation of *inherence* (see chapter 6) and not one of coincidence. It is important to emphasize that that coinciding objects are not existentially dependent on each other. For instance, in the statue/lump of clay example previously mentioned, their lifecycles are completely independent but merely coincident at some possible worlds;

2. In the way presented by the authors, role individuals should be interpreted as *intrinsic* individualized properties, since they are existentially dependent on a single individual. For instance, in the aforementioned example, the student *s* is dependent on person *p*. However, this is actually not the case. In fact, student *s* depends not only on person *p* but also on the existence of another individual, say *e*, instance of Educational Institution. Thus, *s* should be conceived as a *relational* individualized property (see chapter 6) not an intrinsic one.

This problem becomes clear in our approach due to our *relational* restriction condition for role universals;

3. Since no restriction is defined for the allowed type of a role universal, optional cardinalities must be represented in the authors approach (see figure 4.12). As argued in, for instance, (Weber, 1997; Wand & Storey & Weber, 1999), from an ontological standpoint, there is no such a thing as an optional property and, hence, the representation of optional cardinality constraints leads to unsound models (in the technical sense of chapter 2) with undesirable consequences in terms of clarity. Furthermore, as empirically demonstrated in (Bodard et al., 2001), conceptual models without optional properties lead to better performance in problem-solving tasks that require a deeper-level understanding of the represented domain.

In summary, our approach is compatible with one proposed by Wieringa and colleagues. However, it also complements their approach by elaborating on the distinctions among the category of non-sortal universals, and by discussing the admissible relations between sortals and non-sortals. Moreover, by also considering the notion of roles as a *substantial sortals*, we extend their approach towards a more complete typology of sortals universals for conceptual modeling.

These issues related to role modeling are comprehensively discussed in chapter 7 of this thesis.

4.6 Final Considerations

In this chapter, we present a well-founded theory of universals for conceptual (ontology) modeling. The theory proposed is founded in a number of results in the literature of philosophy of language and descriptive metaphysics and supported by substantial empirical evidence from research in cognitive psychology.

In section 4.2, this theory is used in the definition of a modeling profile comprised of:

(i) a set of stereotypes representing the ontological distinctions on types of substantial universals proposed by the theory (depicted in figure 4.1);

(ii) Constraints on the possible relations to be established between these elements, representing the postulates of the theory.

In chapter 8, we again make use of the theory proposed here to analyze and extend the UML metamodel, in the main case study of this thesis.

The use of the modeling profile proposed here to address relevant modeling problems in conceptual (ontology) representation in this chapter is only briefly illustrated by examples. In chapter 7 of this thesis, the profile is used in to develop a design pattern that captures a solution to a recurrent real-world problem in *role modeling with disjoint allowed types* discussed in the literature. Additionally, in chapter 8, the ontologically well-founded version of UML which is produced in conformance with the theory presented here is employed to analyze and integrate concurrently developed semantic web (lightweight) ontologies in the scope of a context-aware service platform.

In order to formally characterize the ontological distinctions and postulates proposed by the theory, we have presented two extensions to traditional systems of quantified modal logics, namely, the languages L_1 and L_2 discussed in section 4.4. These languages are far from complete and could be extended in many ways. For example, order-sorted extensions in the lines of (Kaneiwa & Mizoguchi, 2004) or of the dynamic database logic presented in (Wieringa & de Jong & Spruit, 1995) could be proposed. Moreover, in an alternative formulation, the constraints imposed to the interpretation functions of L_1 and L_2 could be directly considered in the object language, as in the sortal logic proposed by (Lowe, 1989) or as in the system proposed by (Freund, 2000). In the latter, not only quantification and identity are restricted by sortal concepts but (second-order) quantification over sortal concepts is also part of the language. The objective here, instead, is only to formally characterize in a simpler way the distinction between rigid and non-rigid sortal universals and the important distinction between sortals and non-sortals w.r.t. to the former's exclusive ability to supply a principle of persistence and transworld identity to its instances.

Chapter 5

Parts and Wholes

In this chapter we aim at developing an ontological well-founded theory of conceptual part-whole relations.

Parthood is a relation of significant importance in conceptual modeling, being present in practically all conceptual/object-oriented modeling languages (e.g., OML, UML, EER, LINGO). Although it has not yet been adopted as a modeling primitive in the semantic web languages, some authors have already pointed out its relevance for reasoning in description logics (e.g., Lambrix, 2000). Nonetheless, in many of these languages, the concepts of part and whole are understood only intuitively, or are based on the very minimal axiomatization that these notions require. In addition to that, despite of being an active topic in the conceptual modeling literature, there is still much disagreement on what characterizes this relation and about the properties that part-whole relations should have from a conceptual point of view (Saksena et al., 1998; Snoek & Dedene, 2001; Pribbenow, 2002; Opdahl & Henderson-Sellers & Barbier, 2001; Odell, 1998).

Part-whole relations are also fundamental from a cognitive perspective, i.e., for the realization of many important cognitive tasks (Tversky, 1989), and as a foundation for the formalization of other entities that compose our ontology. For these reasons, a theory of parts and wholes is considered as a fundamental part of any *foundational ontology*, regardless if it is a revisionary or a descriptive metaphysics effort (Bunge, 1977; Masolo et al., 2003a; Heller & Herre, 2004).

Theories of parts (Mereologies) have been a central point of interest since the pre-socratic philosophers and along the years many precise theories have been developed. These formal theories provide an important starting point for the understanding and axiomatization of the notion of part, and for this reason they are discussed in section 5.1. Nonetheless, despite of their importance, there are many controversial properties ascribed to the part-whole relation by these theories that cannot be accepted by cognitive and conceptual theories of parthood. These controversial properties are discussed in section 5.2.

On one hand, formal theories of parts are ontologically extravagant and non-parsimonious, allowing for the existence of entities and for the derivation of transitive relations that are not accepted by cognition. On the other hand, they are too weak to characterize what makes something an integral whole composed of many parts. We thus need a theory of wholes, asides from a theory of parts. For this reason, in section 5.3, we discuss the topic of a *principle of unity* (which relates the parts composing a whole), and its relation to the transitivity of parthood.

Another problematic feature of formal mereologies is the non-differentiation among the roles that parts play within the structure of an aggregate, i.e., perceiving all parts as being of the same type. An important issue in any conceptual theory of parthood is to stipulate the different status that parts can have w.r.t. the whole they compose. For instance: (i) whether objects can share parts; (ii) whether an object *only exists* being part of a specific whole (or of a whole of certain kind); (iii) whether an object *only exists* having a specific object as part (or a part of a specific kind). These different modes of associations, known as *secondary characteristics of parthood*, are discussed in section 5.4.

In section 5.5, we discuss in depth a classical *linguistic* study about meronymic relations, which propose a typology of different types of parthood, as well as a refinement of this study that aims towards a typology of cognitive part-whole relations.

In section 5.6, we demonstrate how the different types of parthood relations elaborated in the typology presented in section 5.5 can be exploited to derive more direct guidelines for prescribing and proscribing transitive parthood relations.

Section 5.7 discusses some relevant related work and section 5.8 elaborates on some final considerations related to the chapter.

5.1 Formal Theories of Parts

The study of parthood relations can be traced back to the early days of philosophy, beginning with the presocratic atomists and continuing throughout the writings of Plato, Aristotle, Leibniz and the early Kant, to cite just a few. The first attempt at a rigorous formulation of the theory was made by Edmond Husserl, (see his third *Logical Investigation* (Husserl, 1970)), but the first completely theory of parts, named *Mereology* (from the Greek μερος, 'part'), was proposed in 1916 by the Polish philosopher Stanislaw Lesniewski (Lesniewski, 1992), who used the part-whole relation as a substitute for the class membership in standard set theory. This theory was later elaborated by Leonard and Goodman in their *The Calculus of Individuals* (Leonard & Goodman, 1940).

Lesniewski's *Mereology* and *The Calculus of Individuals* have been used as a common basis for almost all developments in theories of parthood that we have in formal ontology today. An exception is the *Assembly Theory* developed in (Bunge, 1977). We shall also consider Bunge's theory in the analysis that follows, since it has been applied by the conceptual modeling community as a foundation for the part-whole relation in conceptual modeling.

As conveyed by the name *Calculus of Individuals*, these theories describe relations among *individuals*, irrespective of their ontological nature or the meta-level category to which they belong. In other words, the relata can be as different as material bodies, events, geographical regions or abstract entities. Mereologies are formal (i.e., domain independent) theories, which formally characterize the principles underlying the relations between an entity and its constituent parts, just like set theory formally characterizes the underlying relationships between a class and its members.

The presentation and axiomatization of the different mereological theories in this section is based on (Varzi, 1996, 2003), (Simons, 1987) and (Herre & Heller, 2004).

5.1.1 Core Concepts (Minimal Mereology)

In all philosophical theories of parts, including *Lesniewski's mereology*, the *Calculus of Individuals* and the *Assembly theory*, the relation of parthood stands for a partial ordering, i.e., a reflexive, antisymmetric and transitive relation. This is reflected in the following axioms, in which ≤ symbolizes the relation of parthood:

(1). $\forall x\ (x \leq x)$
(2). $\forall x,y\ (x \leq y) \land (y \leq x) \rightarrow (x = y)$
(3). $\forall x,y,z\ (x \leq y) \land (y \leq z) \rightarrow (x \leq z)$

These axioms amount to what is referred in the literature by the name of *Ground Mereology (M)*, which is the core of any theory of parts.

Taking reflexivity (and antisymmetry) as constitutive of the meaning of 'part' implies regarding identity as a limit case of parthood. A stronger *proper part* relation (<), whereby nothing counts as part of itself, can thus be defined in terms of this one:

(4). $(x < y) =_{def} (x \leq y) \land \neg(y \leq x)$

and consequently the following equivalence holds

(5) $\forall x,y\ (x \leq y) \leftrightarrow (x < y) \vee (x = y)$

The *proper part* (<) relation is a strict partial ordering, i.e., an asymmetric and transitive relation, from which irreflexivity follows:

(6). $\forall x\ \neg(x < x)$
(7). $\forall x,y\ (x < y) \rightarrow \neg(y < x)$
(8). $\forall x,y,z\ (x < y) \wedge (y < z) \rightarrow (x < z)$

In summary, if we interpret the concept of *being a part* as that of *being a proper part*, we have that for all individuals x,y,z:

(i) x is not a part of itself;
(ii) if x is part of y then y is not part of x;
(iii) if x is part of y and y is part of z then x is part of z.

Either the *proper part of* or *improper part of* relations can be taken as primitive, and the choice is more a matter of convenience than anything else. While the former is a more natural concept, the latter is algebraically more convenient.

Another important relation in theories of parts is that of *overlapping*. We say that two individuals *overlap* (•) if they have a part in common. This also includes the case in which one is part of another and also the case of identity. Overlap is, hence, reflexive and symmetric but not transitive:

(9). $(x \bullet y) =_{def} \exists z\ (z \leq x) \wedge (z \leq y)$
(10). $\forall x\ \ (x \bullet x)$
(11). $\forall x,y\ \ (x \bullet y) \rightarrow (y \bullet x)$

When two individuals overlap but neither is a part of the other then we say that they *properly overlap* (°):

(12). $(x \circ y) =_{def} (x \bullet y) \wedge \neg(x \leq y) \wedge \neg(y \leq x)$

The example of a proper overlap depicted in figure 5.1 below makes it clear that overlapping is the mereological counterpart of the intersect relation in set theory.

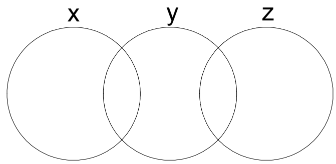


Figure 5-1 Example of a *proper overlap* between individuals x and y, and y and z

If (and only if) two individuals x and y do not overlap, they are said to be *disjoint*. In other words, x and y are disjoint iff they have no part in common (e.g., x and z in figure 5.1). Disjointness (∫) can be defined as follows:

(13). $(x \int y) =_{def} \neg(x \bullet y)$

Like overlapping, disjointness is symmetric

(14). $\forall x,y \ (x \int y) \rightarrow (y \int x)$

The axioms (1-3) define the minimal (partial ordering) constrains that every relation must fulfill to be considered a part of relation. Although necessary, these constraints are not sufficient, i.e., it is not the case any partial ordering qualifies as a parthood relation. For example, the diagram of figure 5.2.a depicts a situation with only two objects x and y, such that y is a proper part of x. This represents a model of the *ground mereology* (in the logical sense), but which can hardly be considered a part of relation, since whenever an object has a proper part, it has more than one, as in the case depicted in figure 5.2.b. The question mark in figure 5.2.a, thus, represents an inexistent proper part of x that would be assumed to exist were y to be considered a part of x

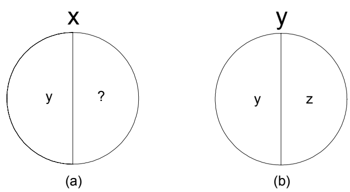


Figure 5-2 (a) Example of a model that does not satisfy the *weak supplementation* axiom (15); (b) Example of a model of *minimal mereology*

Models such as the one of figure 5.2.a are thus excluded by the follow ing additional constraint:

(15). $\forall x,y \ (y < x) \rightarrow \exists z \ (z < x) \wedge (z \int y)$

Formula (15) is termed in the literature the *weak supplementation principle*, and the theory composed by (1-3) and (15) is named the *Minimal Mereology (MM)*. Some authors (e.g., Simons, 1987), regard (15) as constitutive of the meaning of part and, hence, consider (1-3) plus (15) as the minimal constraints that a mereological theory should incorporate.

Traditionally, MM has been extended in two different ways, both of which have been subject to philosophical controversy. These two extensions are discussed in the sequel.

5.1.2 From Minimal to Extensional Mereology

The first extension to MM has been created by strengthening the supplementation principle represented by (15). In this system, (15) is thus replaced by the following stronger supplementation axiom:

(16). $\forall x,y\, \neg(y \leq x) \rightarrow \exists z\, (z \leq y) \wedge (z \int x)$

This axiom states that if an individual y is not a part of another individual x then there is a part of y which does not overlap with x. This includes the cases depicted in figure 5.3. In both figures 5.3.a ($x < y$) and 5.3.b ($y \circ x$), there is a part z of y, which is disjoint (does not overlap) with x. In figure 5.3.c ($y \int x$), every part of y is disjoint with x.

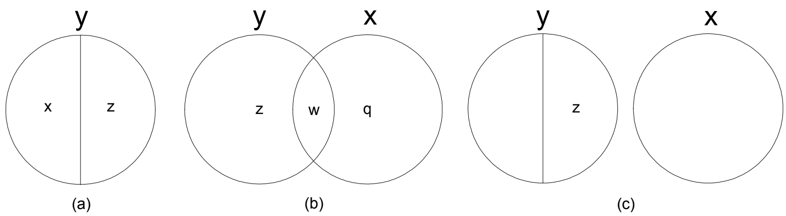


Figure 5-3 Examples of models of *extensional mereology*. In (a), x is a proper part of y; in (b), x and y properly overlap; in (c), x and y are disjoint. In each of these cases, y is not a part of x and, therefore, there is always a part of y that does not overlap with x.

Formula (16) is named the *strong supplementation principle*, and the theory that incorporates (1-3), (15) and (16) is named *Extensional Mereology (EM)*.

If the axioms of ground mereology are assumed (1-3), then (16) implies (15). However, the converse is not true. For instance, the situation represented in figure 5.4 depicts a MM-model that is not a valid EM-model. In this example, x and y are two distinct entities (symbolized by the double line) but that are composed of the same parts. Every part of x (and y) is supplemented by the other one so that (15) is satisfied. However, the same does not hold for (16), i.e., y is not a part of x, but there is no part of y that is disjoint with x.

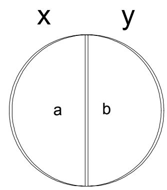


Figure 5-4 Two distinct entities that are composed of the same parts; example of a model of *minimal mereology* that is not a model of *extensional mereology*

The name *Extensional* Mereology is exactly motivated by the exclusion of countermodels such as the one figure 5.4. In fact, the following is a theorem of EM:

(17) $\exists z\ (z < x) \rightarrow (\forall z\ ((z < x) \rightarrow (z < y)) \rightarrow (z \leq y))$

from which it follows that

(18) $\exists z\ (z < x) \vee (z < y) \rightarrow ((x{=}y) \leftrightarrow \forall z\ ((z < x) \leftrightarrow (z < y)))$

Theorem (18) states that two objects are identical iff they have the same (proper) parts[30]. This is the mereological counterpart of the extensionality principle in set theory, which states that two sets are identical iff they have the same members. The philosophical controversy of EM arises exactly because of this axiom. In a *multiplicationist* ontology (Masolo et. al, 2003a) such as the one proposed here, we can have continuants that share the same parts but which are not identical. To use an example already mentioned in chapter 4, a statue and a lump of clay can be composed of the same parts. They are, nonetheless, diverse, since they possess incompatible meta-properties. We return to this point in section 5.2.2.

5.1.3 From Extensional to Classical Mereology

A second way that MM has been extended is with the aim to provide a number of closure operations to the mereological domain. As discussed, for example, in (Varzi, 1996, 2003) and (Simons, 1987), theories named *CMM (Closure Minimal Mereology)* and *CEM (Closure Extensional Mereology)* can be obtained by extending MM and EM with the following operations:

a) **Sum:** also named *mereological fusion* (or *juxtaposition* in the case of Assembly Theory). The sum z of two objects x and y, symbolized as Sum(z,x,y), is the entity such that every object that overlaps with z, overlaps either with x or with y (or with both);

[30] An analogue for the improper part-of is already provable in M due to the reflexivity and antisymmetry of ≤.

(19) Sum(z,x,y) $=_{def} \forall w((w \bullet z) \leftrightarrow ((w \bullet x) \vee (w \bullet y)))$

b) **Product:** also named *superposition* (in Assembly Theory). The product of two objects x and y is the entity z such that every part of z is either part of x or y;

(20) Pro(z,x,y) $=_{def} \forall w((w \leq z) \leftrightarrow ((w \leq x) \wedge (w \leq y)))$

c) **Difference:** the difference of two objects x and y is the entity z such that every part of z is part of x and does not overlap with y;

(21) Dif(z,x,y) $=_{def} \forall w((w \leq z) \leftrightarrow ((w \leq x) \wedge \neg(w \bullet y)))$

d) **Complement:** the complement of an entity x is the entity z such that every part of z does not overlap with x;

(22) Comp(z,x) $=_{def} \forall w((w \leq z) \leftrightarrow \neg(w \bullet x))$

These operations are the mereological counterpart of the set theoretical operations of *union*, *intersection*, *set difference* and *complement of a set*, respectively. In the presence of the extensionality principle, the z's that are the results of these operations are unique. Thus, for example, in an extensional mereology, if two objects x and y overlap then there is a unique entity z that is composed of the common parts of x and y. In particular, the existence of the product and difference of two individuals x and y, and of the complement of an individual x, are only guaranteed in certain cases. These conditions are expressed in formulas (23-25) below, respectively.

(23) $\forall x,y\ (x \bullet y) \rightarrow \exists z \forall w((w \leq z) \leftrightarrow ((w \leq x) \wedge (w \leq y)))$
(24) $\forall x \exists y\ \neg(y \leq x) \rightarrow \exists z \forall w((w \leq z) \leftrightarrow ((w \leq x) \wedge \neg(w \bullet y)))$
(25) $\forall x \exists y\ \neg(x \bullet y) \rightarrow \exists z \forall w((w \leq z) \leftrightarrow \neg(w \bullet x))$

The mereological sum (Sum), conversely, is guaranteed by the presence of an entity, termed the Universe of which everything is part (Simons, 1987):

(26) Universe(z) $=_{def} \forall x\ (x \leq z)$

Once more, in an extensional mereology, the universe z is unique. The existence of a "null individual" that is part of everything would also guarantee the existence of the product and difference for any two individuals, and the existence of a complement for any individual. However, most theories do not define such an entity. An exception is Bunge's Assembly theory (Bunge, 1977).

As demonstrated in (Simons, 1987), if the product operator is functional then the strong supplementation axiom (16) is implied by the weak supplementation principle (15). As a consequence, CMM and CEM collapse in one single theory.

Finally, traditionally, unrestricted operations of fusion and product are also defined for closure mereologies. For the (unrestricted) *mereological sum* we define the following formula schema:

(27) $\exists x F(x) \rightarrow \exists z \forall y((y \bullet z) \leftrightarrow \exists w(F(w) \wedge (y \bullet w))$

This expresses that for every satisfied predicate F there is an entity consisting of all those things that satisfy F or, to put it differently, z is the sum of the arbitrary non-empty set of entities w_i such that $F(w_i)$ holds. Once more, in the presence of extensionality (16), the entity which is the sum of all entities satisfying predicate F has its uniqueness guaranteed. In this case, we can define a general sum as:

(28) $\sigma x F(x) =_{def} \iota z \forall y((y \bullet z) \leftrightarrow \exists w(F(w) \wedge (y \bullet w))$

The product of all members of a set G of overlapping objects can be defined as follows: Let W be set of all those things that are part of every member of set *G*, i.e.,

(29) $\forall x\ W(x) \leftrightarrow \forall y\ (G(y) \rightarrow (x \leq y))$

The unrestricted product of all members of set G can, hence, be defined as the sum of all members of W, that is, by replacing F for W in the formula schema (27). The result of adding schema (27) to CMM or CEM is a theory named *GEM (General Extensional Mereology)* or *Classical Extensional Mereology*.

As demonstrated, for instance, in (Varzi, 2003), all closure operations (19-22) can be defined via choice of suitable predicates F to be substituted in (27). This gives the full strength of GEM, which has the algebraic structure of a quasi-boolean algebra (boolean algebra with a zero element removed)[31].

Figure 5.5 below represents schematically the logical space of these different mereological theories.

---

[31] In the case of Bunge's Assembly theory, it is as expressive as a complete Boolean lattice.

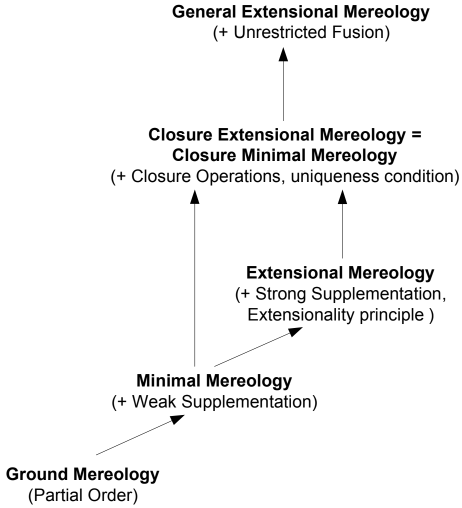


Figure 5-5 Relations between different mereological theories. The arrows go from the weaker to the stronger theory

A final way to extend the theories in figure 5.5 is by considering the issue of *Atomism*. A *mereological atom* is an entity that has no proper parts:

(30) $\mathbf{At(x) =_{def} \neg\exists y\ (y < x)}$

In an atomistic mereology everything has atomic parts, i.e.,

(31) $\mathbf{\forall x\ \exists y\ At(y) \wedge (y \leq x)}$

Conversely, in an atomless mereology the following axiom holds:

(32) $\mathbf{\neg\exists x\ At(x)}$

Formulas (31) and (32) are clearly incompatible, but taken in isolation they can be added to any of the theories depicted in figure 5.5. In other words, the question of atomism is ortogonal and compatible to any of the mereologies discussed so far. Adding (31) to a theory X yields its atomistic version AX, whereas adding (32) yields its corresponding atomless version $\underline{A}X$.

Finally, it is important to emphasize two important features of atomistic mereologies:

1. Atomistic mereologies (and only them) admit finite models, i.e., decomposition of parts must eventually come to an end;

2. When atoms are considered, important simplifications can be made to many of the axioms discussed so far (Varzi, 2003). For instance, the *weak supplementation* principle (15) can be replaced by

(33) $\forall x,y\ (y < x) \rightarrow \exists z\ At(z) \wedge (z < x) \wedge \neg(z < y)$

5.2 Problems with Mereology as a Theory of Conceptual Parts

Mereology has shown itself useful for many purposes in mathematics and philosophy (Varzi, 1996; Simons, 1987). Moreover, it provides a sound formal basis for the analysis and representation of the relations between parts and wholes, and among parts that compose a whole, regardless of their specific nature. However, as pointed out by (Gerlst & Pribbenow, 1995) and (Pribbenow, 2002) (among other authors), it contains many problems that make it hard to directly apply it as a theory of conceptual parts. As it shall become clear in the discussion that follows, on one hand the theory is too strong, postulating constraints that cannot be accepted to hold generally for part-whole relations on the conceptual level. On the other hand, it is too weak to characterize the distinctions that mark the different types of conceptual part-whole relations.

We shall carry out our discussion in a bottom-up approach w.r.t. figure 5.5, i.e., we first discuss the problems of the basic axioms of Ground Mereology, and then continue upwards in the graph discussing the problems of stronger theories.

5.2.1 Conceptual Problems in Ground Mereology

Let us take the proper-part relation, which better reflects our common sense notion of part. According to the axioms (6-8), proper parthood is an irreflexive, antisymmetric and transitive relation. Irreflexivity and antisymmetry are generally accepted as meta-properties that all (proper) part-whole relations should have. However, the same does not hold for transitivity. Take as examples the following cases:

(i) The hand is part of the arm
The arm is part of the person
Ergo, the hand is part of the person

(ii) A person is part of the KR (Knowledge Representation) group
The KR group is part of the AI (Artificial Intelligence) group
Ergo, the person is part of the AI group

In these two examples, transitivity holds without a problem. However, putting (i) and (ii) together, the following fallacious argument can be constructed:

(iii) The hand is part of the person
The person is part of the KR group
(?) The hand is part of the KR group

Another example where transitivity fails is the following:

(iv) Eschede is part of The Netherlands
The Netherlands is part of the European Union
(?) Enschede is part of the European Union

In (iii) and (iv), the problem arises because the mereological parthood relation fails in taking into account the different roles that parts play within the whole. In order to do this, we must complement mereology (the theory of parts) with a *theory of wholes* (Gangemi et al., 2001), in which the relations that tie the parts of a whole together are also considered. In section 5.3, we use a theory of integral wholes (Simons, 1987) to address the problem of transitivity in conceptual part-whole relations. We advocate that conceptual parthood relations should not be interpreted in an unconstrained manner, but always in regards to a specific *context*, and that these contexts demarcate the scopes in which transitivity can be guaranteed to hold. Moreover, we claim that understanding and defining what a context is amounts to the same task as understanding what makes something an integral whole. Finally, as we demonstrate in section 5.5, there is not just one, but many distinct notions of conceptual parts, and the issue of transitivity must be considered differently for each of these notions.

Finally, something that becomes clear in section 5.3 is that, although we deal with atomistic theories, the notion of atom as defined in the classical theories is too coarse. When defining contextual parthood, asides from absolute atoms, we need the notion of atoms relative to a given context or level.

5.2.2 Conceptual Problems in Extensional Mereology

The problem with extensional mereologies from a conceptual point of view arises from the introduction of the strong supplementation principle and, consequently, of formula (16) which states that objects are completely defined by their parts. In EM (and all its extensions), two objects are identical iff they have the same (proper) parts. This is the mereological counterpart of the extensionality principle in set theory, and the problems associated with extensionality in part-whole relations from a conceptual point of view are similar to those associated with equating universals with sets in classical nominalistic theories of universals (Armstrong, 1989).

Take, for example, an organization A composed by the members John, Peter and Mary. Now suppose a different organization B, composed of exactly the same members. According to (16) the following can be observed:

(i). A and B are identical;
(ii). If John leaves organization A then the organization A' whose members are now only Peter and Mary is not the same organization as A;
(iii) since John did not leave organization B, organizations A and B are identical in world w but not in world w'.

In parity with authors such as (Gerlst & Pribbenow, 1995), we believe that these conclusions are unacceptable. First, we can easily imagine two different organizations which share the same members. Imagine the situation in which John Lennon, Paul McCartney, George Harrison and Ringo Star are also the members of the indoor football team Liverpool F.C. The Beatles and the Liverpool F.C. are clearly different individuals despite of sharing the same parts. Not only they might obey completely different principles of identity but also their members play completely different roles within the internal structure of the composite. Using the same example, (ii) is clearly false for the case of The Beatles, since Ringo Star replaced Pete Best without spawning the creation of a new entity (and destruction of the former). Finally, (iii) is simply unacceptable because it contradicts Leibniz's Law (see chapter 4), which is accepted as an axiom in practically all theories of identity (van Leeuwen, 1991).

Examples such as this one can be found for many non-identical entities that happen to coincide in a certain world (e.g., the statue and lump of clay example previously discussed). In summary, extensional theories differentiate entities that in common sense we deem as the same and equate entities that we deem different. Moreover, it rejects the (strongly supported) claim defended in chapter 4 that there are several distinct principles of identity that are employed by human cognition. To accept extensionality amounts to the same as accepting the existence of one single principle of identity, namely, the *extensionality principle.* As discussed in section 4.3, the only period in which humans employ a unique principle of individuation and identity is in the pre-language period when they are younger than 9-months old. In this period, infants' judgment of identity statements is supported by the unique sortal *"maximally-self-connected-physical-object"*, i.e., the notion of an object with clear boundary contrasts with its background and whose parts move along together with the whole (van Leeuwen, 1991, part 3). Perhaps, extensionality holds for this primitive notion of physical object as well as for *amounts of matter* (see section 5.5.1 of this chapter) but it certainly does not hold in general for substantials. Furthermore, as demonstrated by (Xu, 2004), this initial object-based individual system is overridden later in human cognition by a kind-based system with a multitude of sortal-supplied principles of identity.

A different perspective on this problem is that extensionality considers all parts of an entity as essential, i.e., an entity is equal to the mereological sum of its parts, thus, changing any of its parts changes the identity of that entity. Ergo, an entity cannot exist without each of its parts, which is the same as saying that all its parts are essential. This conclusion is clearly false for many objects accounted by common sense. For example, as a person, there are many parts that I can loose without ceasing to be the same. For instance, during a person's lifetime one typically looses hair, teeth and nails. There are also more tragic cases when one can loose a member such as a finger, an arm or a leg. In fact, during a lifetime a person changes all cells in her body! Nonetheless, none of these changes alter the identity of the individual in question. This is not to say that there are no essential parts in the human body. For instance, one cannot survive without a heart or a brain. Additionally, despite that both relations person/heart and person/brain constitute ontological dependence relations, they are of quite diverse nature. Whilst in the former, we have a case of generic dependency; the latter exemplifies a case of specific existential dependency that characterizes real *essential parts*.

In summary, while some parts of a common sense object are essential, not all of them are essential. This topic is thoughtfully discussed in section 5.4.2.

5.2.3 Conceptual Problems in General Extensional Mereology

From a conceptual point of view, the problem with the theory of General (Classical) Extensional Mereology is related to formula schema (27), which defines the existence of a sum (or fusion) for any arbitrary non-empty (but non-necessarily finite) set of entities. Just as in set theory one can create a set containing arbitrary members, in GEM one can create a new object by summing up individuals that can even belong to different ontological meta-categories. For example, in GEM, the individual Θ created by the sum of Noam Chomsky's left foot, the first act of Puccini's Turandot and the number 3, is an entity considered as legitimate as any other.

In the literature, this feature of GEM is termed *ontological extravagance*, referring to the commitment of the theory to the existence of a wealth of entities that are utterly counterintuitive and that have no place in human cognition. In addition, GEM is also deemed to be *ontologically exuberant*, since it dramatically increases the number of entities to be included in the inventory of the domain (Varzi, 2003).

As argued by (Pribbenow, 2002), humans only accept the summation of entities if the resulting mereological sum plays some role in their conceptual schemes. To use an example cited by Pribbenow: the sum of a frame, a piece of electrical equipment and a bulb constitutes an integral whole that is considered meaningful to our conceptual classification system. For this reason, this sum deserves a specific concept in cognition and name in human language. The same does not hold for the sum of bulb and the lamp's base. Another example of meaningful sums occurs when entities are summed up to provide a target for a linguistic plural reference such as in "two of the children of my sister".

A way to understand why only meaningful sums are accepted by cognition is to recall the purpose of classification in our conceptual system. As discussed in chapter 4, the ability to categorize is central for our construal of the environment, and without it, learning would be hardly possible. For example, if I am able to recognize that *Felix* is an instance of the category Cat, then I am able to infer that *it* has all the properties that I know that apply to cats. Additionally, all properties that I come to learn about *Felix* can be carried out to other cats that I may encounter. Finally, as point out by (Milikan, 1998; Mcnamara, 1986; Schuun & Vera, 1995), our cognitive system seems to be specially tuned to form conceptual categories that are based on a common *essence*, supported by a theory-based explanation for their principle of application (see section 5.8). This is because these are the categories that provide the most useful sources for inductive knowledge. In this perspective, what is the point for our cognitive system to recognize the existence of entities as unique and alien as the mereological sum Θ aforementioned? What properties can we expect Θ to have? What can we learn about it that can be carried to other individuals?

For this reason, we advocate that a theory of conceptual part-whole relations should forbid the construction of unrestricted sums, and formula schema (27) should only be defined for predicates that represent genuine universals (see discussion on chapter 6). In contrast, the notions of *integrity* and *unifying condition* that comprise a theory of wholes should be taken seriously. This is to say that a theory of conceptual parts and wholes should only countenance the existence of composite objects that are unified by bona fide unifying conditions. These issues are discussed in depth in section 5.3 below.

5.3 Integral Wholes

As we have previously discussed, one of the major conceptual problems with Classical Extensional Mereology comes from the generalized fusion axiom, which allows for the existence of a sum (or fusion) for any arbitrary non-empty set of entities. For example, it allows for the definition of an aggregate composed by the state of California and the number 3. A question that comes to the mind is: If these entities are unobjectionable from a formal point of view, why are they deemed unnatural by common sense? Or, equivalently, what makes an aggregation such as car, a football team, a forest, a human body, to be accounted as a *natural conceptual whole*?

According to (Simons, 1987), the difference between purely formal ontological sums and, what he terms, *integral wholes* is an ontological one, which can be understood by comparing their existence conditions. For sums, these conditions are minimal: the sum exist just when the constituent parts exist. By contrast, for an integral whole (composed of the same parts of the corresponding sum) to exist, a further *unifying condition* among the constituent parts must be fulfilled.

The distinction between a mere sum and an integral whole also appears in (Bunge, 1977, 1979) in the form of the distinction between, what he terms, a *mere aggregate* and a *system*. Bunge defines a mere aggregate as "a compound thing, the components of which are not coupled, link, connected or bounded [, and which…] therefore lacks integrity or unity" (Bunge, 1979, p.4). In contrast, the components of system are "interrelated rather than loose".

Let us start by defining the Bungean concept of a *link*, *coupling* or *bond* among two things. Bunge explicitly states that "[w]e must distinguish between a mere relation, such as that of being older, and a connection, such as that of exerting pressure.… [T]wo things are *connected* (or *coupled*, or *linked*, or *bounded*) if at least one of them acts upon the other…where the action need not consist in eventuating something but … one thing acts upon the another if it modifies the latter history" (p.6).

Bunge emphasizes the distinction between *the composition of a system* and the merely formal notion of composition: "A social system is a set of socially linked animals. The brains of such individuals are parts of the latter but do not qualify as members of components of a social system because they do not enter independently in social relations: only entire animals can hold social relations. In other words, the composition of a social system is not the collection of its parts but just the set of its *atoms*, i.e., those parts that are *socially connectable*" (ibid., p.5).

The relative composition of a system can thus be defined as follows:

Definition 5.1 (relative composition): Let S be the set of substantial individuals, and let the composition of a system $x \in S$ be defined as

$$\textbf{(34).}\ \mathcal{C}(x) = \{y \mid y \leq x\}.$$

Then the A-composition of a system x is defined as

$$\textbf{(35).}\ \mathcal{C}_A(x) = \{y \in A \mid y \leq x\},$$

or simply, $\mathcal{C}_A(x) = \mathcal{C}(x) \cap A$. The set $A \subseteq S$ is said to contain the A-atoms of *x*.

■

Although this notion conceptualizes an important intuition, it begs some important clarifications. To start of with, Bunge is not explicit on how to define the set of atoms A. However, the text makes clear that the elements of A must be those elements in the world that participate in certain *bonding* relations. He uses examples such as: the molecular composition of a mass of water is the set of H_2O molecules that are part of that mass, which, hence, excludes individual Hydrogen and Oxygen atoms. Alternatively, one could say that the molecular composition of a mass of water is composed by the set of parts of that mass that can engage in molecular relations. An analogous argument can be made in the case of a social system aforementioned: the set of social atoms of a system is composed by those parts of the system that can engage in social relations. For instance, the social composition of a school is the set of staff and pupils, which are held together by relations of teaching and learning, managing and being managed, among others (Bunge, ibid., p.5).

Some important notions that can be used to provide a foundation for the set of A-atoms of a system have been proposed by the philosopher Peter Simons in (Simons, 1987), namely, the notions of a *R-closed*, *R-connected* and *R-closure system*.

Definition 5.2 (R-closed): A set B is closed under a relationship R, or simply R-closed, iff

$$\textbf{(36).}\ \mathbf{cl}\,\langle R\rangle\, B =_{def} \forall x\, (x \in B) \rightarrow ((\forall y\, R(x,y) \vee R(y,x) \rightarrow (y \in B)),$$

i.e., the relationship R does not cross B's borders in any direction.

■

Definition 5.3 (R-connected): A set B is connected under the relation R, or simply, R-connected iff

$$(37).\ \mathbf{con}\ \langle R\rangle\ B =_{def} \forall x\ (x\in B) \rightarrow (\forall y\ (y\in B) \rightarrow (R(x,y) \vee R(y,x)),$$

i.e., every member of B bears a relation with another member of B.

■

Definition 5.4 (R-closure system): A set B is a closure system under the relation R, or simply, R-closure system iff

$$(38).\ \mathbf{cs}\ \langle R\rangle\ B =_{def} (\mathbf{cl}\ \langle R\rangle\ B) \wedge (\mathbf{con}\ \langle R\rangle\ B),$$

i.e., iff it is both connected and closed under R.

■

Definition 5.5 (A-parthood): Let the set of A-atoms of a system x be defined in terms of a relationship R, such that *A* is a set of parts of x that form a closure system under R. In this case, x is said to be an *integral whole unified under relationship R*, and R is termed a *characteristic relation* or *unifying condition* for x. Therefore, we can finally define a parthood relation relative to A, or simply, an A-part as

$$(39).\ (y \leq_A x) =_{def} (y \in \mathcal{C}_A(x)).$$

■

An example of an integral whole is depicted in figure 5.6 below. The unifying relation R, in this case, is the relation of *carrying out research in the area same area in the University of Twente*. In this figure, the parts of the Distributed Systems Research Group (DSRG) that are not R-connected are excluded from its relativized composition (e.g., all the anatomical parts of John, the cellular parts of these anatomic parts, the atomic parts of these cellular parts, etc.).

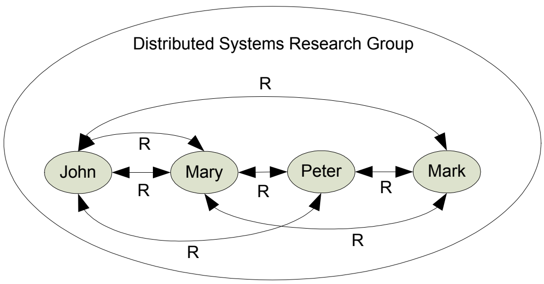
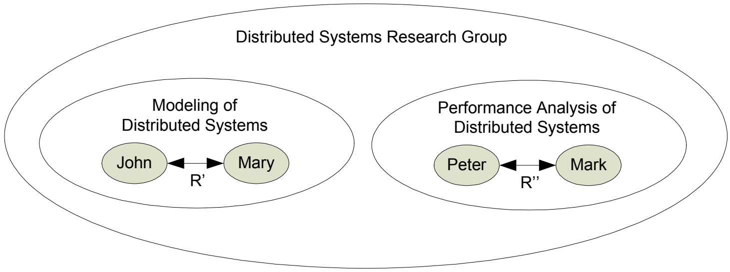


Figure 5-6 Example of an *integral whole* unified by a characteristic relation R

Bunge also provides no explanation for why the members of A are to be considered atoms. However, according to the interpretation just given for the origin of set A, its members must indeed be A-atomic, i.e., if $\forall x\ (x \in A) \rightarrow \neg\exists y\ (y <_A x)$[32]. This is because, the whole x unified under R is maximal under this relation, by the definition of an R-closure system. As a consequence, no member of A can be unified under the same relation R.

This is far from saying that they must be atomic w.r.t. the formal parthood relation. In fact, there can be other relations R' and R'', such that they can be used to form other closure systems among the subsets of A. Suppose that R' is the relation of *carrying out research in the same sub-area of the area of distributed systems in the University of Twente,* we can then form the set B which is an R'-closure system. Since every member of B is a formal part of the Distributed Systems Research Group (DSRG), we can say that the individual *Modeling of Distributed Systems Research Group (MDSRG)* is an integral whole unified under R', and also a part of DSRG. An analogous procedure can be made for relation R'' creating the individual *Performance Analysis of Distributed Systems Research Group (PADSRG)*. This situation is depicted in figure 5.7.

Figure 5-7 Refinements of a unifying relation R providing further structure within a whole



The set of B-atoms under R' is {John, Mary}, and the set of C-atoms under R'' is {Peter, Mark}. Now, let us redefine the set of A-atoms under R as {John, Mary, Peter, Mark, MDSRG, PADSRG}. It is clear that DSRG can be unified under R, since: (i) every member of A is part of DSRG; (ii) every member of A is connected under R (e.g., John carries research in the same area as the MDSRG, which in turn, carries research in the same area as the PADSRG). The relations R' and R'' are specializations of R and, thus, the sets B, C and A defined as closure systems of R', R'' and R, respectively, are such that $B \subseteq A$ and $C \subseteq A$.

Now, it is clear that $\neg$(John $<_A$ MDSRG), since MDSRG is an A-atom. However, it is true that (John $<_B$ MDSRG), (MDSRG $<_A$ DSRG) and (John $<_A$ DSRG). In fact, it is generally the case that if a characteristic relationship R' is an specialization of another characteristic relationship R,

[32] The relative proper parthood can be defined as usual, i.e., $(y <_A x) =_{df} (y \leq_A x) \wedge \neg(y = x)$.

and if the sets B and A are unified under R' and R, respectively, then it is true that

(40). $\forall x,y,z\ (x <_B y) \wedge (y <_A z) \rightarrow (x <_A z)$.

Proof: Suppose that $\neg(x <_A z)$. Since, $(x <_B y)$ we have that $(x < y)$ and $(x \in B)$, by the definition of $<_B$. Likewise, since $(y <_A z)$ then $(y < z)$ and $(y \in A)$. From $(x < y)$ and $(y < z)$ we have $(x < z)$, since formal parthood is always transitive. Moreover, since R' is a unifying subrelationship of R then $B \subseteq A$. Therefore, $(x \in A)$, and, consequently, $(x \in \mathcal{C}_A(z))$, by definition 5.1. Finally, we have that $(x <_A z)$, which contradicts the hypothesis.

□

We can now define the concept of a context as follows:

Definition 5.6 (Context): Let $(R_i \prec R_j)$ mean that if I is a R_i-closure system and J is an R_j-closure system then $(I \subseteq J)$. Let $(R_1 \prec \ldots \prec R_n)$ be a series of unifying relations, and let $(\leq_1,\ldots,\leq_n)$ be a series of relative part-whole relations defined such as in definition 5.5. We name $(\leq_1,\ldots,\leq_n)$ a context, and, by a generalization of (40), we have that

(41). $\forall x,y,z\ (x \leq_i y) \wedge (y \leq_{i+1} z) \rightarrow (x \leq_{i+1} z)$.

■

According to definition 5.6, transitivity holds generally within a context. However, for some types of part-whole relations there are no possible unifying subrelationships and, therefore, it is senseless to discuss contextual transitivity. For instance, the relationship *being-a-direct-functional-part*, by definition, cannot have a unifying subrelationship. Otherwise, it would have an "indirect direct-functional-part".

The discussion on the problem of transitivity carried out in this section is just a preliminary account of the relation between unifying relations for integral wholes and the transitivity of their parts. In sections 5.6 and 7.4, we are able to elaborate on this discussion by considering other distinctions of our ontological theory.

5.4 Secondary Properties of Part-Whole relations

In this section, we focus on some important aspects of conceptual part-whole relations, which are not taken into account in classical mereological theories. Besides the classical notion of parthood defined, for example, by the axioms of Minimal Mereology, there are other axiom groups that can be used to formally characterize further ontological distinctions among part-whole relations. Two of these so-called *secondary characterisitics* (Opdahl & Henderson-Sellers & Barbier, 2001) are discussed as the following subsections. In subsection 5.4.1, we discuss *shareability*, whereas *separability* is discussed in 5.4.2.

5.4.1 Shareable Parts

Shareability is considered a secondary characteristic of part-whole relations, in the sense that it is not the case that it is a property held by all relations of this type. In contrast, it can be used as *differentiae* among distinct types of parthood. Shareability is simpler to understand than the other secondary characteristics discussed in section 5.4.2, since it does not involve modal properties. A first tentative definition of a non-shareable (exclusive) part is given as follows:

Definition 5.7 (exclusive part – first version): An individual x is said to be an exclusive (proper) part of another individual y (symbolized as $x <_X y$) iff (i) x is a (proper) part of y; (ii) for every z such that x is a (proper) part of z then either: (a) z is a (improper) part of y or (b) y is a (improper) part of z, or (c) both, i.e., y and z are identical.

(42). $(x <_X y) =_{def} (x < y) \land (\forall z\ (x < z) \rightarrow (z \leq y) \lor (y \leq z))$ ■

From *x being an exclusive part of y* it does not follow that x is not part of anything else, as it is sometimes wrongly stated in the literature of conceptual/object-oriented modeling. This is because, due to transitivity, x must also be part of everything y is part of.

An example of an exclusive parthood is represented in figure 5.8. In this picture, every instance of an engine is an exclusive part of a car, i.e., if the engine is also part of some other object (e.g., the power system of the car, not represented) then this object must also be part of the car, or it must have the car as one of its parts.

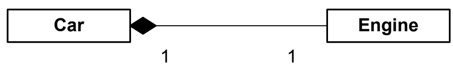


Figure 5-8 An exclusive (i.e., non-shareable) parthood relation represented by the UML composition meta-construct

At first, definition 5.7 seems to suitably represent what it is meant by exclusiveness of parthood in conceptual modeling. However, on second thought, it becomes clear that it is too restrictive to figure as the only definition of exclusiveness to be employed. Take, for instance, the model of figure 5.9. This is an example that represents a recurrent pattern in conceptual models. Since the identity of the roles researcher and family member are supplied by the substance sortal person that they both subsume, it can certainly be the case that the same individual, say the person John Smith, is both a researcher and a family member and, therefore, is part of both a family and a research group. The logical model corresponding to this specification would be excluded by definition 5.7.

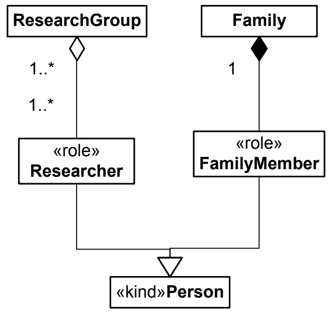


Figure 5-9 Conflict in the notion of exclusiveness. Examples of exclusive and shareable parthood relations represented by the UML meta-constructs of composition (black diamond) and aggregation (white diamond), respectively.

In order to remedy this situation, we propose a much weaker type of parthood exclusiveness, more adequate for the purposes of conceptual modeling. In the definition that follows, we use the notation x::U to represent the relation of instantiation between an individual x and a universal U.

Definition 5.8 (exclusive part): An individual x of type A is said to be an exclusive (proper) part of another individual y of type B (symbolized as $<_X$(x,A,y,B)) iff y is the only B that has x as part.

(43). $<_X(x,A,y,B) =_{def} (x::A) \wedge (y::B) \wedge (x<y) \wedge (\forall z\ (z::B)\ (x < z) \rightarrow (y = z))$

■

This type of exclusive part-whole relation can be also defined between the universals A and B as a general exclusive part-whole relation.

Definition 5.9 (general exclusive part-whole relation): A universal A is related to a universal B by a relation of general exclusive parthood (symbolized as A $<_{GX}$ B) iff every instance x of A has an exclusive part of type B.

(44). $A<_{GX}B =_{def} \forall x\ (x::A) \rightarrow \exists y\ (y::B) \wedge <_X(x,A,y,B)$

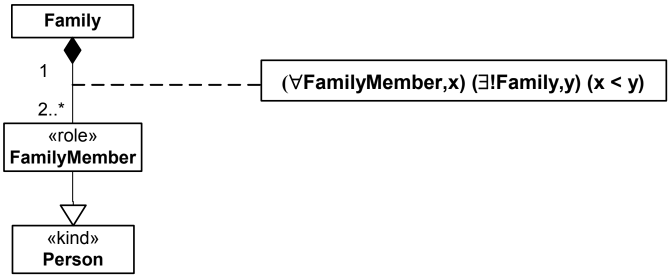
or simply,

$$\textbf{(45). } A <_{GX} B =_{def} \forall x\ (x::A) \rightarrow \exists! y\ (y::B) \wedge (x < y)$$

■

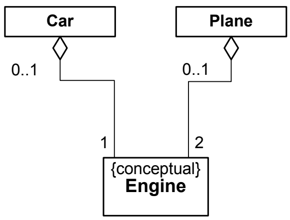
Figure 5.10 shows part of the example of figure 5.9 by using definition 5.9. In the specification of figure 5.10, we employ a language with sortal restriction such as the languages L_1 and L_2 defined in section 4.4.

Figure 5-10 Exclusiveness defined in the type level and corresponding axiomatization



In the articles (Saksena & France & Larrondo-Petrie, 1998) and (Saksena et al., 1998), Monika Saksena and colleagues have proposed a different notion of shareability, which they term *conceptual shareability*. This idea is exemplified in figure 5.11, which is an exact copy from the original model in (Saksena et al., 1998)[33].

Figure 5-11 Conceptual shareable parthood relation (from Saksena et al., 1998)

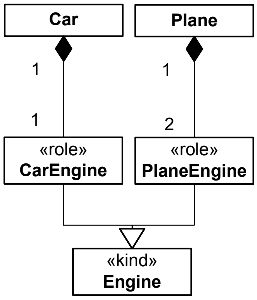


The intention behind the model of figure 5.11 is to represent that although an instance of an engine cannot be part of both a car and a plane, "the concept represented by the class is shared" (Saksena et al., 1998, p.4). Here we defend the idea that this alleged type of shareability is not only superfluous, but also stimulates the creation of poorer conceptual models. As we have discussed in section 4.5.3, conceptual models without minimum cardinality constraints equal to zero can be considered more suitable both in terms of ontological adequacy, and of practical performance in problem-solving tasks. In a situation such as the one

[33] The tagged value {conceptual} in the picture is a proposal of (Saksena et al., 1998).

depicted in figure 5.11, if an engine is built to be part of a car then it is a car engine, if it is built to be part of a plane, then it is a plane engine. Additionally, if the same engine type can be used both in cars and planes then, for every individual engine, being part of a car (or plane) is merely a contingent fact. In other words, every engine that is part of a car in world w could become (or have been) part of a plane in a world w' (or vice-versa). Therefore, in the latter case, both concepts car engine and plane engine must be conceived as roles that an engine can play in certain circumstances. In figure 5.12, we present a version of this latter interpretation of figure 5.11, in which these roles, which are implicit in figure 5.11, are made explicit.

Figure 5-12 The alleged *Conceptual Sharability* relation modelled by real non-shareable relations and explicitly represented roles



5.4.2 Essential and Mandatory Parts

As discussed by (Simons, 1987), there are many issues regarding part-whole relations that cannot be clarified without considering *modality*. One of these issues is the secondary characteristic of *separability*.

In order to formally define separability, we first define some notions related to the topic of *ontological dependence*. In particular, the relations of existential and generic dependence discussed in the sequel are strongly based on those defined in (Husserl, 1970).

Definition 5.10 (existential dependence): Let the predicate ε denote existence. We have that an individual *x* is *existentially dependent* on another individual *y* (symbolized as *ed(x,y)*)iff, as a matter of necessity, y must exist whenever x exists, or formally

(46). ed(x,y) $=_{def} \Box(\varepsilon(x) \rightarrow \varepsilon(y))$

■

From a philosophical standpoint, there is a problem with this definition, namely, that it makes all objects existentially dependent of necessarily existing things (e.g., numbers or platonic forms). Therefore, a better definition would be $ed(x,y) =_{def} \neg\Box\varepsilon(y) \wedge \Box(\varepsilon(x) \rightarrow \varepsilon(y))$. Nonetheless, for the common-sense contingent individuals that we have interest in conceptual modeling, this difficulty can be overseen, since we are mostly interested here in parthood relations between substantial individuals. For this reason, and for the sake of clarity in the presentation, the safeguard condition that rules out necessary existence ($\neg\Box\varepsilon(y)$) is omitted in the subsequent definitions.

With definition 5.10 we can define the concept of an essential part as follows:

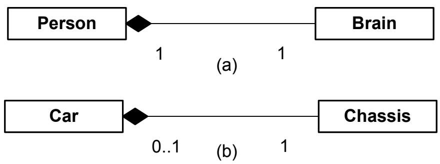
Definition 5.11 (essential part): An individual *x* is an essential part of another individual *y* iff, *y* is existentially dependent on *x* and *x* is, necessarily, a part of *y*: $EP(x,y) =_{def} ed(y,x) \wedge \Box(x \leq y)$. This is equivalent to stating that $EP(x,y) =_{def} \Box(\varepsilon(y) \rightarrow \varepsilon(x)) \wedge \Box(x \leq y)$, which is, in turn, equivalent to $EP(x,y) =_{def} \Box(\varepsilon(y) \rightarrow \varepsilon(x) \wedge (x \leq y))$. We adopt here the *mereological continuism* defended by (Simons, 1987), which states that the part-whole relation should only be considered to hold among existents, i.e., $\forall x,y\ (x \leq y) \rightarrow \varepsilon(x) \wedge \varepsilon(y)$. As a consequence, we can have this definition in its final simplification

(47). $\mathbf{EP(x,y) =_{def} \Box(\varepsilon(y) \rightarrow (x \leq y))}$

■

Figures 5.13.a and 5.13.b below depict examples of essential parts. In figure 5.13.a, every person has a brain as part, and in every world that the person exists, the very same brain exists and is a part of that person. In figure 5.13.b, we have an analogous example: a car has a chassis as an essential part, thus, the part-whole relation between car and chassis holds in every world that the car exists. To put in a different way, if the chassis is removed, the car ceases to exist as such, i.e., it looses its identity.

Figure 5-13 Wholes and their Essential parts



The UML notation used in figure 5.13 highlights a problem that exists in practically all conceptual modeling languages. In order to discuss this problem, let us examine another model represented in figure 5.14.

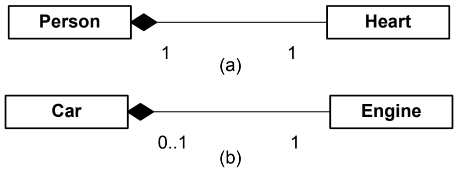


Figure 5-14 Wholes and their Mandatory parts

According to the UML semantics, the models of figure 5.13.a and 5.14.a convey exactly the same kind of information. However, this is not the case, in general, in this domain in reality. Typically, the relation between a person and his brain is not of the same nature as the relation between a person and his heart. Differently from the former, a particular heart is not an essential part of a person, i.e., it is not the case that for every person x there is a heart y, such that in every possible circumstance y is part of x. For instance, the fact that an individual John had the same heart during his entire lifetime was only accidental. With the advent of heart transplants, one can easily imagine a counterfactual in which John had been transplanted a different heart. An analogous argument can be made in the case of figure 5.13.b. Although every car needs an engine, it certainly does not have to be the same engine in every possible world.

The difference in the underlying real-world semantics in the cases of figure 5.13.a and 5.14.a are made explicit if we consider their corresponding formal characterization. In the case of fig.5.13.a, since it is a case of essential parthood, we have that:

(figure 5.13.a) $\Box\, ((\forall \mathbf{Person},x)(\exists! \mathbf{Brain},y)\ \Box(\varepsilon(x) \rightarrow (y < x)))$

Whereas in the case of figure 5.14.a, the corresponding axiomatization is

(figure 5.14.a) $\Box\, ((\forall \mathbf{Person},x)\ \Box(\varepsilon(x) \rightarrow (\exists! \mathbf{Heart},y)(y < x)))$

A similar distinction can be made for the case of figures 5.13.b and 5.14.b:

(figure 5.13.b) $\Box((\forall \mathbf{Car},x)(\exists! \mathbf{Chassis},y)\ \Box(\varepsilon(x) \rightarrow (y < x)))$
(figure 5.14.b) $\Box((\forall \mathbf{Car},x)\ \Box(\varepsilon(x) \rightarrow (\exists! \mathbf{Engine},y)(y < x)))$

In cases such as those depicted in the specifcations of figures 5.13.b and 5.14.b, an individual is not specifically dependent of another individual, but *generically dependent* of any individual that instantiates a given universal. The concept of generic dependence is defined as follows:

Definition 5.12 (generic dependence): An individual *y* is *generic dependent* of a universal U iff, whenever y exists it is necessary that an instance of U exists. This can be formally characterized by the following *formula schema*:

$$\textbf{(48).}\ GD(y,U) =_{def} \Box(\varepsilon(y) \rightarrow \exists U,x\ \varepsilon(x))$$

■

We name individuals such as the instances of Heart and Engine in figures 5.13.b and 5.14.b, respectively, *mandatory parts*.

Definition 5.13 (mandatory part): An individual *x* is a mandatory part of another individual *y* iff, *y* is generically dependent of an universal U that *x* instantiates, and y has, necessarily, as a part an instance of U*:*

$$\textbf{(49).}\ MP(U,y) =_{def} \Box(\varepsilon(y) \rightarrow (\exists U,x)(x < y)).$$

In order to represent the ontological distinction between essential and mandatory parts, we propose an extension to the UML notation used in the examples for the remaining of this chapter. We assume that the minimum cardinality of **1** in the association end corresponding to the part represents a *mandatory part-whole* relation. To represent the case of an *essential part-whole* relation, we propose to extend the current UML aggregation notation by defining the Boolean meta-attribute **essential**.

When the meta-attribute *essential* equals *true* then the minimum cardinality in the association end corresponding to the part must also be **1**. This is expected to be the case, since essential parthood can be seen as a limit case of mandatory parthood. When *essential* equals *false*, the tagged value textual representation can be omitted. This extended notation is exemplified in figure 5.15 below.

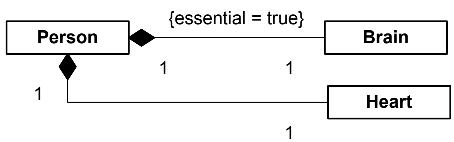


Figure 5-15 Extensions to the UML notation to distinguish between *essential* and *mandatory* parts

We emphasize that the particular examples chosen to illustrate the distinction between *essential* and *mandatory* parts are used here for illustration purposes only. For example, when modeling *brain* as an essential part of persons and *heart* as a mandatory one, we are not advocating that this is a general ontological choice that should be countenanced in all conceptualizations. Conversely, the intention is to make explicit the consequences of this modeling choice, and to advocate for the need of explicitly differentiating between these two modes of parthood. The choice itself, however, is always left to the model designer and is conceptualization-dependent. For example, an ontological choice implicit in the assumption of brain/heart example is that brain transplants are not possible and that the identity of an individual person is determined by the identity of her brain. A discussion on the issue of *personal identity* falls outside the scope of this work, and we are not in position here to discuss the possibility of brain transplants. Again, these ontological choices are taken here merely for the purpose of exemplification. The important point is that there are essential and mandatory parts in the world and one should be able to explicitly and suitably model them.

By using the definition of essential parts, one can formally characterize the notion of those individuals prescribed by Extentional Mereologies (EM), i.e., those individual for which all their parts are essential. These entities, named here *extensional individuals* are formally defined in the sequel.

Definition 5.14 (extensional individual): An individual *y* is named an extensional individual iff for every *x* such that x is a part of y, x is an essential part of y:

$$\textbf{(50). } \mathbf{E(y) =_{def} \Box(\forall x\ (x < y) \rightarrow EP(x,y))}$$

■

Our criticism to EM in section 5.2.2 was that this theory treats all individuals as extensional. This is not to say that there are no such entities. Examples of extensional individuals are discussed in section 5.5.

Up to this moment, we have interpreted possible worlds as maximal state of affairs, which can be factual or counterfactual. In other words, we have assumed a branching structure of time, and each world is taken as a time interval in a (factual or counterfactual) time branch. An alternative is to interpret possible worlds as histories, i.e., as the sum of all state of affairs in a given time branch. In this alternative conception of worlds, we can examine the possible relations between the lifespan of wholes and parts in different types of parthood relations. For instance, figure 5.16 illustrates the possible relations between the lifespan of a whole and one of its essential parts[34].

---

[34]Actually, figure 5.16 depicts the possibilities for the relation between the lifespan of an object x (top) and any object y which is existentially dependent on x.

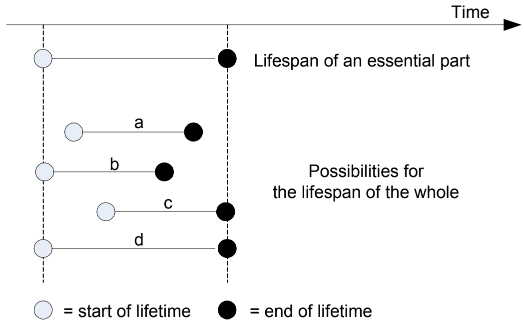


Figure 5-16 Possible relations between the life spans of an individual whole and one of its *essential parts*

This figure illustrates the true possibilities for, for instance, the relation between a chassis and a car as depicted in figure 5.13.b. In this case, the lifetime of the chassis is completely independent from the lifetime of any of the cars it happens to be a part of. Actually, as represented in figure 5.13.b, a chassis does not even have to be connected to a car (whole). This is a case of, what we term, *essential part with optional whole*.

Conversely, if we analyze the relation between a brain and a person, we come to the conclusion that the lifespan (d) in figure 5.16 is the only real possibility in this case. That is to say that the lifespan of a person and her brain should necessarily coincide. This is because, in this case, a brain is also existentially dependent on its host. Whenever we have the situation that a part is existentially dependent on the whole it composes, we name it an *inseparable part.*

Definition 5.15 (inseparable part): An individual *x* is an inseparable part of another individual *y* iff, *x* is existentially dependent on *y*, and *x* is, necessarily, a part of *y:*

(51). IP(x,y) $=_{def} \Box(\varepsilon(x) \rightarrow (x \leq y))$

■

The possible relations between the life spans of an inseparable part and its (essential) whole are depicted in figure 5.17. The case of an essential and inseparable part is shown in figure 5.18.

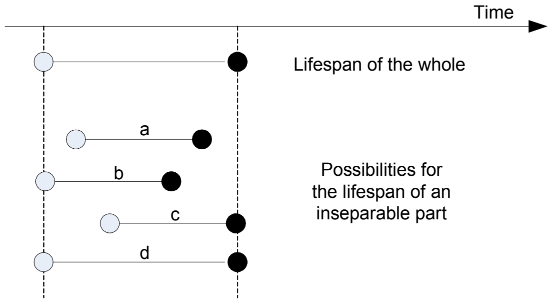


Figure 5-17 Possible relations between the life spans of an individual whole and one of its *inseparable parts*

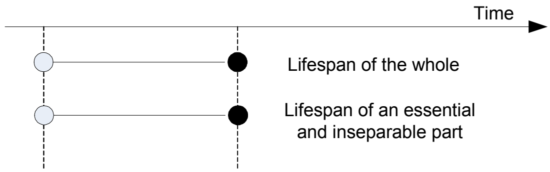


Figure 5-18 Possible relations between the life spans of an individual whole and one of its *essential and inseparable parts*

Figure 5.17 does not represent all the possibilities for, for instance, the relation between a heart and its bearer (figure 5.14.b), since the heart of person is not an inseparable part of a person and, hence, their life spans can be completely independent. A heart can pre-exist its bearer as well as survive its death. Nonetheless, a heart must be part of *a* person, only not necessarily the same person in all possible circumstances. For these cases, of generic dependence from the part to a whole, we use the term *parts with mandatory wholes*.

Definition 5.16 (mandatory whole): An individual *y* is a mandatory whole for another individual *x* iff, *x* is generically dependent on a universal U that *y* instantiates, and *x* is, necessarily, part of an individual instantiating U:

$$\textbf{(52). } MW(U,x) =_{def} \Box(\varepsilon(x) \rightarrow (\exists U,y)(x < y))).$$

■

Once more, the distinction between inseparable parts and parts with mandatory wholes is neglected in practically all conceptual modeling languages. For this reason, we propose to extend the current UML aggregation notation with the Boolean meta-attribute **inseparable** to represent inseparable parts. When *inseparable* is equal to *true*, the minimum cardinality constraint in the association end corresponding to the whole universal must be at least **1**. If *inseparable* is equal to *false*, the tagged value textual representation can be omitted. A UML class representing a whole universal involved in an aggregation relation with minimum cardinality constraint of at least **1** in its association end represents a universal whose instances are mandatory wholes.

5.5 Part-Whole Theories in Linguistics and Cognitive Sciences

In this section, we start by reviewing perhaps the most well known linguistic study on the classification of part-whole relations. In an article entitled *"A taxonomy of part-whole relations"*, (Winston & Chaffin & Herrmann, 1987) (henceforth WCH), propose an account of the notion of part-whole by elaborating on different ways that parts can related to a whole. This study led to a refinement on the formal relation of *partOf* by distinguishing the six types of meronymic relations which are represented in table 5.1 below.

Table 5-1 Different types of Meronymic relations according to (Winston & Chaffin & Herrmann, 1987)

Relation	F	H	S	Examples
Component/ Functional Complex	+	-	+	handle-cup, punch line-joke
Member/Collection	-	-	+	tree-forest, card-deck
Portion/Mass	-	+	+	slice-pie, grain-salt
Stuff/Object	-	-	-	gin-martini, steel-bike
Feature/Activity	+	-	-	paying-shopping, dating-adolescence
Place/Area	-	+	-	Everglades-Florida, oasis-desert

The meta-properties that are used to create these distinctions, namely Functionality (F), Homeomerousity (H) and Separability (S), are explained as follows in a quote from the authors: "Functional parts are restricted by their function, in their spatial or temporal location. For example, the handle of a cup can only be placed in a limited number of positions if it is to function as a handle. Homeomerous parts are of the same kind of thing as their wholes, for example (slice-pie), while non-homeomerous parts are different from their wholes, for example (tree-forest). Separable parts can, in principle, be separated from the whole, for example (handle-cup), while inseparable parts cannot, for example (steel-bike)." (Winston & Chaffin & Herrmann, 1987).

This proposal makes an important contribution in acknowledging that there are different ways that parts composing a whole can relate to each other and to the whole they compose. Nonetheless, the study is overly linguistically motivated, focusing on the linguistic term "part-of" (and its cognates) independently of the ontological and conceptual adequacy of the proposed distinctions.

Firstly, the notion of separability employed by the authors is very different from the one discussed in section 5.4.2 of this chapter. In their case, separability is based on whether "parts can/cannot be physically disconnected, in principle, from the whole to which they are connected." (ibid.). The first problem with this definition is that it applies exclusively to physical objects, since it relies on the notion of "possible physical disconnection" (Gerlst & Pribbenow, 1995). It says nothing, for example, about whether the first act of Turandot is separable or not from the whole play. But more importantly, this notion of separability is of little use for conceptual modelling, since it does not elucidate anything about the ontological dependence relations between parts and wholes, their life-time dependencies, their identity and persistence conditions, etc. If a concept of separability that affords this type of analysis is adopted (such as the one advocated in section 5.4.2), many counterexamples can be found to question the authors' classification. For instance, the relation between a human brain and a human body is of type Component/Functional Complex. However, under the assumptions we have made here, it is not the case that the brain is separable from the body. It certainly can be "physically disconnected", however, due to the existential dependence between brain/body, in the case of separation it will cease to exist as a brain, and, since brain is a kind (in the sense of chapter 4), it will cease to exist as the same object. Therefore, one cannot state that it is the SAME object which has been separated from the body. Conversely, the authors deem the Stuff-Object example of a portion of steel that is part of a bicycle as an instance of an inseparable part. Once more, if an ontologically meaningful notion of separability is adopted, this example is no longer true: if one has a physical-chemical way of separating the steel which constitutes the bike from the bike itself (or the gin from the martini), there is no reason why one should not believe that it is the very same portion of steel (gin) that persists through the separation process. Moreover, the portion of steel might precede the creation as well as survive the destruction of the bike. Additionally, the bike could certainly be made of a different portion of steal. In summary, they are mutually independent from an existential point of view and, hence, separable.

The original taxonomy proposed by WCH has been refined by other authors. For instance, (Gerlst & Pribbenow, 1995) have noticed that the distinction between (a) Component/Functional Complex and (b) Feature/Activity is a superfluous one. What examples such as handle-cup and punch line-joke have in common is a certain role that the part plays w.r.t. the whole, and it is the functionality associated with the role that typically poses constraints on the internal (spatial or temporal) structure of the whole. The same holds for dating-adolescence and paying-shopping. The only difference is that in the former case (a), the relata are endurants whilst in (b) they are perdurants[35].

(Gerlst & Pribbenow, 1995) as well as (Hornsby & Egenhofer, 1998) have noticed that the example of Place/Area are not in any important sense different from the cases of Component/Functional Complex and Feature/Activity aforementioned. Examples such as oasis-desert make clear that the part plays a role w.r.t. to the whole (e.g., providing water) and that, contra WCH, it does not constitute a case of homeomerousity. That is, the parts of desert are not of the same kind as the desert itself (an oasis is not a desert!). Likewise, the Everglades does not instantiate the same universals as Florida, since the latter posses many properties which the former lacks (e.g., having a governor) and vice-versa.

(Gerlst & Pribbenow, 1995) have also observed that the examples of Stuff/Object used by WCH misconceive the phenomena it is supposed to represent. By attempting to differentiate between the linguistic expressions "be partly" (e.g., a martini is partly alcohol) and "made of" (e.g., the lens is made of glass), in the case of Stuff/Object parthood the authors have conflated two propositions: one which is indeed a part-of relation and one which represents a case of *constitution*. For instance, the sentences

(i) A Martini is partly alcohol;
(ii) A bike is partly steel

can be decomposed in two propositions

(iii) A part of Martini is made of alcohol;
(iv) A part of the bike is made of steel

In other words, instead of being primitive, this type of part-whole relation can be analyzed in terms of a constitution[36] relation and one of the other types of part-whole relations (mass/quantity in the case of (iii) and component/Integral Object in the case of (iv)).

In summary, the refinement of the WCH taxonomy proposed in (Gerlst & Pribbenow, 1995) contains the following categories:

(a). Mass/Quantity.
(b). Member/Collection;
(c). Component/Functional Complex;

[35] see discussion of endurants and perdurants on chapter 6.
[36] Constitution is not considered here as an example of a part-whole relation. For instance, it does not satisfy any of the supplementation axioms (Masolo et al., 2003a).

These three new types of part-whole relations are discussed in the following subsections. The Mass/Quantity relation (a) is discussed in subsection 5.5.1; The relations of type (b) and (c) are discussed in subsection 5.5.2.

5.5.1 Quantities

One difference between Mass/Quantity and the other two types of parthood is that the relata of this relation always belong to the category of *amounts of matter* (masses, quantities), while in the Component/Functional Complex and Member/Collection they are *objects*. Quantities (such as water, sand, sugar, martini, wine, etc.) lack both *individuation* and *counting principles*. For this reason, the general terms which are linguistically represented by mass nouns (the linguistic counterpart of amounts of matter) cannot be used to substitute X, Y and Z in sentences such as:

(i) (exactly) five X…;
(ii) the Y which is Z.

A substitution for, for example, water in sentence (i) is not viable, since arbitrarily many parts of water are still water. Likewise, a success in the substitution by water in (ii) depends on the possibility of determining the referent and judge identity statements of individual quantities of water. What exactly should be that referent? Before answering this question we should call attention to what exactly is meant by *homeomerousity* and its relation to the WCH taxonomy.

Traditionally, homeomerousity means that an individual only has parts which are of the same kind (Zimmerman, 1995). This is clearly not the case for all amounts of matter, as the Gin-Martini case demonstrates. However, one can still say that every subquantity of Martini is again Martini and that although Martini is composed of Gin, Gin is itself "homeomerous" in this more liberal sense. This line of reasoning seems to suggest that homeomerousity is equated with infinite decomposability, i.e., for every subquantity of Martini there is always a subquantity of Martini, and the same holds for quantities of Gin. Some authors (e.g., Zimmerman, 1995), nonetheless, admit the existence of quantities of type K having K-atoms, i.e., individuals of type K that have no parts of the same type K. Examples include concrete mass terms such as 'furniture', 'cutlery' or 'crowd'. These allegedly exemplars of quantities are definitely not homeomerous, not even in the more liberal sense. For example, there are parts of a crowd, namely individual persons, which is not are a crowd themselves and which are not homeomerous in any meaningful sense[37]. What can be said in this case is that these aggregates have a uniform structure and, in parity with (Gerlst & Pribbenow, 1995), we consider them as examples of *member/collection* parthood instead.

One could also consider homeomerousity to simply mean that an aggregate can merely have some parts of the same kind while having other parts of other kinds. However, if this were to be the case one could not use it as a meta-property to differentiate Mass/Quantity from Member/Collection. Notice that examples of "homeomerous" parts in this sense can be easily found for Member/Collection: a crowd can be part of a larger crowd; a forest can be part of larger forest.

Since WCH do not consider member/collection to be homeomerous, they would have to agree that quantities should be considered necessarily infinitely divisible in quantities of the same kind. We thus take the liberal approach and consider the case of real homeomerosity as a special case of infinitely divisible pure substances.

Now, an important question that comes to the mind is how we should represent in conceptual models the universals whose instances are quantities in the sense just mentioned? As discussed in chapter 4, individuals in a conceptual model must have a determinate identity, a requirement which is put forward since the initial conceptions of the first conceptual modeling languages. For instance, (Borgida, 1990) writes that, in both semantic data models and knowledge representations, *"an individual is assumed to have a unique, intrinsic identity"*. As we have discussed in chapter 4, in order to be able to make viable references to general terms which are not count nouns (mass terms, adjectives, verbs) they first must be nominalized. A nominalization of a mass noun, verb or an adjective promotes the shift to the category of count nouns (e.g., the fall of Jack, a lump of clay), hence, allowing for the representation of the corresponding sortal universals. An important question that then arises is: what is the best nominalization of mass terms so that they can be satisfactorily represented in conceptual models?

In order to investigate the possibilities let us take the example of a portion of wine which could be differentiated from other portions of wine by the year and source vineyard. What is the meaning (and implied principle of individuation) of a universal whose instances are portions of wine?

A first possibility is to consider the referent of the expression "the portion of wine" as a mereological fusion of all subportions of wine that

---

[37] Of course, one could say that a person is a physical object and that every part of a physical object is itself a physical object. However, if interpreted in this way, homeomerousity becomes completely non-informative since everything can be said to be homeomerous.

constitutes it. This approach is standard in philosophy and (Simons, 1987) suggests that quantities are probably the best case of application of the Classical Extentional Mereologies (CEM), since practically all objections raised against the CEM for the purpose of conceptual modeling can be safely lifted in the case of quantities and their parts. For instance, portions are always transitive and there is always a sum of two portions of quantity regardless how scattered they are. Nonetheless, and still from a philosophical point of view, the first problem with this conception of quantities is whether it is at all possible to have a principle of identity for portions of wine in this sense (Zimmerman, 1995; Lowe, 2001). A mereological principle of identity in this case prescribes that portion of wine A is equal to portion of wine B iff they have the same parts. However, since the parts of A and B are also portions of wine, to decide if A and B have the SAME parts one has to decide about the identity of the parts, and the parts of the parts, leading to an infinite regress, since, by assumption, quantities are infinitely divisible. One could derive some synchronic information about identity by saying that two quantities are different if they do not occupy the same region of space[38]. However, this cannot be used as a diachronic principle of identity. Alternatively, one could say that a quantity A in t_1 is not the same quantity as B in t_2 if they have different properties such as volume or weight (still assuming the mereological principle). Nonetheless, the fact that B in t_2 has the same volume or weight as A in t_1 can only account for the sameness of the quantities in a very loose sense, meaning the same measure. In other words, in this case, the relation between A and B is one of equivalence, not one of *numerical identity*.

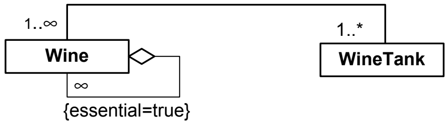
In conceptual modeling, there are a number of situations in which dealing only with qualitative identity of masses does not suffice. For instance, one might be interested in tracking the persistence of a quantity of a certain liquid which has been poisoned, or, in a chemical experiment, it could be important to track the change of properties in the very SAME persisting quantity. For this reason, contra (Gerlst & Pribbenow, 1995), who proposes that "quantities are arbitrary pieces of the whole as long as they are properly characterized by the quantitative measure", we advocate that a treatment of masses in conceptual modeling must deal explicitly with the case of numerical identity.

However, there is still a bigger problem with this idea from a conceptual modeling stance. Figure 5.19 depicts the representation of a portion of Wine universal in the sense just mentioned. In this specification, the idea is to represent a certain portion of wine as the mereological sum of all subportions of wine belonging to a certain vintage. As it can be noticed,

[38] There is still the issue that space itself could be considered to be infinitely divisible (Lowe, 2001).

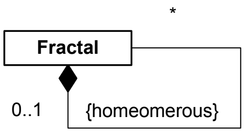
since every portion of wine is composed of subportions of wine, the cardinality of the part-whole relation cannot be specified in a finite manner. The same holds for every cardinality constraint for associations involving portions of wine.

Figure 5-19 Problems with the representation of Quantities as Mereological Sums



Furthermore, "homeomerous" entities represented in this manner can induce to representation errors in the presence of other shareability constraints. For example, figure 5.20 presents an exact copy of a UML class diagram from (Saksena et. al, 1998)[39] that symbolizes a Fractal (perhaps the prototypical example of homeomerous form). The intention of the authors seems to be to represent that a fractal, i.e., the rendering of one iteration step of an IFS, is part of only one instance of the infinite recursion of this function (Peitgen & Jurgens & Saupe, 1992). In other words, a part of an instantiation of a *Julia Series* is not a part of another instantiation of the same fractal form, or part of an instantiation of the *Mandelbrot Series*. However, that is not what is represented in the model. The model states that every instance of a fractal (i.e., every iteration of an IFS) is part of only one other fractal. This is clearly mistaken: the nth iteration of an IFS is part of all previous iterations of the same fractal. This problem is far from being specific to Fractals. In fact, for all homeomerous entities, with the exception of the maximal sum of subquantities, all other parts of quantity are necessarily part of innumerable other quantities of the same kind.

Figure 5-20 Mistaken representation of homeomerous part with non-shareability (from Saksena et al., 1998)



There is still one further philosophical argument invalidating this modeling alternative for the purpose of conceptual modeling. According to (Wandinger, 1998), masses are *identificationally dependent* of substantials that are instances of sortal universals. As he put it: "The formal concepts 'amount', 'part' and 'stuff', like 'mass' and 'matter', are formed 'on the back of' (Simons 1987, p. 191) the formal concept '(material) thing' or '(material) object'. There is no mass, except the mass of a certain object.

[39] The tagged value {homeomerous} in the picture is a proposal of (Saksena et al., 1998) and it is present in the original article.

There is no stuff except the stuff a certain thing consists of". This is to say that individuating a quantity depends on the definite descriptions used for referring, which succeeds only when the referent can be individuated by sortals, i.e., it is the water in the bathtub, the clay that constitutes the statues, a cube of sugar, that can be referred to, not just some water, some clay, or some sugar. Moreover, quantities have no criteria for when they constitute a whole of some sort (unity criteria), except in cases in which we derive those criteria from objects that are only identifiable via sortals. This view is also supported in (Quine, 1960), who proposes that every occurrence of a quantity expression having the form "The K", "The same K" (where K is a concrete quantity universal) is really a masked reference to a *portion* of K to which an ordinary sortal universal applies. Alternatively formulating, as Quine puts it: "[in these situations, always] some special individuation standard is understood from the circumstances". This perspective gives rise to a second option of representation for the wine/wine tank example, as depicted in figure 5.21.

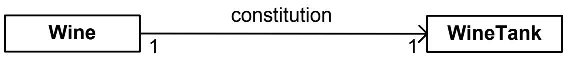


Figure 5-21 Representation of Quantities as *Identificationally Dependent* of *Objects*

There are a number of observations that can be made about figure 5.21. In this second option for nominalization of quantities, Wine means the maximal content of a Wine Tank. Likewise, the referent of a portion of clay means whatever quantity of clay constitutes a given statue, which is in turn individuated by the principle of identity supplied by the sortal statue. In this representation, there is no longer a problem for the specification of cardinality constraints between portion of wine and wine tank: every wine tank has as its content one single definite portion of wine. Additionally, since wine portion means the maximal content of a wine tank, it is not the case that this concept is homeomerous, i.e., there is no part of a portion of wine which is itself a portion of wine (otherwise, it would not be the maximal content). Thus, there is no problem with infinite cardinalities, infinite divisibility and infinite domain of individuals (see section 5.1). Moreover, portion of Wine becomes a genuine sortal: it is always determinate if two portions of wine are identical and it is always determinate how many portions of wine there are. We emphasize that it is still convenient to consider portions of wine as having all parts as essential, i.e., if a part from a portion of wine is removed then it becomes a different portion of wine. The reason for this is that, otherwise, in some cases, the portion of wine becomes identical with the object it constitutes. Take the statue/lump of clay example. If A is the same lump of clay as long as it constitutes the same statue B, A would have necessarily the same properties as B and have a complete life-time dependency. For instance: (a) if a piece of B is removed, B is still the same statue and so is A still the same lump of clay, since it still constitutes the same statue; (b) If the form of B is altered, B ceases to exist and so does A, since it no longer constitutes the same statue.

In summary, in this second alternative nominalization, a quantity is an *inseparable constituent* of the object it constitutes. In a manner which is analogous to the way such objects named *Features* in DOLCE (Masolo et. al, 2003a) (e.g., holes, bumps, edges, stains) are considered here *inseparable parts* of their hosts (see section 5.4.2).

Although this second alternative contains important advantages over the first one from a conceptual modeling point of view, it leads to a problematic consequence. The problem is implied exactly by the *rigid specific dependence* relation between a quantity in this sense and its container. As put by (Cartwright, 1965), a sentence such as the "same K" (where K is a quantity universal) should be understood in a such way that

x is the same K as y iff x is some K, y is some K, and (x = y)

or, as discussed in chapter 4, in statements of identity, the relata must instantiate the same kind, i.e., the same rigid sortal supplying their principle of identity. In parity with (Cartwright, 1965; Zimmerman, 1995), we consider as meaningful a sentence such as "the sugar that was in that cube is the same sugar as the one in this lump". However, if this is the case, which kind of individuation and identity principle should be applied to x and y, that of cubes or of lumps? Notice that x (or y) is not necessarily none of the above (in the modal sense) and there is not an ultimate substance sortal that would cover both universals.

A third nominalization alternative that solves this problem is presented as follows. This last option relies on the notion of *piece* discussed in (Lowe, 2001). According to Lowe, a piece of a quantity K is a maximally self-connected object constituted by portions of K (portions in the first sense discussed above). Like in the second nominalization alternative, a quantity of K in this sense is an instance of a genuine sortal universal, i.e., it has definite individuation, identity and counting principles. Moreover, it is not homeomerous, however it can still be composed of other quantities K' in the same sense of quantity (see figure 5.22). All its parts are also essential, and it does not contain the infinite regress problems mentioned for the first case. Nonetheless, differently from the second alternative, the dependence relation between a quantity and its container is a generic not a specific one. For this reason we can state that for the same maximally self-connected quantity of wine, there can be several "container phases". This idea is represented in the (incomplete) model of figure 5.22. A vintage is an object constituted by (possibly many) quantities of wine, it is, however, not a quantity since it can be scattered over many quantities. Moreover, it is not necessary for its constituent quantities to be essential: even if the quantity of wine now stored in a certain tank is destroyed, we still have numerically the same vintage. We, therefore, propose the use of this third alternative for the nominalization of quantities and their representation in conceptual models. From now on, we shall use the term *quantity of matter K* or *objectified portion of matter K* to refer to a *piece of K* in the Lowe's sense aforementioned, and use the stereotype «quantity» to symbolize a kind whose instances are quantities in this sense.

We can summarize many points of the argument carried out in this section by using an example proposed by (Cartwright, 1965). If a sentence such as "Heraclitus bathed in some water yesterday and bathed in the same water today" is true then for some suitable substituends of x and y we have that: (a) x is a quantity of water and Heraclitus bathed in x yesterday; (b) y is a quantity of water and Heraclitus bathed in y today; (c) x = y. However, (c) when interpreted as (d) "the water Heraclitus bathed in yesterday = the water Heraclitus bathed in today" requires that: (e) there is exactly one x such that x is a quantity of water and Heraclitus bathed in x both yesterday and today. But, if quantity is interpreted in the first sense above, there are not one but infinitely many particulars that would satisfy (a), (b) and (d) without satisfying (c). The same does not hold for the second and third senses.

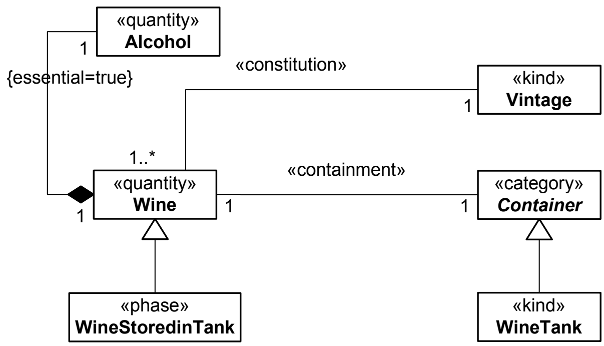


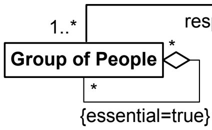
Figure 5-22 Representation of quantities as *maximally-self-connected portions* which are *generically dependent* of the continuants they constitute

5.5.2 Collections and Functional Complexes

As recognized by WCH, collections such as tree-forest, card-deck, brick-pile, lion-pack cannot be said to be homeomerous wholes. Collections have parts that are not of the same kind (e.g., a tree is not forest) and they are not infinitely divisible. As a consequence, a representation of a collection as a mereological sum does not lead to the same complications as those encountered in the first alternative for the representation of quantities.

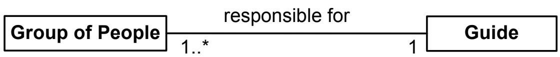
Take, for instance, the example depicted in figure 5.23, which represents a situation analogous to that of figure 5.19. Differently from the case with quantities in figure 5.19, there is no longer the danger of an infinite regress or the impossibility for specifying finite cardinality constraints.

Figure 5-23
Representations of Collections with parts of the same type



In figure 5.23, the usual maximum cardinality of "many" can be used to express that group of people has as parts possibly many other groups of people and that a guide is responsible for possibly many groups of people. Nonetheless, in many examples (such as this one), this model implies a somewhat counterintuitive reading. In general, the intended idea is to express that, for instance, John as a guide, is responsible for the group formed by {Paul, Marc, Lisa} and for the other group formed by {Richard, Tom}. The intention is not to express that John is responsible for the groups {Paul, Marc, Lisa}, {Paul, Marc}, {Marc, Lisa}, {Paul, Lisa}, and {Richard, Tom}. A simple solution to this problem is to consider groups of people as maximal sums, i.e., groups that are not parts of any other groups. In this case, depicted in figure 5.24, the cardinality constraints acquire a different meaning and it is no longer possible to say that a group of people is composed of other groups of people.

Figure 5-24
Representation of Collections as Maximal Sums

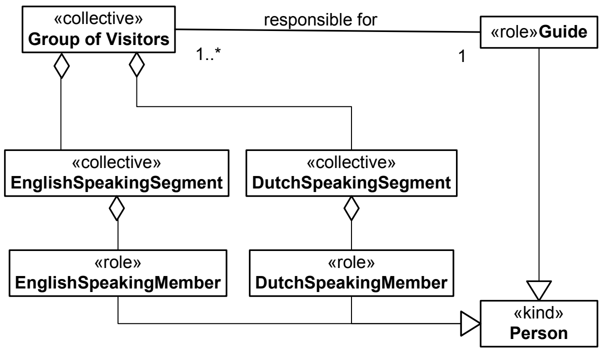


This solution is similar to taking the meaning of a quantity K to be that of a maximally-self-connected-portion of K. However, in the case of collections, topological connection cannot be used as a *unifying relation* (see section 5.3) to form an integral whole, since collections can easily be spatially scattered. Nonetheless, another type of connection (e.g., social) should always be found. A question begging issue at this point is: why does it seem to be conceptually relevant to find *connection relations* leading to (maximal) collections? We defend the idea that, alike quantities, arbitrary sums of collections make little cognitive sense. In the case of quantities, we are not interested in the sum of a subportion of my bathtub's water with a subportion of the North Sea. Instead, we are interested in the quantity of water in that specific bottle, or lying in that specific location. Likewise, we are not interested in arbitrary collections of individuals but aggregations that have a purpose for some cognitive task. As discussed in section 5.3, integral wholes exist because there is some unifying relation that holds among the totality of its parts and only among them, i.e., a closure system. For example, a group of people of interest can be composed by all those people that are attending a certain museum exhibition at a certain time. Now, by definition, a closure system is maximal (see definition 5.4), thus, there can be no group of people in this same sense that is part of another group of people (i.e., another integral whole unified by the same relation).

Nonetheless, it can be the case that, among the parts of a group of people, further structure is obtained by the presence of other collections unified by different relations. For example, it can be the case that among the parts of a group of people A, there are collections B and C composed of the English and Dutch speakers, respectively. This situation is depicted in figure 5.25. Neither the English speaking segment nor the Dutch speaking segment is a *group of people* in the technical sense just defined, since a group of people has properties that apply to none of them (e.g., the property of having both English and Dutch segments). Moreover, the unifying relations of B and C are both specializations of A's unifying relation. For example, A is the collection of all parties attending an exhibition and the B is the collection of all English speakers among the parties attending the same exhibition. For this reason, transitivity holds unrestricted among the part-whole relations depicted in figure 5.25.

We shall use the stereoptype «collective» to symbolize a kind whose instances are collections.

Figure 5-25
Representation of Collections



Judging from this perspective, collections bear a strong similarity to functional complexes in the classifications of WCH and Gerlst & Pribbenow. This similarity is in fact acknowledged by (Gerlst & Pribbenow, 1995) in the following passage: "Depending on what aspect of this structure [i.e., the thing's structure] one focuses on, different types of part-whole relations are possible. However, in many cases there is a primary or prototypical view…the primary view critically depends on the granularity level which is assumed for classifying an entity…configurations such as 'a pile of books' are on one hand similar to collections and on the other hand similar to complexes". In the course of the article, the authors propose that the difference between a collection and a functional complex is that whilst the former has a *uniform* structure, the latter has a *heterogeneous* and *complex* one.

We propose to rephrase this statement in other terms. In a collection, all member parts play the same role type. For example, all trees in a forest can be said to play the role of a forest member. However, a tree is not necessarily a forest-part, i.e., the latter is an anti-rigid concept, and representing the part-whole relation between forests and trees via the role type *forest-tree* prevents one from having to specify minimum cardinality relations which are zero. In complexes, conversely, a variety of roles can be played by different components. For example, if all ships of a fleet are conceptualized as playing solely the role of "member of a fleet" then it can be said to be a collection. Contrariwise, if this role is further specialized in "leading ship", "defense ship", "storage ship" and so forth, the fleet must be conceived as a functional complex. In summary, collections as *integral wholes* (i.e., in a sense that appeals to cognition and common sense conceptual tasks) can be seen as limit cases of Gerlst & Pribbenow's *functional complex*, in which parts play one single role forming a uniform structure.

Finally, we emphasize that, differently from *quantities*, collectives do not necessarily obey an extensional principle of identity. Some collectives can be considered extensional by certain conceptualizations. In this case, the addition or subtraction of a member renders a different collective. However, we also consider here the existence of *intentional collectives* obeying non-extensional principles of identity (Botazzi et al., 2004).

5.6 The Problem of Transitivity Revisited

In section 5.3, we have addressed the problem of fallacious transitivity inferences in part-whole relations by employing the notion of contexts. However, at that point, we are still dealing with one single type of parthood inherited from the formal part-whole relation in mereology. A question begging issue at this point is: by taking advantage of the refinement of this relation developed in section 5.5, can we provide more direct guidelines for prescribing or proscribing transitive relations? In other words, can we establish some relations between the specific type of part-whole relation and the issue of transitivity?

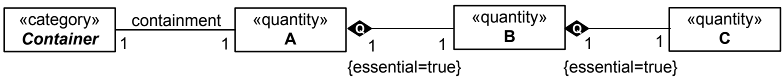
The distinction between the three types of part-whole relation discussed in the previous section (quantity/mass, member-collection, component-functional complex), actually reflects a distinction among different types of relata.

Mass-Quantity:

The quantity/mass relation holds between *quantities* (in the technical sense explained in section 5.5.1). Let us suppose a model such as the one depicted in figure 5.26, in which A, B and C are quantities. We can show that for any A, B, C, the part-whole relation (C < A) holds as a result from the transitivity (C < B) and (B < A). The argumentation can be developed as follows: if A is a quantity then it is a maximal portion of matter unified by the characteristic relation of self-connectedness. That is, any part of A is connected to any other part of A. If B is part of A then B is connected to all parts of A. Likewise, if C is part of B then C is connected to all parts of B. Since connection is transitive, then we have that C is connected to all parts of A. Thus, since A is unified under self-connection, C must be part of A (otherwise the composition of A would not be a closure system, see definition 5.4). Therefore, we conclude that for the case of quantities, transitivity always holds.

Another way to examine this situation is by inspecting A at an arbitrary time instant t. We can say that all parts of A are the quantities that are contained in a certain region of space R (i.e. a topoid, see Guizzardi & Herre & Wagner, 2002a). Since A is an objectified matter, than the topoid R occupied by A must be self-connected. Therefore, if B is part of A then B must occupy a sub-region R', which is part of R. Likewise, if C is part of B, it occupies a region R'', part of R'. Since spatial part-whole relations are always transitive (Johansson, 2004), we have that R'' is part of R, and if C occupies R'', then it is contained in R. Ergo, by definition, C is a part of A.

Figure 5-26 Part-Whole relations among quantities



Parthood relationships between quantities are always non-shareable. For instance, in figure 5.23, B can only be part of one single quantity of A, since A is a maximal. Moreover, A has at maximum one quantity of B as part, since B is also a maximal portion. Finally, as discussed in section 5.5.1, every part B of A is essential.

As in figure 5.26, we decorate the standard UML symbol for composition with a Q to represent a quantity/mass parthood relation. If cardinality constraints are fully specified, then the Q-parthood is a relation which:

(i) is non-shareable;
(ii) the part is essential to the whole;
(iii) the cardinality constraints in both association ends are of one and exactly one;
(iv) only holds between quantities;
(v) it is transitive, i.e., for all a,b,c, if Q(a,b) and Q(b,c) then Q(a,c).

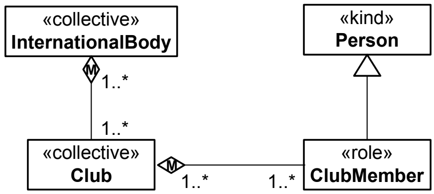
The second type of parthood considered here is the *member-collection* relation. Notice that the relation between EnglishSpeakingSegment and Group of People in figure 5.25 does not exemplify this type of relation. Instead, it is a relation of type *subcollection-collection*. The difference between the two types of parthood can be linguistically motivated in the following manner. According to (Vieu & Aurnague, 2005), classical semantic analysis of plurals and groups (Landman, 1996; Link, 1991) distinguish between atomic entities, which can be singular or collectives, and plural entities. Any collective atomic entity (e.g., a group of men, the herd) is constituted of some plural entity (e.g., {John, Paul, George, Ringo}, or the cows, which are sum of man-atoms, or cow-atoms, respectively). Following (Aurnague & Vieu, 2005), we have that:

Member-Collection:

The *member-collection* relation is one that holds between a *singular entity* and either a *plural or a collective term*. This is the relation, for example, between John and a group of men, the cow Joanne and the herd, or between the province of Overijssel and The Netherlands. Member collection relations are never transitive, i.e., they are intransitive. This can be understood in our analysis in the following way. To say that a member must be a *singular* entity coincides in this case with this entity being an *atom of a given context*. Or to put it differently, the unifying relation underlying membership cannot be further refined. An example of a member-collection relation is the following case from (Henderson-Sellers & Barbier, 1999): "I am member of a club (collection) and my club is a member of an International body (collection). However, it does not follow that I am a member of this International body since this only has clubs as members, not individuals". This situation is depicted in figure 5.27 below, in which we decorate the standard UML symbol for aggregation with an M to represent a member/collection parthood relation. A M-parthood relation is intransitive, i.e., for all a,b,c, if M(a,b) and M(b,c) then ¬M(a,c). Therefore, in this model, although the ClubMember John is part of the F.C.Twente Club, and that the F.C.Twente is part of the an international body Ψ, John is not part of (member of) Ψ.

The statement above about the intransitivity of the M-parthood can be made more general. Members of a collection are considered to be atomic w.r.t. the context in which the collection is defined. As a consequence, if an individual *x* is a part of (member of) a collection y, then for every z which is part of (memberOf, componentOf, subCollectionOf) x, z is not a part of (member of) y. Thus, for instance, although an individual John can be part of (member of) a Club, none of John's parts (e.g., his heart) is part of (member of) that Club.

Figure 5-27 Examples of member/collection part-whole relations



Subcollection/Collection:

The *subcollection-collection*, conversely, is a relation that holds between two plural entities, or collectives constituted by such plural entities, such that all atoms of the first are also atoms of the second. In our analysis, subcollections are always created by refining the unifying relation of a certain collection. In the example of figure 5.25, the *EnglishSpeakingSegment* is part of the *Group of Visitors*. The latter is unified by, for instance, taking all people that visit an exhibition x at date y. The former by taking all people that visit an exhibition x at date y, and that speak English. As another example, take a forest and the north part of the forest. The latter is a part of (subcollection-of) the former. The forest is unified by taking all the trees located in a given area A. The north-part of the forest is unified by the taking all the trees located in the north part of A.

As in figure 5.26, we decorate the standard UML symbol for composition with a C to represent the subcollection/collection parthood relation. If cardinality constraints are fully specified, then the C-parthood is such that:

(i) the part is essential to the whole;
(ii) the cardinality constraints in the association end relative to the part is one and exactly one;
(iii) only holds between collectives;
(iv) it is transitive, i.e., for all a,b,c, if C(a,b) and C(b,c) then C(a,c). In this figure this means that every C is also part of an A.

Notice that if we have M(x,y) and C(y,z) then it is also the case that M(x,z), since this would clearly be a case of (x $<_A$ y) and (y $<_B$ z), where if R' and R are the unifying relations of A and B then R' is a refinement of R. Therefore, in figure 5.28 we have that every D is a part of (member of) a B and also a member of an A. In figure 5.25, this means that every *EnglishSpeakingMember* is part of (member of) the *Group of Visitors*.

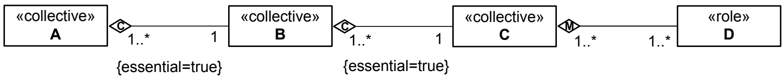


Figure 5-28 Examples of *SubCollection/ Collection* and *Member/Collection* part-whole relations

Component-Functional Complex:

Now, we can turn our attention to the most important type of parthood relation for the purposes of conceptual modeling, namely, the component/functional complex relation.

Differently from collectives, complexes are composed by parts that play a multitude of roles in the context of the whole. As a consequence, in practical cases, the isolation of contexts for complexes can result in a complicated task, and in particular, one that requires a strong knowledge of the domain being modeled. We believe it is reasonable to assume that conceptual modelers working on the representations of specific domains have access to sufficient domain knowledge that enables them to define unifying relations for complexes, and to consequently isolate suitable contexts. Nonetheless, as we show in the remaining of this section and also in section 7.4, we can provide some patterns of relations for the case of component/functional complex parthood, in which transitivity is guaranteed to hold. The proposal of these patterns is very much in line with one of the main objective of this thesis, namely, to provide well-founded engineering tools for conceptual modelers.

The parts of a complex have in common that they all posses a functional link with the complex. In other words, they all contribute to the functionality (or the behavior) of the complex. Therefore, if it is generally the case that essential parthood entails dependence (see definition 5.11), in this type of parthood relation, an essential part represents a case of *functional dependence*. To put it more precisely, for all complexes, if x is an essential part of y then y is functionally dependent on x.

Let us take the example depicted in figure 5.29[40]. According to definition 5.11, we have that:

[40] Since the component/complex parthood relation is the most common one in conceptual modeling, we use the standard presentation of the UML aggregation symbol to represent relations of this type. In the examples concerning these relations in the remaining of this section, for simplification, we shall refrain from specifying cardinality constraints.

(i) $\Box(\forall \text{Person},x)(\exists!\text{Brain},y)\ \Box(\varepsilon(x) \rightarrow (y < x))$;

(ii) $\Box(\forall \text{Brain},x)(\exists!\text{Cerebellum},y)\ \Box(\varepsilon(x) \rightarrow (y < x))$.

From (i) and (ii) it follows that

(iii) $\Box((\forall \text{Person},x)(\exists!\text{Cerebellum},y)\ \Box(\varepsilon(x) \rightarrow \varepsilon(y)))$.

Hence, following definition 5.10, we have that

(iv) $\Box((\forall \text{Person},x)(\exists!\text{Cerebellum},y)\ ed(x,y))$,

i.e., every person is dependent on a specific cerebellum.

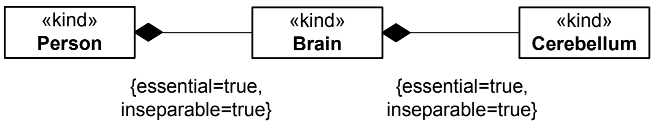


Figure 5-29 Examples of (functional) essential parthood between complexes

In chapter 6, we use the notion of existential dependence (definition 5.10), to distinguish between what we name *substantial* and *moments*. In short, substantials are concrete objects of every day experience such as cats, persons, houses, cars, brains and cerebellums. Moments, in contrast, are particulars such as a weight, a though, an electric charge, a smile, a hand shake. Substantials are mutually independent, except in case they are related via parthood, i.e., in the case of essential or inseparable parts. That is to say that if a substantial x is external to a substantial y (i.e., disjoint from y) then y is independent of x (see formula 10 in chapter 6).

Now, since the elements in figure 5.29 are substantials, and since we have shown that every person is dependent of a specific cerebellum, if the cerebellum x is not part of the person y, we have to conclude that y is existentially dependent on a substantial that is disjoint from it, and consequently, that y (a particular person) is not a substantial. This result is certainly absurd. Therefore, we must conclude that transitivity always holds across essential parthood relations.

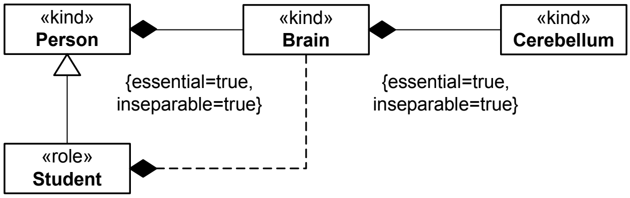
Figure 5.30 is an extension of the model of figure 5.29 by adding the role universal Student. We can show that, in this case, transitivity holds between the relationships cerebellum/brain and the inherited brain/student (represented by a dashed line) in the figure. Since the semantics we assume for the subsumption relation is that $\Box((\forall \text{Student},x)\ person(x))$ (see section 4.1), from this and from (i) and (ii) above we can show that

(v) $\Box((\forall \text{Student},x)(\exists!\text{Cerebellum},y)\ \Box(\varepsilon(x) \rightarrow \varepsilon(y)))$, or equivalently, that

(vi) $\square((\forall\ \textbf{Student},x)(\exists!\textbf{Cerebellum},y)\ \textbf{ed}(x,y))$.

Now, since a student is also a substantial then we must conclude that every student has a cerebellum as an essential part.

Figure 5-30 Transitivity of parts in the case of Roles



In section 7.4, we resume this discussion to consider also the case of *generic (functional) dependence* between complexes, i.e., in which situations transitivity holds across relationships of *mandatory* (as opposed to *essential*) *parthood* between functional complexes.

5.7 Related Work

In this section, we discuss several attempts in the literature to provide a theoretical foundation for part-whole relations in conceptual and object-oriented modeling.

5.7.1 The Approach of Odell and Bock

In an article entitled *"Six Different Kinds of Composition"* (Odell, 1998), James Odell has proposed an adaptation of the taxonomy of part-whole relations proposed by Winston, Chaffin and Herrman (see section 5.5) for the purpose of modeling object-oriented systems. We recognize the pioneering nature of Odell's work as one the first attempts in the object modeling literature to emphasize the multitude of part-whole relations that should be considered. However, there are a number of important issues in which our approach differs from his, some of which are related to features that Odell's approach inherits from WCH.

On page 3 of the *"Six Different Kinds of Composition"*, the author makes the following comment on, what he terms, *material-object* composition (stuff-object in WCH's terminology): "The word *partly* is not a requirement of the material-object relationship. For instance, a windshield could be made entirely of glass-not just partly. Other material-object relationships require a subjective judgment. For example, can the ceramics (of the spark plugs) be removed from a car? If so, ceramics is a component-integral object relationship, instead". From our perspective, in this passage there is confusing between the relations of parthood and constitution. If a glass would be considered the only part of a window, it would not satisfy the weak supplementation axiom, which is considered a minimum requirement for all part-whole relations (Simons, 1987). However, the biggest problem is with the second part of the statement: the difference between a spark plug and the ceramics it is made of is not a subjective one, but an ontological one. Akin to the statue/lump of clay problem, the plug and the quantity of ceramics in its *constitution* have different modal properties and, by Leibniz law, cannot be deemed identical. Finally, if we were to respect our common sense intuitions, both the ceramics and the spark plugs should be considered separable parts of the car. In other words, if separability is taken in an ontologically meaningful modal sense (see section 5.5): the ceramics (or the plugs) pre-existed the car, i.e., there are worlds in which the ceramics (or the plugs) exist without being part of the car.

This conception of separability (as being physically entangled) has its source in the original WCH proposal. However, they reappear in other parts of Odell's article. For instance, on page 4 of the same paper, he proposes that the difference between a place-area and portion-object compositions (portion-mass in WCH's terminology) is that while both are homeomerous, only the former is constituted solely by inseparable parts. As discussed in section 5.5, place-area compositions cannot be deemed homeomerous in any meaningful sense of the word, but, other than that, it is not the case that their parts are necessarily inseparable either. For instance, the province of Trentino-Alto Adige is a (place-area) part of Italy, but not an inseparable part, since there are worlds (namely before 1921), in which it belonged to the Austrian-Hungarian Empire. This is far from being an isolated case: Uruguay used to be part of Brazil (under the name of Cisplatina), Strasbourg used to be part of Germany, and so forth. In summary, there is nothing in the place-area composition relation that requires the parts to be inseparable.

One of the main intentions of the typology of meronymic relations proposed by Winston, Chaffin and Herrmann (WCH) is to address the problem of fallacious cases of transitivity in part-whole relations. By adopting the WCH framework, Odell states that "when the same kind of [part-whole] relationship is used, the conclusion is always correct". A similar statement is also found in (Artale et al., 1996): "as long as we are careful to keep a single sense of part, it seems that the part-whole relation is always transitive. However, when we inadvertently mix different meronymic relations problems with transitivity arise". As we discussed in section 5.6, *member-collection* relations are intransitive and *component-functional complex* relations are non-transitive (i.e., transitive in certain circumstances and not transitive in others) irrespective of whether we maintain the same "sense of part".

As we have also discussed in section 5.5.1, if homeomerousity is given its standard interpretation in philosophy, portion-mass compositions are not homeomerous either. This actually becomes clear in an alleged example of portion-mass composition mentioned by Odell (p.3, ibid.): *"A meter is part of a kilometer"*. Instances of Meter and Kilometer clearly do not have the same properties (e.g., only the latter has the property of corresponding to 1000 meters). Although both are subtypes of Measure, this should not be an argument for considering it a case of homeomerousity. After all, any part of a physical object can be considered to be a physical object.

In differentiating the member-bunch composition (WCH's member-collection) from the component-functional complex, Odell writes (page 4, ibid.): "In the composition relationships above, the parts bear a particular functional or structural relationship to one another or to the object they comprise. Member-bunch composition has no such requirement…The member-bunch relationship is different-based, instead, on spatial proximity or social connection. For a shrub to be part of a garden implies a location close to the other plants. For an employee to be part of a union implies a social connection". As we have discussed in section 5.3, in order to provide a suitable foundation for conceptual part-whole relations, a mereology (a theory of parts) should be complemented with a theory of wholes. This is because the aggregates that interest us in a commonsense ontology of reality are integral entities, unified by genuine relations, not arbitrary formal ones. In fact, in our theory, an integral whole is defined in terms of closure systems containing all and only those objects that bear a unifying (material) relation to one another. Now, if this is the case, this should hold for all types of conceptual aggregates, i.e., both for forests and televisions. Otherwise, if trees in a forest bear no genuine relation to each other, how can we decide which trees are part of the forest and which are not? The point we want to emphasize here is that the relationships unifying collectives are genuine unifying relations, and that deciding whether they are "functional or structural" may be a matter of conceptualization. After all, for instance, employees who are *socially connected* to each other can very well be functionally related to each other.

Finally, in section 5.5.2, we have articulated a distinction between member-bunch and component-functional complex based on the fact that only in the former all parts seem to play the same role w.r.t. to the whole. In the examples of *member-partnership* (a special case of member-bunch) provided by Odell (e.g. "Ginger and Fred are a waltz couple", p.5), the members can only be said to play *the same role* if one is willing to raise to a certain extent the level of abstraction. This seems to support our argument that perhaps the distinction between member-collection and component-integral object is strongly dependent on the conceptualization at hand.

5.7.2 The BWW Approach

In (Opdahl & Henderson-Sellers & Barbier, 2001), the authors employ the BWW (Bunge-Wand-Weber) ontology (Wand & Weber, 1993; Weber, 1997) as a foundation for a conceptual framework defining a taxonomy of part-whole relations in terms of their *primary* (e.g., reflexivity, asymmetry and transitivity), *secondary* (e.g., shareability, mutability, separability) and *consequent properties* (e.g., ownership, propagation of operations, encapsulation). Moreover, the article analyses the different kinds of part-whole relations in terms of ontological soundness, i.e., if the proposed concepts are meaningful in terms of real-world semantics. Finally, some UML stereotypes are proposed in order to provide syntactical representations to the proposed ontological distinctions.

Here we are only interested in those properties that are meaningful in terms of ontological correspondence. For that reason, we have chosen to ignore properties that are relevant only in terms of implementation decisions (e.g., ownership, propagation of operations, encapsulation, by-value or by-reference, used or not used).

In terms of the primary and secondary characteristics of part-whole relations there are three points of disagreement between our proposal and the one of (Opdahl & Henderson-Sellers & Barbier, 2001):

Emergent and Resultant properties

Both (Opdahl & Henderson-Sellers & Barbier, 2001) and (Wand & Storey & Weber, 1999) propose that we should only model a *thing* as an aggregate if we are interested in modeling its emergent and resultant properties. Emergent and Resultant properties are defined by Bunge in (1977, p.97) as follows: "P is a *resultant* or *hereditary property* of x iff P is a property of some component $y \in \mathcal{C}(x)$ (see definition 5.1) other than x; otherwise P is an *emergent* or *gestalt property* of x".

The UML class diagram depicted in figure 5.31 is an exact copy from (Opdahl & Henderson-Sellers & Barbier, 2001, p.391). In describing this model, the authors explicitly state that in the problem domain all *Departments* are aggregates of *Employees*. Nevertheless, Opdahl and colleagues deem unacceptable to add a part-whole between these two classes since the diagram would comprise no resultant/emergent property of Department relative to Employee. We strongly disagree with this view and we think that this restriction arises from a misinterpretation of Bunge's ontology.

According to Bunge, every aggregate certainly has emergent and resultant properties. However, his ontology makes explicit the distinction between the properties possessed by a thing and the representations of these properties, namely attributes. According to Bunge, there are no bare individuals, i.e., things without properties: a thing possesses at least one substantial property, even if we humans are not or cannot be aware of them. Humans get in contact with the properties of things exclusively via the things attributes, i.e. via a chosen representational view of its properties.

In summary, we agree that emergent/resultant properties are basic characteristics of part-whole relations, in the sense they are present in all of them. What we do not agree is to use the existence of resultant/emergent *attributes* as a criterion for deciding whether to represent part-whole relationships. In other words, emergent/resultant properties of aggregates will always exist but we do not have always to be interested in them and sometimes we cannot even be aware of them. Additionally, we think that the representation of these attributes is not a necessary condition for one to benefit from the representation of part-whole relations in terms of communicability, understanding and problem-solving.

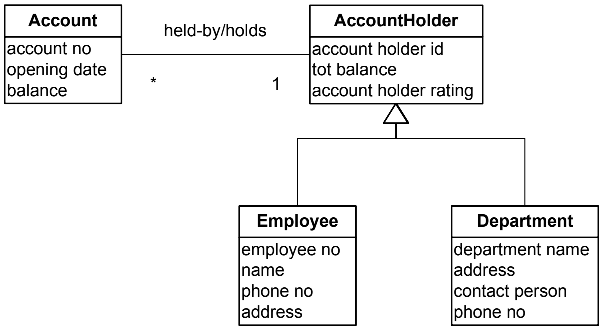


Figure 5-31 A part-whole relationship between Employee and Department belonging to the domain is not represented in the model because neither *emergent* nor *resultant* "properties" are represented (from Opdahl & Henderson-Sellers & Barbier, 2001, p. 391)

Mixing-up different properties

Lifetime dependency is a characteristic of part-whole relations with essential and/or inseparable parts. In this sense, we disagree with examples such as the one used in (Henderson-Sellers & Barbier, 1999) to justify the existence of parts that are separable, but that share the same destruction as the whole: *"a car wheel is independent of the car but if the wheel is in the car during the car's destruction then it is also destroyed"*. In this case, the wheel is clearly separable from the car, it just happened to be the same event that caused the destruction of both objects (had the wheel been separated from the car, the car's destruction would not propagate to the wheel).

This confusion seems to be motivated by an object-oriented programming bias towards conceptual modeling. Traditionally, in OO programming languages, an object can be made responsible for the destruction of other objects as a procedure for memory deallocation named *garbage collecting*. Thus, it can be warranted that an object X should trigger the destruction of other objects coupled with X in the moment of its destruction, even if the coupling is merely a contingent one.

The mixing of different conceptual properties due to an implementation-oriented attitude is not uncommon in conceptual modeling. Another example that is recurrent is the unfounded association between shareability and separability. For instance, (Saksena & France & Larrondo-Petrie, 1998) claim that "shared parts are necessarily separable", and that "sharing implies that if the membership of a part in one aggregate is nullified then that part can continue to exist." In other words, according to the authors, *shareability implies separability*. A converse confusion appears in (Snoeck & Dedene, 2001), in which the authors claim to present a problematic case of part-whole modeling w.r.t. separability. They use the following example of the relation between a paper and a journal issue: "once a paper has been accepted and published in a journal issue, it has become a part of this issue and is in addition *inseparable from the issue and cannot be shared by other issues*"(p.15). This example shows indeed a case of non-shareability. In fact, one could argue, a case of non-shareability w.r.t. to a given universal (see definition 5.9): although a paper cannot belong to different journals, it could perhaps be part of a *collected papers* series of some given authors.

As a case of non-shareability, this example is perfectly unproblematic, as it can be modeled in a standard way such as in the model of figure 5.32. Anyhow, the important point here is that the paper can be non-shareable without being inseparable. In this case one can perfectly argue that the paper is not *inseparable* from the journal, since it could perfectly well exist prior to the journal existence, and even if the journal issue goes out of existence it could still very well exist as part of the author's collected paper series[41].

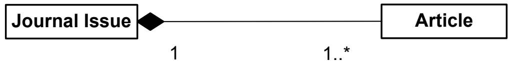


Figure 5-32 Standard UML notation for non-shareable parts

Examples of non-shareability without inseparability abound. To cite just another one: the relation between a Heart to a Human Body is typically conceived as one of non-shareability. Nonetheless, the heart is clearly not

[41] By the terms *Journal Issue* and *Article* here we do not refer to their physical copies (tokens), but to their logical existence.

an inseparable part of the body, since it can survive the extinction of the body. The example actually (fortunately) extends to most human organs, and that is what organ transplants is all about! Moreover, counterexamples of the converse also exist, i.e., there can be shareable parts that are inseparable to all (or some of) the wholes of which they are part. One of such examples is presented in figure 5.33: although a specific lecture of a regular course is an inseparable part of that course, it may be shared with any number of *studium generale* courses. In summary, shareability and separability are orthogonal secondary characteristics.

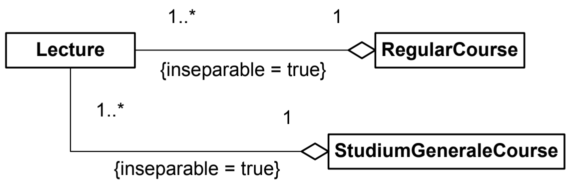


Figure 5-33 An example of wholes with shareable parts but that are inseparable

Transitivity

Opdahl and colleagues chose to exclude transitivity from the list of primary characteristics of part-whole relations, based on the problematic examples of transitivity paradoxes, such as those discussed in section 5.2.1.

Parthood, as a formal relation, is considered transitive by all theories of parts. In fact, both (Varzi, 2003) and (Simons, 1997) consider transitivity as intrinsic to the very meaning of part. It is true that there are examples of meronymic relations that when put together derive fallacious conclusions. But it is also the case that these meronymic conceptual relations *depart from the formal meaning that mereological binary relations intend to capture* (Johansson, 2004). In a purely mereological sense it is indeed the case that both the door and the handle are parts of house, or that soldiers are parts of battalions. Therefore, a theory of parthood relations for conceptual modeling must explicitly differentiate between the formal relations of parthood defined in formal mereologies and the conceptual relations of parthood that are employed in cognitive tasks. The conceptual relation of parthood should not be interpreted in the same way as its purely formal counterpart.

From a conceptual and computational point of view there are many benefits from reasoning on parthood transitivity. Examples include propagation of properties such as movement, rotation, and (in some cases) creation and destruction, among others. For this reason, on one hand, we agree with Opdahl and colleagues that transitivity is not a "primary characteristic" of *conceptual* part-whole relations, in the sense that it does not hold for all relations of this type. However, on the other hand, we defend that besides assuming non-transitivity for these relations, it is fundamental to understand why transitivity holds in some cases and not in others, and to determine the *contexts* in which part-whole relations are indeed transitive. Therefore, in this sense, the work presented here can be considered as an extension of the work of Opdahl and colleagues.

5.8 Final Considerations

In this chapter we have concentrated on giving examples of part-whole relations between substantials. However, this is far from implying that this type of relation only holds between individuals pertaining to this ontological category. Conversely, part-whole relationships clearly exist among other types of individuals. Examples of parthood can easily be found among processes and events (perdurants) (buying is part of shopping, singing "happy birthday" is part of birthday party), among conceptual spaces (e.g., the color space is composed of hue, saturation and brightness), moments (e.g., a musical chord is composed of at least three notes; a thought can have proper parts), time and space, etc. Therefore, in this sense, we disagree with the BWW approach (Evermann & Wand, 2001b), which defines parthood only between *things*, i.e., substantial individuals. The categories of perdurants, moments and conceptual spaces are defined in chapter 6.

The focus on substantials here is motivated by the main objectives of this thesis, which emphasizes on providing ontological foundations for structural conceptual modes. Nonetheless, we strongly believe that most of the results developed here can be carried to part-whole relations holding among individuals of different categories. For example, the basic axioms of MM also hold for parts of processes, which also have mandatory and contingent parts, and so forth. For a study of the part-whole relation among processes, we refer to (Simons, 1987).

One of the main objectives for the development of the typology of meronymic relations proposed by Winston, Chaffin and Herrmann (WCH) is to address the problem of fallacious transitive cases. As demonstrated in section 5.6, the typology of parthood relation that can be extracted from a revision of WCH's original account can be exploited to provide more direct guidelines for helping the user in the definition of these contexts. In chapter 7, we resume the discussion on this topic by proposing a number of visual patterns that can be used to identify transitivity context in parthood relations among complexes, which are the most typical sort of parthood relations to be found in conceptual modeling in computer science.

We believe that the problem of transitivity bears some strong connection to another (very difficult) philosophical question, namely, that of why are some parts essential and other parts accidental to a thing. As discussed in (Simons, 1987; Wiggins, 2001), the parts of a thing behave much like any other property of the thing. For this reason, parts are also determined by the nature of the particular or, to put in Aristotelian terms, by "what the particular is"? Since, in conformance with Wiggins, we consider that the *essential properties* of a thing are those implied by the (unique) *substance sortal* that it instantiates, the same applies to its essential parts.

This makes clear that any deeper discussion on the reasons for the essentiality of some parts requires a discussion on the essentiality of properties. In our opinion, such a discussion cannot be carried out avoiding the philosophical discussion on the very meaning of Kinds, which in turn, is related to the ancient issues of essentialism and the problem of universals (Armstrong, 1989). Some authors (most notably Boyd, 1991), deny the existence of essences in traditional terms (i.e., as a list of essential characteristics) in favour of what can be termed *nomological essentialism*. Boyd proposes a view of essence in terms of what he names *causal homeostasis*. For him, the essence of a thing is a network of *causal dispositions* relating the properties of the thing. He argues that: "kinds, properties, relations, etc. are natural if they reflect important features of the causal structure of the world… the clustering of properties is not merely chance coincidences…the presence of some properties tends to favor the presence of others where there are common underlying properties that tend to maintain the presence of the clusters." (cited by Keil, 1992).

This notion of essence as a cluster of causally related properties is akin to the notion of *organization* proposed by Maturana and Varela (Maturana & Varela, 1987; Varela, 1979) in their theory of Autopoiesis (as noticed by Simons, 1998). This view implies that an individual can change all its manifest properties (e.g., perceptual characteristics) and keep its identity as long as its organization is maintained. Actually, Boyd's argument is even stronger than that: some properties of the thing have to vary for the *homeostasis* to be maintained. This position is also shared by Bunge who states that the laws of a thing are essential to it (by definition of a law, it must hold necessarily), and the necessary properties of a thing are those which are lawfully related. In summary, both laws and lawfully related properties are essential. Since (i) the laws of an object are determined by its Natural Kind (Bunge, 1977, chapter 3), and (ii) a Natural Kind in Bunge is equivalent to what is termed here *substance sortal* (see discussion in section 4.5.2), we can conclude that indeed the necessary properties of a thing are determined by the (unique) substance sortal it instantiates.

In discussing individual adaptation, Bunge and more recently (Rowe & Leaney, 1997) and (Rowe & Leaney & Lowe, 1998) argue about change of properties in terms of Bunge's concept of the *nomological state space* of a thing. The idea is that a thing can only assume states that are contained in its nomological state space, i.e., *lawful states*. Suppose that a property *p* of a thing *x* changes, placing it an unlawful state *s*. This change must trigger changes in other property(ies), say *p'*, so that another lawful state *s'* can be reached. If no such compensatory change is possible, then object *x* will remain in a state that falls outside its nomological state space and, according to Bunge, "it will change its name". In our terminology, this means that *x* will change its identity and therefore will cease to exist as such. As a consequence, we can state that the essential parts of a thing are those that are directly related to the things's identity and, hence, those whose loss puts the thing in an unrecoverable unlawful state, i.e., in a state where no compensatory change exists to bring it back to its nomological state space.

This conception of essence finds support in many authors in cognitive sciences (Keil, 1992; Milikan, 1998) and it has been empirically supported by works such as (Schunn & Vera, 1995). In particular, Schunn & Verna provide some empirical evidence that human cognition employs *causal domain theories* (and not typicality or frequency of properties) as its most important principle of application, and this is the case for both endurants and perdurants, natural entities and artifacts, and for entities of familiar and unfamiliar kinds.

The cognitive psychologist Frank Keil (Keil, 1992) subscribes to the stronger idea that *homestatic property clusters* exist not only for natural but also for nominal kinds (Schwartz, 1980). He argues that it is unlikely that the elements in a nominal essence are fully arbitrary and defends that, in contrast, they are systematically connected to a set of real, or at least supposed, causal relations – not those in biology or physics but those governing human interactions. For instance, chairs have a number of properties, which are used to identify them. Although there may not be internal causal homeostatic mechanisms of chairs that lead them to have these properties, there may well be external mechanisms having to do with the form and function of the human body and with typical social and cultural activities of humans. For example, certain dimensions of chairs as well as the weight range they are supposed to support are determined by properties of the human anatomy. Some of the features of nominal kinds can be arbitrary. For instance, take the rules of traffic: the choice of driving on the right or left of the road seems a random choice. However, having a rule that decides which side to drive it is not, since there are good causal justifications related to the consequence of collisions to humans on certain velocities and the difficulties in responding quickly enough with such a rule at certain velocity. In summary, for Keil, although the definition of nominal kinds (including artifacts) is heavily dependent of human intentions, the latter are not in themselves arbitrary. According to him, perhaps a clearer distinction between natural and nominal kind is that for the former the causal homeostatic mechanisms are closely related to various domains of science (e.g. biology, chemistry, physics), whereas for the latter social and psychological domains of causality are involved.

According to (van Leeuwen, 1991), the principle of identity and, in particular, the principle of persistence that an individual obeys is implied by its principle of application. This is compatible with his view that endurants are *persisting integral wholes*, i.e., if the principle of application is determined by the presence of a *causal network of properties* (essence) then a continuant is an object that persists in time maintaining that network, or as system theorists would say, maintaining its *organization*. Moreover, in parity with Bunge, the individual *is the same* as long as this network is maintained. Deciding what is the *integral whole* that should persist is therefore the very task of the principle of application. This supports the thesis that the *principle of unity* of an individual is also strongly related to its principles of application and identity. In order to individuate something we must decide what its parts are, i.e., we must see it as an *integral whole*. Thus counting presupposes individuation, individuation presupposes unity and unity presupposes application. Therefore, it is no coincidence that the universals that carry principles of identity, persistence, individuation and counting are also exactly those that also carry a principle of unity, namely sortal universals.

Finally, if the principle of unity for an individual x is related to its principle of application (and so is its *unifying condition*), to determine "what x is" helps to determine what its parts are. In particular, if a principle of application is a causal network of properties then to determine x's parts is to determine the particulars that contribute to this network. Ergo, objects that are unified under the same unity condition are those that contribute to the causal network of properties that supply this condition. This argument seems to support the thesis defended in this chapter that transitivity holds within wholes that are unified by (refinements of) the same unity condition. In other words, material part-whole relations can be seen as embedding the implicit condition ϕ (see discussion in section 6.2.7) that their relata contribute to the same cluster of properties (nomological essence).

A number of results in cognitive psychology provide empirical evidence for some of the points aforementioned. For example, (Tversky, 1989) shows that taxonomies and partonomies are strongly interrelated, and that children are more likely to group taxonomically when instances share parts than when instances do not share parts. This clearly suggests the relation between a criterion of unity and one of application. Moreover, as discussed in chapter 4, the establishment of kind categories (substance sortals) in the language of thought is essential for the learning of proper names, common nouns and for individuation and categorization tasks in general. According to (Mcnamara, 1986), the first notion of kind which young infants construct is in association with the so-called *basic-level sortals* (Rosch et al., 1976). In taxonomy, basic-level sortals have a privileged status in many cognitive tasks, and it is at this level that subjects are first able to substantially produce attributes for the given categories. (Tversky & Hemenway, 1984) shows that most of these attributes are parts, and propose that parts "may underlie the informativeness of the basic level and account for the convergence of many cognitive operations at that level as well". Finally, (Hall, 1998) conducts empirical studies with regard to the perceived conceptual relation between the identity of an object and the change of its parts. As his results show, "knowledge about *specific kinds of objects* and their canonical transformations exerts an increasingly powerful effect, over the course of development, upon people's tendency to rely on continuity as a criterion for attributing persistence to objects that undergo change."

Finally, we summarize this discussion from a more pragmatic conceptual modeling point of view. Recently, many authors have proposed that an interesting way to conceptualize structure/organization of entities is to view it as a set of interrelated roles (Dignum, 2003; Odell & Parunak & Fleischer, 2003; Odell & Nodine & Levy, 2005). This possibility is also accounted for in (Odell & Bock, 1998), in which integral wholes are claimed to be *"aggregates of qua types"*. Although we mostly agree with this conception, there is still one point that should be rephrased. As discussed in chapter 4, roles are anti-rigid universals, and stating that wholes are unified by objects that play roles in its inner structure implies that all objects are only contingently part of this structure. We advocate that this is the case for some objects, namely mandatory parts[42]. However, for some objects this relation is not contingent, in which case one could say that they "play the roles" of themselves. These are the inseparable and/or essential parts. In summary, an object *x* is part of another object *y* if and only if it "plays a role" defined in the organization (context) of *x*.

We formally define the notion *qua individuals* in chapter 7. Moreover, in that chapter, we resume the discussion on the connection between qua individuals and part-whole relations.

---

[42]Theoretically they could also be merely contingent parts but this would demand the specification of zero minimum cardinalities in the (part-whole) relations between the aggregate and the role these parts play. This in turn would entail the creation of invalid models according to the postulates of chapter 4, since roles are relationally dependent entities.

Chapter 6

Properties

This chapter aims at providing ontological foundations for some of the most basic conceptual modeling constructs, namely *types*, *attributes*, *data types*, and *associations*.

Types (e.g., *Person* or *Car*), attributes (e.g., *being colored*, or *being happy*), and associations (e.g., *being married to*, *being enrolled at*) are all considered sorts of *universals*, i.e., predicative terms that can possibly be applied to a multitude of individuals. In chapter 4, we have focused our discussion on types. Universals such as attributes and associations are referred here by the general name *Property*. This chapter is therefore mostly about properties.

The notion of universal underlies the most basic and widespread constructs in conceptual modeling. Therefore, before we can provide ontological foundations for these constructs, we start here by investigating the nature of universals from a philosophical point of view. We start the chapter on section 6.1 by discussing the so-called *problem of universals* in philosophy. In particular, we analyze and criticize a specific theory of universals that underlies most current conceptual modeling languages, including the *semantic web languages*. In virtue of these criticisms, we justify the choice of some theoretical entities that are incorporated in our foundational ontology, which is fully elaborated in section 6.2.

In section 6.3, we apply the proposed ontological categories to interpret the modeling concepts of types, attributes, datatypes, and associations, and to provide methodological guidelines for their use in conceptual modeling.

In section 6.4, we present a more general discussion on related work in the conceptual modeling literature. Section 6.5 closes the chapter with some final considerations.

6.1 The Problem of Universals

In chapter 4 of this thesis we have discussed different types of universals, and how they could relate to each other. In particular, we saw that some universals, such as Man or Dog provide a principle of identity to the objects they classify. In contrast, other universals, such as Red, Square or Hard, do not provide such a principle. Strawson (1959) names these two types of classifiers *sortal* and *characterizing universals*, respectively.

Before proceeding, there is an important notion that should be defined, namely the distinction between *determinables* and *determinates* (Johnson, 1921). Universals can have different degrees of determinateness. Determinates can be understood as restrictions of determinables, providing a higher degree of specificity. For instance, *being colored* is a determinable and *being red* is a determinate for it. Since these are relative notions, *being red* can also be considered a determinable for *being scarlet* (one of its determinates). (Funkhouser, 2004) uses the terms *super-determinable* and *super-determinate* for the universals that are the root and the leaves of a specialization chain, respectively.

Now, take for example a sortal like Person. For every individual person, say John, there are many characterizing universals that apply to John, in virtue of John being of that specific kind. For example, John can be *1.80 meters tall*, *80 kilograms of weight*, be *29 years old*, and so forth. Notice that properties such as *being colored*, *being red*, *being-1.80-meters-tall* or *being married* are indeed sorts of universals, since they can be multiple instantiated in different individuals (i.e., they are repeatable). This gives us the idea that *sortal universals form clusters of characterizing ones*. Actually, most of the categories in the profile proposed in chapter 4 are typically used to model clusters of characterizing universals, with the exception of *mixins*. Sometimes characterizing universals are represented as mixins for the purpose of improving the structure of the resulting models, but more typically they are represented in conceptual models as attributes (e.g, color), in the case of super-determinables, or as attribute values (e.g, red), in the case of determinates. In fact, (Guarino & Welty, 2000) use the term *Attribution* for mixins, highlighting the correlation between the two representations of characterizing universals.

The issue of universal properties has been a topic of great interest (and controversy) throughout the entire history of western philosophy, from Plato and Aristotle to early analytic philosophers such as Russel and Wittgenstein, and contemporary philosophers such as Bunge (1977), Boyd (1991), Armstrong (1989, 1997) and Thomasson (2004). This topic is known in philosophy as *"the problem of universals"* (Loux, 2001) and can be summarized in the following manner. We know that proper names (e.g., Noam Chomsky or Spot) refer to individual entities, but what do general terms (or universal properties) refer to (if anything at all)? We classify objects as being of the same type (e.g., person) and use the same predicate or general term (e.g., red) to different objects. What exactly *is the same* in different objects that justify their belonging to the same category? In the sequel, we briefly present different approaches in answering these questions that amount to different philosophical theories of universals.

6.1.1 Lightweight Ontologies and the *Class* Theory of Universals

A first group of theories is constituted by what we name here *Class theories of universals* (henceforth, *C-theories*). In a C-theory, the meaning of a type is associated with that of class in the mathematical sense, i.e., roughly a set. Hence, if *a* and *b* are of the same type, it is because they belong to the same set X, where X is the ontological interpretation of that type.

In the sequel we discuss C-theories in some detail, mainly because there are many languages in conceptual modeling that commit to this view of universals. Examples include OWL (Bechhofer et al., 2004), LINGO (Falbo & Menezes & Rocha, 1998; Guizzardi & Falbo & Pereira Filho, 2002), CCT (Dijkman & Ferreira Pires & Joosten, 2001); Z (Spivey, 1988), ER (Chen, 1976), among others. The choice for an underlying class ontology in these languages is sometimes motivated by mathematical simplicity and convenience, sometimes related to practical performance trade-offs. For example, LINGO was designed with the specific objective of achieving a positive trade-off between expression power of the language and the ability to facilitate bridging the gap between the conceptual and implementation levels (a preoccupation that also seem to be present in Peter Chen's original proposal for ER diagrams). OWL, on the other hand, has been designed to explore the relation between a minimum expressiveness and computational tractability in the reasoning procedures. Although these choices can be justifiable in *lightweight* ontology modeling languages for practical reasons, there are many philosophical problems that would make them unsuitable as *foundational*[43] conceptual modeling languages.

To start with, a major problem with the simplest form of a C-theory is the difficulty in correlating classes and types. If C-theory is to be true, there should be a one-to-one correlation between the two. However, this correlation is not to be found. Take the set-theoretical semantics given to some of the modeling languages aforementioned (e.g., CCT):

- a type is interpreted as a set;

[43] See the distinction between lightweight and foundational ontologies in section 3.3.3 of this thesis.

– the subtyping relation is interpreted as the subset relation;
– the instantiation relation is interpreted as the membership relation.

In this case, the following problems are immediately found:

a1. If the subtyping relation between types A and B is interpreted as the subset relation between their extensions, then any type C with an empty extension (i.e., that has no instances) can be considered a subtype of any other type (the empty set is a subset of any other set). Moreover, any two types C and D with empty extensions (e.g., unicorn and centaur) become identical;

a2. Starting with an individual, for instance the person John, one can construct an infinite series of sets, all which have exactly the same content (Sowa, 2000): the singleton set {John}; the set {{John}}; the set {{{John}}}, {{{{John}}}}… and so forth. Although, in set-theoretic terms, these are all different entities, they correspond to the same entity in reality. In fact, this type of *ontological extravagance* of set-theory was one of the motivations for the proposal of formal mereologies (see section 5.1), since in extensional mereology there is no distinction between an entity A and the sum of its parts. This is captured in the following statement by (Goodman, 1956), one of the fathers of mereology: *"No distinction of individuals without distinction of content."*

a3. As discussed in chapter 4, the categorization mechanism employed by human cognition have the purpose of supporting learning, improving inductive knowledge, making memory and language possible, etc. Now, take the set formed by the last thought I had on last Christmas, the number two, the apple over my table and the last two minutes of a football game. This is a perfectly acceptable set (from a mathematical point of view), but one which has no place in a cognitive model of reality.

An attempt to solve some of these problems is made by a refinement of the pure C-theory, named the *Natural Class theory*. This theory was first proposed by philosopher Anthony Quinton (1957), who proposes a fundamental distinction in the world between *natural* and *unnatural* classes. According to Quinton, it is a feature of the world that things fall into natural classes, and this primitive notion of belonging to a natural class cannot be further analyzed. Thus, particulars, by virtue of falling *naturally* into a class, can be said to be of this or that type. In other words, the meaning of a type (e.g., horse) is the natural class of things that are instances of that type, i.e., the set of individuals that are members of the *natural class of Horses*. By recognizing that only certain sorts of classes are related to a type, the Natural Class theory can address the criticisms (a2) and (a3) above. In particular, the set mentioned in (a3) would lack naturalness and therefore would not be associated to a type.

To meet the criticisms stated in (a1), one could just go a little bit further and postulate that the empty class is not natural. However, underlying the criticisms in (a1) there is a more general problem, due to the extensional principle of identity of sets. This principle of identity, know as the *extensionality principle*, states that two sets are identical iff they have the same members. According to this principle then, the following consequences are implied:

b1. If a class is taken to be the ontological interpretation of a type, then the identity of classes become the identity of corresponding types. According to the extensionality principle, if a class gains or looses a member it becomes a different class. Take for example the type Electron. If the meaning of this type is the class of existing electrons, then addition or deletion of any electron would change the meaning of what is to be an electron. This is taken by many philosophers (and we agree) to be an absurd consequence;

b2. Any two classes that happen to be co-extensional must be considered identical. The case of empty classes mentioned previously in (a1) is just an extreme case of this problem. Take for example the types *human* and *featherless biped*, or *liquid* and *viscous.* These types have completely different intentions and can have different associated principles of identity (see chapter 4). The fact that two types happen to coincide in their extensions should not be sufficient for equating them;

b3. Suppose a subtyping relation between the types Organization and Group of People in the loose sense that "every organization (e.g., musical group, football team, bird-watchers association) is a group of people". If subtyping is interpreted as the subset relation, then this is necessarily true in this case. However, as discussed in chapter 4, it can be the case that these universals supply incompatible principles of identity for their instances. For instance, if a football team exchanges one of its members it is still the same organization but it becomes a different group of people. As a consequence, despite of apparently sharing the same extension these two types cannot be related via subtyping, since they carry incompatible principles of identity.

Consequently, a minimum requirement of a set-theoretic conceptual modeling language would be the explicit consideration of *possible worlds*. In other words, the extension of natural class should consider individuals in all possible worlds[44]. This way, the theory can exclude all cases where the extensions of classes coincide only contingently. To cite one last example, it is a fact in our world that the classes of cordates (individuals that have a heart) and of renates (individuals that have a kidney) coincide. Now, suppose that this is just an evolutionary accident, in the sense that there is no fundamental *direct functional dependence* (see chapter 7) between hearts and kidneys. Therefore, one can conclude that there are possible worlds in which the extensions of cordate and renate do not coincide and, since the two classes do not coincide necessarily, they cannot be said to be the same. Moreover, by considering Natural classes as the extension of types in every possible world, the theory becomes more refined, and can choose to deem as unnatural empty classes, only those that are necessarily empty. Finally, it can also escape criticism (b1) regarding the dependence that the meaning of type has on its members in its extensional counterpart. Notice that there is support in the cognitive psychology literature for taking the meaning of a general term to refer to particulars in counterfactual situations. For example, in the second of the psychological claims defended by John Mcnamara in (Mcnamara, 1994) it is proposed that "the word DOG refers to all dogs that ever existed, that ever will be, or that could ever have been. Not to the dogs that currently exist."

Extending the notion of natural class by considering possible words would solve some of the problems related to the ontological semantics of languages such as EER, Z, CCT and LINGO, when considered as conceptual modelling languages. However, there is still another philosophical disadvantage of the natural class approach. It is related to the so-called *direction of explanation* (Amstrong, 1989). Consider the following question: Is the sort of thing something is – say a horse - because it is a member of the class of Horses? Or it is rather a member of the class of Horses because it is a horse? In conformance with Armstrong, we claim the latter is the correct direction of explanation.

This becomes more evident when considering the *causal action* of things. Things act causally by virtue of the *properties* they have, not because they belong to certain classes. For instance, the fire makes the water boil by virtue of its temperature and the object depresses the scale by virtue of its mass. It is the individual *four kilograms object* that acts on the scale in a way that is completely independent of the other four kilograms objects that exist (or may have existed). In other words, it is not the case that the whole class of four kilograms object is relevant for the causal action of an object on another, but the presence of certain property of the object. Finally, from an epistemological point of view, acknowledging the priority of properties over

---

[44] See our function *ext* (as opposed to ext_w) in section 4.1.

classes seems to afford a more intuitive explanation for our ability to classify things. Suppose that we face a new particular of a familiar type and that we have no difficulty in correctly classifying it as being of that type. If this is the case, what happened is that the object has acted on us by virtue of certain properties. To use a simplistic example, we correctly apply the term Red to an object because we recognize the color red in it, not because we recognize it as a member of a (possible infinite) class of objects.

We therefore propose an account of universals that recognize the precedence of properties over classes, and the precedence of intention over extension. Therefore, in the world view proposed here objects are endowed with a number of properties that make them what they are.

There is a final problem with C-theories that is worth mentioning, since it is related to a recurrent practice in some modeling languages, in particular, in the so-called *Semantic Web Languages* (Bechhofer et al., 2004; McGuinness et al., 2002a,b). This problem becomes even more evident in a particular C-theory that we name here the *predicate theory.* Let us first briefly explain the predicate theory.

The predicate theory of universals belongs to the group of the so-called *nominalist* theories. Nominalism (from the latin word *Nomem* for *name*) is a project of explaning the unity of the tokens falling under a certain type by some linguistic device (Armstrong, 1989). According to it, two things are of the same type iff the same predicate applies to them. To put it differently, the only thing that is universal among, for example, two red objects x and y is the fact that we apply the linguistic predicate "red" to both of them. This is a serious problem from a philosophical point of view because it commits to the idea that there are no real universal properties shared by individuals outside the minds of cognitive subjects. Even worse, it equates the set of universal properties to the set of predicates available in a language. Other C-theories such as the natural class theories do not make this commitment. The naturalness of a class (or lack of it) is taken to be a feature of reality and, thus, independent of language.

However, there is a problem with predicate theory that is also common to the simplest form of C-theories. There are many predicates that can be constructed in the language that do not correspond to properties in reality, in particular, this is the case for the *negative* and *disjunctive predicates* (Bunge, 1977). Take, for example, the complex predicate $P(x) =_{def} C(x) \vee M(x)$ which is true for all objects x that have either an electrical charge or a mass. Although, P is a perfectly good predicate, it is not a universal (type) in the philosophical sense. To see that, consider the following situation: let a and b be two individuals such that both $P(a)$ and $P(b)$ hold. $P(a)$ holds because a has an electrical charge and $P(b)$ holds because b has a mass. Now, from the fact that the disjunctive predicate P applies to both a and b, can one say that in any serious sense that a and b have something in common? Asides from that, as put by (Armstrong, ibid.), there is a close link between universals and causality. If a thing instantiates a certain universal then, in virtue of that, it has a certain power to act in a certain way. Moreover, different universals bestow different powers. For instance, by virtue of having a certain mass, *b* can act upon a scale. Likewise, by virtue of having a certain charge, *a* can repel certain things. Now, were P a genuine predicate of *a* (or *b*), it would add something to *a*'s (or b's) capacity to act. This is clearly not the case. A similar case can be made for opposing negative predicates. Although the predicate $Q = \neg C(x)$ is a perfectly acceptable predicate, it does not correspondent to a bonafide universal in reality, since: (i) there is nothing necessarily in common between two things to which Q applies; (ii) there are no causal powers which are bestowed by not instantiating a universal. In summary, in a foundational ontology, there should be no negative or disjunctive universals (Bunge, 1977; Armstrong, 1989; Schneider, 2002).

We can summarize the discussion in this section by considering the following: mathematical set (class) theories have been developed in advanced forms for the last hundred years. The idea that we can give an account in set-theoretical terms of what is to be of certain type is attractive to logicians and mathematically inclined computer scientists. However, as we have made clear in the course of this section, developing purely set-theoretical semantics for classifiers in conceptual modeling amounts to one of the exemplar cases in which formal semantics, for the sake of mathematical convenience, has been given an unfortunate precedence over real-world semantics.

In a foundational ontology, the account of universals must be an intensional, not an extensional one. As discussed in depth in chapter 4, it is not the case that all properties that apply to an individual should be seen as ontologically equivalent. Some universals are sortal, thus providing a principle of individuation, persistence and identity. Other universals are merely characterizing. Some properties apply contingently to their instances, others essentially. In line with several proposals in the literature (e.g, Guarino & Welty, 2000, 2002b, 2004; Welty & Guarino, 2001; Gupta, 1980; van Leeuwen, 1991), we advocate that a full account of these distinctions is a fundamental feature for any conceptual modeling theory of universals. One, which is, unfortunately, ignored in many current conceptual modeling and ontology representation languages.

6.2 Basic Ontological Categories

In this section, we start collecting some categories that have been proposed in previous chapters and, in the light of what has been discussed in the previous section we aim at constructing the backbone of our foundational ontology.

As discussed in the previous section, a bicategorial ontology only based on sets and its members (as supplied by set theory) does not constitute a suitable inventory of formal entities that can be used to model reality. The approach presented here preserves set theory as a part of the foundational ontology proposed. Thus it accepts set membership as one of the ontologically basic (formal) relations that are adopted. At the same time, however, it introduces a number of other ontologically basic entities and relations. These entities are discussed in the sequel.

6.2.1 Sets and Urelements

A fundamental distinction in this ontology is between the categories of the so-called *urelements* and *sets*. Urelements are entities that are not sets. They form an ultimate layer of entities without any set-theoretical structure in their build-up, i.e., neither the membership ($\in$) relation nor the set inclusion ($\subseteq$) relation can unfold the internal structure of urelements.

We assume the existence of both urelements and sets in the world and presuppose that both the impure sets and the pure sets[45] constructed over the urelements belong to the world. This implies, in particular, that the world is closed under all set-theoretical constructions.

Here, we do not discuss any particular version of set theory, but only employ it to formally characterize aspects of other entities in our ontology. Therefore, we use common set theoretical operations such as element membership, set inclusion and proper inclusion ($\subset$), union ($\cup$), intersection ($\cap$) and difference ($\setminus$) without providing any formalization. This is partly because many formal specifications of set theory already exist and are widely available (e.g., Heller et al., 2004), but also because this neutrality allows the ontology to be extended in that regard to suit particular needs, where one could choose for particular versions of set theory. Examples are the systems of Zermelo-Fraenkel and Neumann-Bernays-Gödel (Weisstein, 2004), but also theories of quasi-sets (Kreuse, 1992) can in principle be adopted[46]. In particular, sets (and associated notions) are assumed to play a fundamental role here in the formal

[45] Pure Sets can only contain other sets as members. Impure Sets can also contain urelements (Weisstein, 2004).

[46] In the current version of our ontology we exclude the so-called quasi-objects, i.e., individuals that have determinate countability but indeterminate identity (Lowe, 2001). The reason for that lies in the purpose of our ontology (a commonsense based one) and the associated belief that quasi-objects do not exist in the mesocopic level. However, a theory of quasi-sets would have to be incorporated in a possible extension of our theory that admits quasi-objects.

characterization of conceptual spaces, quality dimensions and quality domains (see section 6.2.5).

Urelements are classified into two main categories: *individuals* and *universals*. A urelement has to be either an individual or a universal, but not both. This can be expressed by the following axioms:

(1). $\forall x\ (Ur(x) \leftrightarrow Ind(x) \lor Univ(x))$
(2). $\neg \exists x\ (Ind(x) \land Univ(x))$

The categories of Set and urelements (individuals and universals) and some of their interrelations are illustrated in figure 6.1.

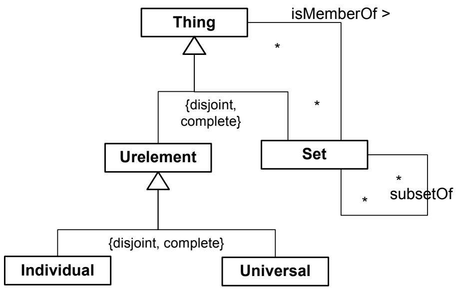


Figure 6-1 Fundamental distinction between sets and urelements

6.2.2 Individuals

Traditionally, in the philosophical literature, there is a fundamental distinction in the category of individuals between enduring and perduring entities (see, for instance, the distinction between endurants and perdurants in DOLCE (Masolo et al., 2003, or the distinction between 3D (presentials) and 4D individuals (processes) in GOL (Heller et al., 2004; Heller & Herre, 2004). Classically, the distinction between enduring and perduring individuals (henceforth named *endurants* and *perdurants*) can be understood in terms of their behavior w.r.t. time. Endurants are said to be wholly present whenever they are present, i.e., they *are in time*, in the sense that if we say that in circumstance c_1 an endurant *e* has a property P_1 and in circumstance c_2 the property P_2 (possibly incompatible with P_1), it is the very same endurant *e* that we refer to in each of these situations. Examples of endurants are a house, a person, the moon, a hole, an amount of sand. For instance, we can say that an individual John weights 80kg at c_1 but 68kg at c_2. Nonetheless, we are in these two cases referring to the same individual, namely the person John.

Perdurants are individuals composed of temporal parts, they *happen in time* in the sense that they extend in time accumulating temporal parts. Examples of perdurants are a race, a conversation, a football game, a symphony execution, a birthday party, the Second World War and a business process. Whenever a perdurant is present, it is not the case that all its temporal parts are present. For instance, if we consider the business process "buy product" at different time instants when it is present (figure 6.2), at each time instant only some of its temporal proper parts are present. As a consequence, perdurants cannot exhibit change in time in a genuine sense since none of its temporal parts retain their identity through time. Whereas for an endurant we can say that the very same individual John changes his weight from from 80 kg in c_1 to 68 kg in c_2, if we say that "buy product" has the property of being electronically secure at c_3 but non-secure in c_1, there are different proper parts of "buy product" that exhibit these properties. In a figure 6.2, "buy product" is an instance of a social process (the Consumer-Buyer Behavior model, see for instance, Almeida & Guizzardi & Pereira Filho, 2000) whereas its temporal proper-parts are actions, i.e., intentional events (atomic processes) performed by (physical or social) agents.

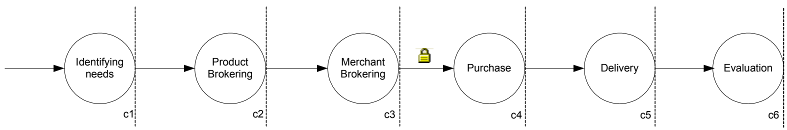


Figure 6-2 A "buy product" business process execution

As previously mentioned, the urelement part of our foundational ontology complements set theory (and, thus the current set theoretical approaches present in the literature) by encompassing a number of basic ontological entities. In conformance with the motivations presented in chapter 3, the ontology proposed here accounts for a descriptive commonsensical view of reality, focused on structural (as opposed to dynamic) aspects. For this reason, our foundational ontology shall be an *ontology of endurants*. Therefore, from now on, we focus our discussion on endurant individuals and endurant universals.

The core of the urelement fragment of our ontology amounts to a so-called *Four-category ontology*. The idea of an ontology centered on the four specific categories highlighted in figure 6.3 comes originally from the second chapter of Aristotle's *Categories*[47]. However, it finds support in many works in contemporary philosophical literature (Lowe, 2001; Mulligan & Simons & Smith, 1984; Smith, 1997; Neuhaus & Grenon & Smith, 2004) and, in particular, in some of the foundational ontologies developed in

---

[47] See, for example, the 1984 english translation (Aristotle, 1984).

computer science (Heller et al., 2004; Herre & Heller, 2004; Schneider, 2002; see also the Basic Formal Ontology in Masolo et al., 2003). The categories comprising this ontology are two pairs individuals-universals, namely substantial and substantial universals; moments and moment universals. Individuals are discussed in the sequel. Universals are discussed in section 6.2.5.

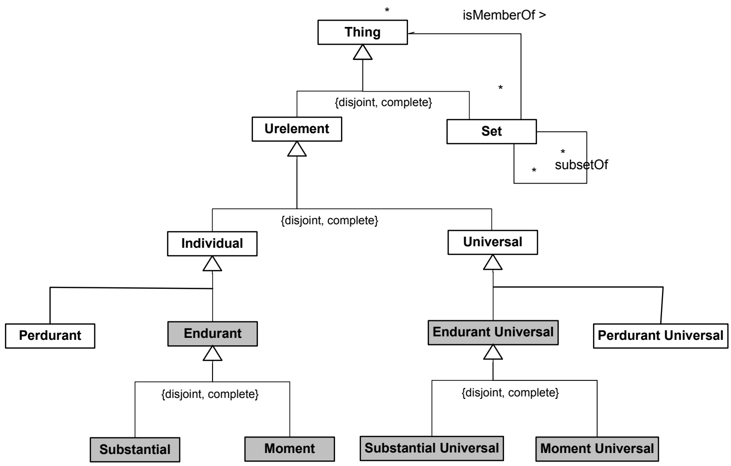


Figure 6-3 The urelement fragment on the proposed foundational ontology centered in a four-categorical account

6.2.3 Moments

The word *Moment* is derived from the german *Momente* in the writings of Husserl and it denotes, in general terms, what is sometimes named *trope* (Williams, 1966; Schneider, 2003b), *abstract particular* (Stout, 1921; Campbell, 1990), *mode* (Lowe, 2001; Schneider, 2002), *particular quality*, *individual accident*, or *property instance*. In the scope of this work, the term bears no relation to the notion of time instant in ordinary parlance. The origin of the notion of moment lies in the theory of individual accidents developed by Aristotle in his Metaphysics and Categories. For him, an accident is an individualized property, event or process that is not a part of the essence of a thing. We here use the term "moment" in a more general sense and do not distinguish *a priori* between essential and inessential moments.

As pointed out by (Schneider, 2002), there is solid evidence for moments in the literature. On one hand, in the analysis of the content of perception (Lowe, 2001, p. 205; Mulligan & Simons & Smith, 1984, p. 304-308), moments such as colours, sounds, runs, laughter and singings are the immediate objects of everyday perception. On the other hand, the idea of moments as truthmakers (Mulligan & Simons & Smith, 1984, p. 295-

304) underlies a standard event-based approach to natural language semantics, as initiated by Davidson (1980, pp. 118-119) and Parsons (1990, chaps. 1-3).

The notion of moment employed here comprises:

1. *Intrinsic Moments*: qualities such as a color, a weight, a electric charge, a circular shape; modes such as a thought, a skill, a belief, an intention, a headache, as well as dispositions such as the refrangibility property of light rays, or the disposition of a magnetic material to attract a metallic object;

2. *Relational Moments (or relators)*: a kiss, a handshake, a covalent bond, but also *social objects* such as a flight connection, a purchase order and a commitment or claim (Wagner, 2003; Guizzardi & Wagner, 2005a).

An important feature that characterizes all *moments* is that they can only exist in other individuals (in the way in which, for example, electrical charge can exist only in some conductor). To put it more technically, we say that moments are *existentially dependent* on other individuals (see definition 5.10, in chapter 5), named their *bearers*. Existential dependence can be used to differentiate intrinsic and relational moments: intrinsic moments are dependent of one single individual; relational moments depend on a plurality of individuals.

Existential dependency is a necessary but not a sufficient condition for something to be a moment. For instance, the temperature of a volume of a gas depends on, but is not a moment of its pressure. Thus, for an individual x to be a moment of another individual y (its bearer), a relation of *inherence* – sometimes called *ontic predication* – must hold between the two, symbolized as $i(x,y)$. For example, inherence glues your smile to your face, or the charge in a specific conductor to the conductor itself. In summary, moments are *ways things are* (Armstrong, 1989; Lowe, 2001) and, hence, cannot be conceived independently of the particulars they inhere in.

Inherence is an irreflexive, asymmetric and intransitive relation between moments and other types of endurants. These formal properties are represented in the formulas (5), (6) and (7) below, respectively. Additionally, as expressed in formula (4), inherence is a special type of existential dependence relation between particulars (symbolized as *ed* below):

(3). $\forall x,y\ (i(x,y) \rightarrow \mathrm{Moment}(x) \wedge \mathrm{Endurant}(y))$
(4). $\forall x,y\ (i(x,y) \rightarrow ed(x,y))$
(5). $\forall x\ \neg i(x,x)$
(6). $\forall x,y\ (i(x,y) \rightarrow \neg i(y,x))$

(7). $\forall x,y,z\ (i(x,y) \wedge i(y,z) \rightarrow \neg i(x,z))$

According to (3), moments inhere in other endurants, which can themselves be moments. An example of moment inhering in another moment is the radius of a circular form. The infinite regress in the inherence chain is prevented by the fact that there are endurants that cannot inhere in other individuals, namely, *substantials* (see section 6.2.4). We can, thus, formally characterize a moment as an individual that inheres in (and, hence, is existentially dependent upon) another individual:

Definition 6.1 (Moment): A moment is an endurant that *inheres in*, and, therefore, is *existentially dependent* of, another endurant. Formally,

(8). $Moment(x) =_{def} Endurant(x) \wedge \exists y\ i(x,y)$

■

In our framework we adopt the so-called *non-migration* (Guizzardi & Herre & Wagner, 2002a) or non-*transferability* (Martin, 1980) *principle*. This means that it is not possible for a moment *m* to inhere in two different individuals *a* and *b*:

(9). $\forall x,y,z\ (Moment(x) \wedge i(x,y) \wedge i(x,z) \rightarrow y = z)$

If a moment x inheres in an individual y, y is termed the *bearer* of x and is defined as follows:

Definition 6.2 (Bearer of a Moment): The bearer of a moment x is the unique[48] individual y such that x *inheres in* y. Formally,

(10). $\beta(x) =_{def} \iota y\ i(x,y)$

■

This characteristic of moments seems at first counterintuitive. For example, if we have two particulars *a* (a red apple) and *b* (a red car), and two moments m_1 (particular redness of *a*) and m_2 (particular redness of *b*), we consider m_1 and m_2 to be different individuals, although perhaps qualitatively indistinguishable. What does it mean then to say that *a* and *b* have the *same* color? Due to (9), sameness here cannot refer to strict (numerical) identity, but only to a qualitative one (i.e., equivalence in a

[48]The iota operator (ι) used in a formula such as $\iota x\varphi$ was defined by Russel in (Russel, 1905) and implies both the existence and the uniqueness of an individual x satisfying predicate φ.

certain respect). In conformance with DOLCE (Masolo et al., 2003a), we distinguish between the color of a particular apple (its quality) and its 'value' (e.g., a particular shade of red). The latter is named *quale*, and describes the position of an individual quality within a certain *quality dimension*. The notions of quale and quality dimension are discussed in depth in section 6.2.6.

Figure 6.4 depicts the inherence relation between moments and their bearers. Relators and modes are further discussed in section 6.2.7.

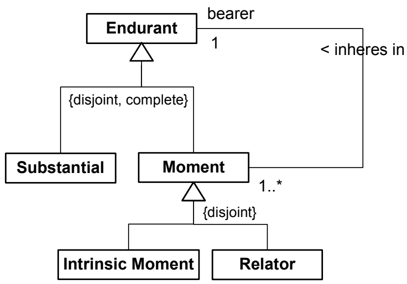


Figure 6-4 Moments and their unique bearers

6.2.4 Substantials

In the previous section, we have formally stated that moments inhere in other individuals, forming a chain of inherence that ends in a *substantial*. Since inherence is a sort of existential dependence, we have that all moments are existentially dependent on other individuals. Substantials, in contrast, by not inhering in anything, enjoy a higher degree of independence. We, thus, define the category of substantials as follows.

Definition 6.3 (Substantial): A substantial is an endurant that does not *inhere in* another endurant, i.e., which is not a moment. Formally,

(11). Substantial(x) $=_{def}$ Endurant(x) $\wedge$ $\neg$Moment(x)

■

Substances are individuals that posses (direct) spatial-temporal moments and are founded on matter. They can be further classified in either *Amounts of matter* or *Objects*. The distinction is made based on whether members of these categories satisfy a unity condition. *Amounts of Matter* (are also known as Stuff) are substantials with no unity; all their parts are essential, i.e., they are mereologically invariant: the identity of an amount of mater is determined by the sum of its parts and, thus, a change in one of the parts changes the identity of that particular (see section 5.5.1). Since countability is strongly related to unity (see section 5.8), amounts of matter are individuals with determinate principles of identity but with indeterminate counting principles. (Lowe, 2001) terms these entities *quasi-objects*. Examples of *Amounts of Matter* are individuals linguistically referred by mass nouns such as "sugar", "sand", and "gold".

Objects, conversely, are substantials with unity, i.e., integral wholes unified by a certain unity criteria (see section 5.3). Contrary to amounts of matter, it is not the case that all parts of an object are essential. Instead of necessarily obeying a mereological principle of identity, the principle of identity of objects is supplied by the kinds (substance sortals) that they instantiate. These principles of identity determine which parts are essential, mandatory or merely contingent. Examples of objects include ordinary mesoscopic entities that are linguistically referred to as count nouns, such as a dog, a house, a hammer, a car, Alan Turing and The Rolling Stones but also *Fiat Objects* such as the North-Sea and its proper-parts, postal districts and a non-smoking area of a restaurant (Smith, 1994).

We also consider as objects those parasitic substantials such as stains, edges, bumps, which are named features in DOLCE, but also what (Pribbenow, 2002) terms a *negative object* (e.g., a hole, the interior of a drawer). As a consequence, the relation between these parasitic entities and their hosts is one of *inseparable parthood* (see section 5.4.2) not one of inherence. For instance, we say that a hole in a piece of cheese is an inseparable part of the cheese, as opposed to one of its moments.

Objects can also have parasitic substantials as essential parts. For instance, the key whole of a locker may be considered an essential part of the locker. In (Guizzardi & Wagner, 2005b), we make a further distinction between two types of objects: (i) agents - to which we can ascribe mentalistic notions (mental moments) such as beliefs, desires and intentions, and non-agentive objects. Here, in contrast, we shall not consider this distinction and, thus, we concentrate only on the common properties of objects.

As discussed in chapter 5, conceptual models require that instantiated objects have determinate identities. For this reason, in order to represent *amounts of matter*, we must refer to them in a special sense, namely, that of a maximally-self-connected amount of matter. These objects, termed in chapter 5 *quantity* are genuine objects, possessing both determinate identity and determinate countability.

Figure 6.5 below depicts the types of substantials considered here and their relation to moments.

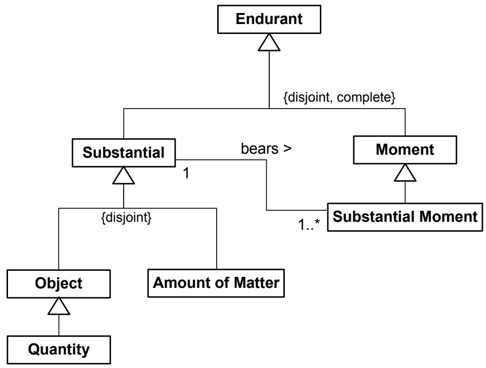


Figure 6-5 Types of Substantials and their relation to moments

Figure 6.5 also illustrates the *bears* relation between a substantial and its moments. This relation is the inverse of the *inheres in* relation shown in figure 6.4. As moments must inhere in some in individual (not necessarily a substantial), substantials must bear some moments, i.e., *"there are no propertyless individuals"* (Bunge, 1977), or bare particulars (see section 4.2). This is expressed in formula (12) below. We name *substantial moments* those qualities that necessarily inhere in a substance.

(12). $\forall x\ \mathbf{Substantial}(x) \rightarrow \exists y\ (\mathbf{Moment}(y) \wedge i(y,x))$

Finally, in the beginning of this section we state that substantials enjoy a higher degree of independence when compared to moments. Can we make a stronger statement? Can we say that substantials are existentially independent from all other individuals?

If we take the notion of existential dependency that we give in definition 5.10 in chapter 5, the answer is no. Existentially dependence has been defined as follows:

Definition 5.10 (existential dependence): Let the predicate ε denote existence. We have that an individual *x* is *existentially dependent* on another individual *y* iff, as a matter of necessity, y must exist whenever x exists, or formally

$$ed(x,y) =_{def} \Box(\varepsilon(x) \rightarrow \varepsilon(y))$$

■

Now, there are certainly pairs (x,y) where x is a substantial that satisfy ed(x,y). For example, if y is any of the essential moments of x. Moreover, even if both x and y are substantials, ed(x,y) can be satisfied. Take for example a substantial and any of its essential parts. Or, alternatively, a substantial x and another object y of which x is an inseparable part (see definition 5.15). However, we can say that if x and y are two substantials and they are disjoint then they must also be independent from each other (symbolized as *indep*):

(13). $\text{indep}(x,y) =_{def} \neg ed(x,y) \land \neg ed(y,x)$
(14). $\forall x,y\ \text{Substantial}(x) \land \text{Substantial}(y) \land (x \int y) \rightarrow \text{indep}(x,y)$

For example, a person depends on her brain, and a car depends on its chassis. However, a person (car) does not dependent on any other substance which is disjoint from her (it). Notice that formula (14) also excludes the case of mutual existential dependence between substantials that share a common essential part (see chapter 5).

6.2.5 Universals

To complete the Aristotelian Four-Categories ontology depicted in figure 6.3, we consider the existence of both *substantials universals* and *moment universals*. We use the term universal in a broader sense than it is sometimes used in the philosophical literature, such as for instance in (Armstrong, 1989). Therefore, we do not necessarily commit to existence of universals as *abstract entities that are multiple instantiated in several individuals*. Instead, we employ the term in a sense which is equivalent to term *Category* in (Heller et al., 2004). The position should be made clearer in the course of this section. For now, a universal can be considered simply as something (i) which can be predicated of other entities (or, in the Aristotelian sense, *said of* or attributed to other entities) and (ii) that can potentially be represented in language by *predicative terms*. In summary, as a synonymous of *type* as we have used it in chapter 4.

We use the symbol :: to denote the instantiation relation, a basic formal relation defined to hold between individuals (first argument) and universals (second argument). Hence, when writing x::U we mean that x is an instance of U or that x has the property of being a U:

(15). $\forall x,U\ x::U \rightarrow \text{Individual}(x) \land \text{Universal}(U)$

The difference between substantial and moment universals can be characterized as follows (Schneider, 2002):

Definition 6.4 (Substantial and Moment Universals): A Substantial universal is a universal that is instantiated only by substantial individuals. Analogously, a moment universal is a universal that is only instantiated by moment individuals. Formally,

(16). $\text{SubstantialUniversal}(U) =_{def} \text{Universal}(U) \land$

$$\forall x\ (x::U \rightarrow \text{Substantial}(x))$$

(17). MomentUniversal(U) $=_{def}$ Universal(U) $\wedge$ $\forall$x (x::U $\rightarrow$ Moment(x))

■

Although we want to avoid making unnecessary commitments to particular theories of universals, we believe that some interpretations for the nature of universals (and the corresponding theories) must be explicitly ruled out. To begin with, for all the reasons discussed in section 6.2.1, we exclude all sorts of *class and predicate nominalisms* (including the *natural class theory*). These theories are named in (Armstrong, 1989) *Blob theories* because of their failure to account for the existence and priority of properties. In conformance with Armstrong we defend that individuals belong to a certain category because they share a number of properties, not the other way around. For this reason, we also exclude another type of Blob theory that has not been discussed so far, namely, the classical formulation of the *resemblance theory* (Price, 1953).

Traditionally, class theories of universals are extensional theories. Thus, as discussed in section 6.2.1, two classes are identical if they apply to the same individuals. Here, we reject this strong nominalist idea that equates universals with sets. For every universal U there is a set *Ext(U)*, called its *extension*, containing all instances of U as elements. However, even if two universals U_1 and U_2 have identical extensions ($Ext(U_1) = Ext(U_2)$), they are not necessarily considered to be identical.

The identity of universals instead should be analyzed in terms of the identity of the supplied principles of application and identity, and/or in terms of the causal powers bestowed by them (Armstrong, 1989). However, a fuller consideration of this issue falls outside the scope of this thesis. The following definition relates the instantiation relation, universals and their extensions:

Definition 6.5 (Extension of a Universal): The extension of a universal U is the set S that contains all instances of U, and only them. Formally,

(18). Ext(U) $=_{def}$ $\{x \mid x::U\}$

■

In chapter 4, we discuss the ontological distinction between sortals and non-sortals. There is, in particular, an intimate relation between the category of non-sortals named mixins (or attributions) and moment universals. This distinction is also present in Aristotle's original differentiation between what is *said of a subject* (*de subjecto dici*), denoting instantiation and what is *exemplified in a subject* (*in subjecto est*), denoting inherence. Thus, the linguistic difference between the two meanings of the copula "is" reflects an ontological one. For example, the ontological interpretation of the sentence "Jane is a Woman" is that the object Jane *instantiates* the (substantial) universal Woman. However, when saying that "Jane is tall" or "Jane is laughing" we mean that Jane *exemplifies* the moment universal Tall or Laugh, by virtue of her specific height or laugh. Thus, in pace with (Schneider, 2002), besides instantiation, we recognize another formal relation that can obtain between individuals and universals, namely, the relation of *exemplification*:

Definition 6.6 (Exemplification): An individual x exemplifies the moment universal U iff there is a particular moment y instance of U that inheres in x. Formally,

(19). exemplification(x,U) $=_{def}$ MomentUniversal(U) $\wedge$ $\exists$y (i(y,x) $\wedge$ y::U)

■

In formula (19) above the variable x is restricted to the category of endurants but not to its subcategory of substantials. This is because, since moments can also inhere in moments, the exemplification relation can also exist between moments and moment universals. For example, in "Jane is laughing", Jane's laugh can exemplify the moment universal *frankness* (Schneider, 2002), or in "The ball is red", the color of the ball can exemplify the universal beauty.

As mentioned above, universals are not sets. Differently from sets, they are defined intensionally. According to (Guizzardi & Herre & Wagner, 2002a,b; Guizzardi & Wagner & Herre, 2004), we capture the intension of a universal by means of an axiomatic specification, i.e., a set of axioms that may involve a number of other universals representing its essential features. A particular form of such a specification of a universal U is called an *elementary specification*.

Definition 6.7 (Elementary Specification): An elementary specification of a universal U consists of a number of universals $U_1,\ldots,U_n$ and corresponding functional relations $R_1, \ldots, R_n$ which attach instances from the U_i to instances of U, expressed by the following schema:

(20). $\forall a(a :: U \rightarrow \exists e_1 \ldots e_n \bigwedge_{i \leq n}(e_i :: U_i \wedge R_i(a, e_i)))$

The universals $U_1,\ldots,U_n$ used in an elementary specification are called *features*. A special case of an elementary specification is a *intrinsic moments specification* where $U_1,\ldots,U_n$ are intrinsic moment universals.

■

The relation between a universal and the features in its elementary specification is one of *characterization*:

Definition 6.8 (Characterization): A universal U is characterized by a moment universal M iff every instance of U exemplifies M. Formally,

(21). $\mathbf{characterization(U,M) =_{def} Universal(U) \wedge MomentUniversal(M) \wedge \forall x\ (x::U \rightarrow \exists y\ y::M \wedge i(y,x))}$

■

Finally, there is another sort of *Blob theory* that we have not explicitly excluded yet, namely, the theory of universals known as *Platonic realism*, or simply, *Platonims* (Balaguer, 2004). Platonism amounts, in a nutshell, to the following: (a) universals are self-existent, ideal and external to individuals, and (b) universals precede their instances but can be exemplified by the latter (Bunge, 1977, p.103). Due to the independence of universals from their instances, Platonic realism leaves open the possibility for the existence of universals that have no instances. According to Bunge, *"there is not a shred of empirical evidence for the hypothesis that forms [i.e., universals] are detachable from their carriers [e.g., their bearers]"*. Therefore, we take here an Aristotelian view of universals w.r.t. accepting the so-called *principle of instantiation* (Armstrong, 1989, p.75). This principle states that, in order to exist, universals must have instances:

(22). $\mathbf{\forall x\ Universal(x) \rightarrow \Diamond \exists y\ y::x}$

Much of what has been discussed in this section is summarized in figure 6.6 below.

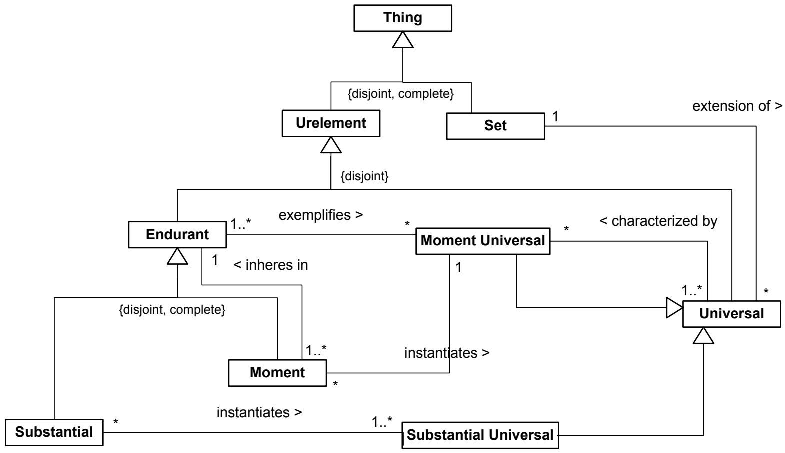


Figure 6-6 Substantials, Moments, Substantial Universals, Moment Universals and their interelations

6.2.6 Qualities, Qualia, Quality Dimensions and Quality Domains

An attempt to model the relation between moments and their representation in human cognitive structures is presented in the theory of *conceptual spaces* developed by the Swedish philosopher and cognitive scientist Peter Gardenfors (Gardenfors, 2000; 2004). The theory is based on the notion of *quality dimension*. The idea is that for several perceivable or conceivable moment universals there is an associated quality dimension in human cognition. For example, height and mass are associated with one-dimensional structures with a zero point isomorphic to the half-line of nonnegative numbers. Other moments such as color and taste are represented by several dimensions. For instance, taste can be represented as a tetrahedron space comprising the dimensions of saline, sweet, bitter and sour, and color can be represented in terms of the dimensions of hue, saturation and brightness. Here, we do not distinguish between physical (e.g., color, height, weight, shape) and nominal moments (e.g., social security number, the economic value of an asset).

According to Gardenfors, some of the quality dimensions (especially those related to perceptual qualities) seem to be innate or developed very early in life. For instance, the sensory moments of color and pitch are strongly connected with the neurophysiology of their perception. Other dimensions are introduced by science or human conventions. For example, the representation of Newton's distinction between mass and weight is not given by the senses but has to be learned by adopting the conceptual space of Newtonian mechanics. Another example is the cognitive dimension of time, which is deemed culture-dependent (while in western cultures we perceive time as a one-dimension linear structure, other cultures perceive time as circular or recurrent). Whereas scientific or conventional dimensions are prescribed by a community of users or defined by a scientific theory, phenomenal quality dimensions have to be empirically extracted via analysis of subject's judgement of similarity between different stimuli (Wilson & Keil, 1999; Shepard, 1962a,b).

As discussed in section 4.3, there is empirical support in the cognitive psychology literature for the thesis that some dimensions are innate and have been hard-wired in the brain for evolutionary reasons. Examples include are our tri-dimensional perception of space and our perception of spatio-temporal continuity. These dimensions are used in our first applied principles of individuation, persistence and identity. On the solid ground provided by these innate dimensions, we expand our conceptual space by including different dimensions associated with the perception of other moments. In fact, Gardenfors explicitly considers learning as the activity of expanding one's conceptual space into new quality dimensions (ibid., p.28) and Quine makes note that some innate quality dimensions are needed in order for learning to be possible (Quine, 1969, p.123).

Gardenfors distinguishes between *integral* and *separable* quality dimensions: "certain quality dimensions are integral in the sense that one cannot assign an object a value on one dimension without giving it a value on the other. For example, an object cannot be given a hue without giving it a brightness value… Dimensions that are not integral are said to be separable, as for example the size and hue dimensions." (Gardenfors, 2000, p.24). He then defines a *quality domain* as "a set of integral dimensions that are separable from all other dimensions (Gardenfors, 2000, p.26)" and a *conceptual space* as "collection of one or more domains" (ibid.). Finally, he defends that the notion of conceptual space should be understood literally, i.e., quality domains are endowed with certain geometrical structures (topological or ordering structures) that constrain the relations between its constituting dimensions. In his framework, the perception or conception of a moment individual can be represented as a point in a quality domain. In accordance with DOLCE (Masolo et al., 2003a) and (Goodman, 1951), this point is named here a *quale*.

We adopt in this work the term *quality structures* to refer to quality dimensions and quality domains. Additionally, we use the term *quality universals* for those moment universals that are associated with a quality structure. Finally, we name *quality* a moment individual that instantiates a quality universal. In the sequel we define these notions formally. We use the predicates QS, QDom and QDim for quality structure, quality dimension and quality domain, respectively.

(23). $\forall x\ \mathbf{QS(x)} \leftrightarrow \mathbf{QDom(x)} \vee \mathbf{QDim(x)}$

Definition 6.9 (Quality, Quality Universal and the association relation): We define the formal relation of association (*assoc*) between a quality structure and an intrinsic moment universal. A quality universal is an intrinsic moment universal that is associated with a quality structure, i.e.,

(24). qualityUniversal(U) $=_{def}$ intrinsicMomentUniversal(U) $\wedge$ $\exists!$x QS(x) $\wedge$ assoc(x,U)

Qualities are the instances of quality universals:

(25). quality(x) $=_{def}$ $\exists!$U qualityUniversal(U) $\wedge$ (x::U)

Finally, quality structures are always associated with a unique quality universal

(26). QS(x) $=_{def}$ $\exists!$U qualityUniversal(U) $\wedge$ assoc(x,U)

■

An example of a quality domain is the domain of tone with the dimensions of pitch and loudness. Another example is the set of integral dimensions related to color perception, which is further explored in the sequel.

A color quality *c* of an apple *a* takes it value in a three-dimensional color domain constituted of the dimensions hue, saturation and brightness. Figure 6.7 depicts the geometric space spawned by the three quality dimensions that form this domain. The *hue* dimension stands for a term whose meaning is close to that of the word color in everyday life. It is a characteristic of the co-called *chromatic* colors (e.g., red and blue) but not of the achromatic colors (i.e., black, white and the totality of neutral greys in between). In figure 6.7, this quality dimension is represented as a circle and the polar coordinates describing the angle of the color around the circle gives the hue value.

Saturation (or Chromaticness) refers to the "purity" of the color, that is, the degree to which it is chromatic instead of achromatic. The more grey (or black and white) is mixed with a color, the less saturation it has. In figure 6.7, this quality dimension has a representation that is homomorphic to the real line ranging from grey (zero color intensity) to increasingly greater intensities. Figure 6.6 depicts the color circle view of the dimensions of hue and saturation together.

Brightness is a measure that varies both among the chromatic and achromatic colors, although it stands out most clearly in the latter. In figure

6.7, brightness is represented as linear quality dimensions with two end points, namely white and black. In other words, White is hueless and maximally bright; Black is hueless and minimally bright (Gleitman, 1991).

Some points in figures 6.7 and 6.8 are worth mentioning regarding the geometry of these representations:

– In figure 6.8 one can easily observe the relation of *complementariness* between colors. Orange can be seen as the overlapping region between Red and Yellow, and Violet as overlapping between Red and Blue. The Red region can therefore be accounted as the sum of the Orange, Violet and (what is termed in color theory) *unique Red*. Likewise, the Green region in the color space is the sum of unique Green, Yellowish green and Blueish green (Fischer Nilsson, 1999). Since Red and Green do not overlap they are said to be *complementary colors*. The same holds for Yellow and Blue. Quality dimensions, quality regions, quality domains and conceptual spaces are one of the best examples of entities related by a type of parthood that obeys the full axiomatization of the Classical Extensional Mereology (see chapter 5). In fact, in DOLCE (Masolo et. al., 2003a), quality regions are defined as mereological sums of qualia, and a *quality space* (a notion analogous to conceptual space) as a mereological sum of quality regions. Finally, from a pragmatics point of view (see section 2.2.3), the visual geometric representation of conceptual spaces represent a case of *pragmatic systematicity* and afford cases of the so-called inferential *free-rides* (Shimojima, 1996);

– The notion of opposition (complement) is derived from the geometrical structure generated by the color (hue) circle (e.g., it is meaningless to talk about opposite weights). In the linear dimensions of saturation and brightness, conversely, an ordering relation can be defined. For instance, the ordering {minimum (black) < low < medium < high < maximal (white)} can be defined for regions of the brightness quality dimension (Fischer Nilsson, 1999). Likewise, the axiomatization of *total orders* can be defined for the dimensions of weight, height, age, etc. In section 6.4.3, we show how the structure of a quality dimension Q can be used to derive characteristics for the so-called *comparative formal relations* that are based on Q;

– The geometric structure of figure 6.7 constrains the relation between some of these dimensions. In particular, saturation and brightness are not totally independent, since the possible variation of saturation decreases as brightness approaches the extreme points of black and white, i.e., for almost black or almost white, there can be very little variation in saturation. A similar constraint could be postulated for the relation between saturation and hue. When saturation is very low, all hues become similarly approximate to grey.

Figure 6-7 The Quality Dimensions of Hue, Saturation and Brightness forming the Color Splinter (Gardenfors, 2000)

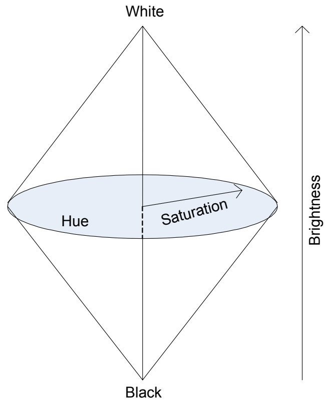
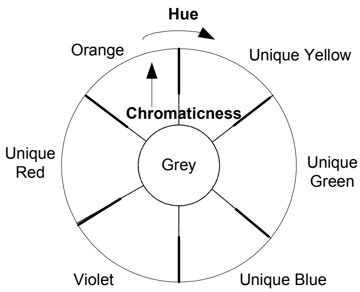


Figure 6-8 The Color Circle view of the dimensions of Hue and Saturation



The trichromatic color quality domain is closely related to the human physiology of color perception. In the animal kingdom, a variety of color systems can be found (Thompson, 1995). For instance, many mammals are dichromats, while others are tetrachromats (e.g., goldfish and turtles). Additionally, for some specific artificial systems, the color domain can be specified in terms of different quality dimensions (e.g., the RGB system) (Raubal, 2004). The important point to be emphasized here is the following: for the same quality universal, there can be potentially many conceptual spaces associated with it, and which one is to be adopted depends on the objectives underlying each a specific conceptualization.

A similar point is made in the GOL ontology (Heller et al., 2004). In GOL, there is an explicit distinction between what is termed a *property* (e.g., size) and what is termed a *property value* (e.g., small, medium, big) of that property. The idea is similar to the one defended here that for a quality universal (GOL-property) there can be a *measurement structure* associated with it. For a given property P of a given individual x there is an *instrument* I that associates P with a “value” in a measurement structure. This value can be a point or a *region* of the measurement structure.

The GOL notion of a measurement structure is akin the notions of quality domain (or quality dimension as a border case of quality domain) as used here. Likewise, the GOL’s property value can be interpreted as our notion of quale. In summary, in conformance with GOL, we propose that for a given quality universal there can be different quality domains associated with different *measurement instruments*. In particular, human perceptual systems can be seen as an example, in which case the quality regions roughly correspond to qualitative sensorial experiences of humans. We also consider social conventions as instruments that are able to generate abstract quality dimensions and domains for nominal quality universals such as address, social security number, license place, etc.

Once more, the notion of *quality region* should be taken here literally, i.e., as spatial regions determined by the geometry or the topology of a conceptual space. For instance, in a one-dimensional quality domain, say the time dimension, the time point *now* divides this dimension into two regions the *past* and *future* regions, which are both one-dimensional regions (lines) of the time domain obeying the same ordering axiomatization of the entire dimension. Conversely, in a bidimensional domain represented in a cartesian space, a region is a sub-area of the domain. Most quality domains are *metric spaces* (Gardenfors, 2000). The concepts of quale, metric space, distance and region are formally defined as follows.

Definition 6.10 (quale): A point in a n-dimensional quality domain can be represented as a vector $v = \langle x_1 \ldots x_n \rangle$ where each x_i represents each of the integral dimensions that constitute the domain. A multidimensional quale is therefore the vector representing the several quality dimensions that are mutually dependent in a quality domain.

■

Definition 6.11 (metric space) (Weisstein, 2004)**:** Let S be a quality domain (set of qualia). A metric space is obtained by associating with S a distance function *d* (called the metric of S) such that, for every two quality values x,y in S, d(x,y) represents the distance between x and y in the domain S. The distance function obeys the following constraints:

a) $d(x,y) = 0$ iff $(x = y)$;
b) $d(x,y) = d(y,x)$;
c) $d(x,y) + d(y,z) \geq d(x,z)$ (triangle inequality)

Generally, the similarity between two vectors x = ‹$x_1 \ldots x_n$› and y = ‹$y_1 \ldots y_n$› in a n-dimensional domain can be computed by the *Minkowskian Metric* (Wilson & Keil, 1999).

$$d(x,y) = \left[\sum_{i=1}^{n} | x_i - y_i |^r \right]^{(1/r)}$$

■

In cognitive structures such as quality domains, the distance function is interpreted as being inversely related to the notion of conceptual *similarity* between two individuals (Wilson & Keil, ibid.). An example of distance function for quality domains (i.e., among integral dimensions) that suitability finds strong empirical evidence in the literature is the *Euclidean Metric* (i.e., for r = 2). In this case,

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \ .$$

■

The definition of a distance function is very important for many reasons, both in theoretical and in practical terms. Given a quality domain and its constituting dimensions, the quality function represents the degree of similarity between two individuals. One can for example give a precise account for why a certain shade of unique red is more similar to orange than to a shade of green, or why the taste of hazelnut is more similar to the taste of walnut than to the one of limes. The other reason is that, as shown by (Gardenfors, 2000), given a number of *prototype* points in a metric quality domain, and a similarity function, a partition of the domain in different quality regions can be generated. Moreover, all generated regions are *convex regions*. The notion of a convex region in a conceptual space can then be used to define the notion of a *quality region*:

Definition 6.12 (quality region): A quality region is a convex region C of a quality domain. A region C is convex iff: for all two points x,y in C, all points between x and y are also in C (Weisstein, 2004).

■

For example, take the color circle generated by the dimensions of hue and saturation in the color domain. By selecting certain color prototypes (for example, empirically) and by applying a distance metric to calculate the similarity between points in this circle, a so-called *Voronoi tessellation* (Weisstein, 2004) is generated, partitioning the circle in the familiar color regions. Moreover, all these regions are indeed convex[49]. Therefore, properties such as Red and Green, which are determinables for the determinate moment universal color, correspond to convex regions in this domain.

It is well understood in philosophical ontology that one should differentiate between the property that things have and the perceptions we have of these properties according to certain measurement instruments employed. Bunge (1977), for instance, uses the term *substantial property* for the ontological entity and the term *attribute* (or predicate) for its logical or linguistic counterpart, and emphasizes the lack of direct correspondence between the two groups of entities. On one hand, there can be properties of things that do not find representation on current measurement systems. On the other hand, predicates can be freely created in conceptual systems and in language that do not represent any *substantial property*. Examples include negation of predicates or predicates formed by logical disjunction, e.g., while one can use the predicate ¬horse(x), there is nothing that has the negative property of *not-being-a-horse*. Likewise, while one can use a predicate equivalent to (car(x) ∨ plane(x)), there is nothing that has the optional property of *being-a-plane-or-a-car*.

According to (Gardenfors, ibid.), only attributes representing substantial properties will form quality regions in a conceptual space. Borrowing from David Lewis' terminology (1986), we name these attributes *natural attributions*, as opposed to *abundant attributions*. This distinction, which has been unfortunately obliterated in terminological logics-based languages (e.g., OWL), is of great conceptual importance. Abundant attributions are, from a cognitive point of view, very poor sources of inductive knowledge (see section 5.2.3 on a similar argument about mereological sums).

The notion of quality dimensions presented here does not require the dimensions to be dense sets. An example of a discrete quality dimension is a collection of qualia that form a graph (see figure 6.9). Still in this dimension, it is possible to define the distance function as the shortest path between two elements in the graph[50]. Consequently, we still have a measurement of similarity between elements, and we can still generate convex quality regions. Finally, there can be quality dimensions that are not metric (e.g., enumerations). In fact, typically, for the same quality domain,

---

[49]In the case of color circle the relevant notion of line is not one of a familiar Euclidian straight line. This is because the distance metric used (since it is a circle) is a polar distance function not a Euclidean one. For this reason, the lines in this metric space are actually arcs from an Euclidian perspective.

[50]Also in this case, the notion of "line" is one of a path in the graph. In general, line in a conceptual space is a metaphor whose Euclidean appearance depends on the nature of the metric used.

many different conceptual modeling representations can be generated. For instance, from the color spline of figure 6.7, the following representations could be derived (among others):

1. an unstructured enumeration containing only the lexical representatives of color quality regions;
2. a UML datatype with attributes x,y and z (whereas x is a polar coordinate for hue and y and z belong to an interval of real numbers), together with a set of constraints on the possible values that the triplets ‹x,y,z› can assume;
3. a concept lattice in which each node represents a quality region and the arcs represent the topological relations between these regions (e.g., similarity and complementariness).

In other words, depending on the perspective and also on the need of accuracy for a given representation, for a given quality moment m, its corresponding quale can be represented in conceptual models, for example, as a vector of quale values (e.g., a particular shade of red with specific hue, saturation and brighness components) or a quality region (e.g., the red region in the color spline).

In (Fischer Nilsson, 1999), the author proposes a *conceptual space logic* (an extended concept lattice formalism) as a way to represent conceptual spaces that is conformant with the traditional knowledge representation paradigm and that, hence, can be used for deriving datatype specifications in languages such as OWL and RDF. A hue lattice using Nilsson's formalism is depicted in figure 6.9. By combining this lattice with others representing the saturation and brightness dimensions, other formal definitions can be made. For instance, the *pink* region can be defined as the intersection between the red region in the hue dimension with the *high* region in the brightness dimension (pink = red × saturation(weak + strong) × brightness(high)[51]). *Shocking pink* can be defined as red × saturation(strong) × brightness(high) and the region of *warm colors* as (red + yellow + yellowishgreen) × saturation(weak + strong). From the lattice it can be automatically inferred, for instance, that green and red, and blue and yellow are pairwise complementary (green × red = blue × yellow = ⊥), and that "shocking pink is pink" (shocking pink < pink).

---

[51] The operators of + and × correspond, in an extensional perspective, to the set-theoretic operators ∪ and ∩, respectively. The symbols ⊤ and ⊥ are named the *Top* and *Bottom* elements in Lattice theory, respectively. The binary operation γ(X) named the *Peirce Product* can be understood as the restriction of ⊤ to those elements that are related via γ to an element in X. For details, one should refer to (Fischer Nilsson, 1999).

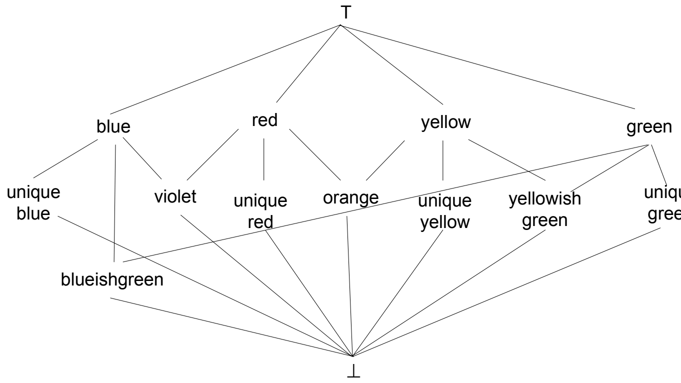


Figure 6-9 A Concept Lattice representing Quality Regions in the Hue dimension and some relations among them

From a metaphysical point of view, quality dimensions and the relations between them, as well as quality domains are *"theoretical entities that can be used to explain and predict various empirical phenomena concerning concept formation"* (Gardenfors, 2000, p.31), i.e., abstract theoretical entities. As a tool for ontological analysis, they provide an interesting *conceptualist*[52] interpretation for W. E. Johnson's notions of *determinable* and *determinate* (Johnson, 1921): every quality region in a quality domain represents a determinable for all its subregions, which in turn, represent its determinates. Since determination is a relative notion (e.g., scarlet is a determinate of red, which is a determinate of color) we should define the ultimate determinable (super-determinable) and the ultimate determinate (super-determinate) in the determination chain (Funkhouser, 2004). In the theory presented here, super-determinables and super-determinates are represented by quality domains and its member qualia, respectively.

From a modeling perspective, the position defended here is that the notion of a quality domain (and the constraints relating different quality dimensions captured in its structure) can provide a sound basis for the conceptual modeling representations of the corresponding quality universal, constraining the possible values that its attributes can assume. This point is further discussed and illustrated in section 6.4.2. In the remainder of this chapter, in order to conform to the representation tradition in conceptual modeling languages, we also take qualia to be abstract entities and represent quality dimensions as *sets of qualia*. Moreover, according to definition 6.11, we take quality domains to be defined in terms of the cross-product of their constituent quality dimensions, and stipulate that the formation rule for the tuples that are members of a quality domain must obey the constraints that relate its quality dimensions. For instance, the *mass* domain can be represented as a subset of the set of Real numbers

[52]In a nutshell, *conceptualism* is the doctrine that equates universals to conceptual categories in cognition. For details, one should refer to (Cocchiarella, 1986; Gardenfors, 2000).

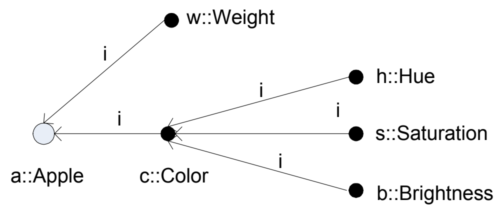
(respecting the same axiomatization) and and the *hue* dimension can be represented as an enumeration of color qualia augmented with a set of formal relations between its members (e.g. *complementaryOf* and *closeTo*). Finally, the color domain can be defined as ColorDomain ⊂ HueDimension × SaturationDimension × BrightnessDimension.

Thus, in this work, quality structures are non-empty sets:

(27). $\forall x\ QS(x) \rightarrow Set(x) \wedge (x \neq \emptyset)$

Following (Masolo et al., 2003a), we take that whenever a quality universal U is related to a quality domain D, then for every individual quality x::U there are *indirect qualities* inhering in x for every quality dimension associated with D. For instance, for every particular quality c instance of Color there are quality individuals h, s, b which are instances of quality universals Hue, Saturation and Brightness, respectively, and that inhere in c. The qualities h, s, b are named *indirect qualities* of c's bearer. This idea is illustrated in figure 6.10 below.

Figure 6-10 Example of an inherence chain from (indirect) quality to quality to substantial



Qualities such as h, s, b and w in figure 6.10 are named *simple qualities*, i.e., qualities which do not bear other qualities. A quality such as c in this figure, in contrast, is named a *complex quality*. The quality universals instantiated by simple and complex qualities are named *simple quality universals* and *complex quality universals*. Formally,

Definition 6.13 (Simple and Complex qualities, and simple and complex quality universals):

(28). $simpleQuality(x) =_{def} quality(x) \wedge \neg\exists y\ i(y,x)$

(29). $complexQuality(x) =_{def} quality(x) \wedge \neg simpleQuality(x)$

(30). $simpleQualityUniversal(U) =_{def} qualityUniversal(U) \wedge (\forall x\ (x::U) \rightarrow simpleQuality(x))$

(31). $complexQualityUniversal(U) =_{def} qualityUniversal(U) \wedge (\forall x\ (x::U) \rightarrow complexQuality(x))$

■

Since the qualities of a complex quality x::X correspond to the quality dimensions of the quality domain associated with X, then we have that no two distinct qualities inhering a complex quality can be of the same type

(32). $\forall x,y,z,Y,Z$ **complexQuality(x)** $\wedge$ **(y::Y)** $\wedge$ **i(y,x)** $\wedge$ **(z::Z)** $\wedge$ **i(z,x)** $\wedge$ **(Y = Z)** $\rightarrow$ **(y = z)**

For the same reason, since there are not multidimensional quality dimensions, we have that complex qualities can only bear simple qualities:

(33). $\forall x$ **complexQuality(x)** $\rightarrow$ **(**$\forall y$ **i(y,x)** $\rightarrow$ **simpleQuality(y))**

Moreover, we have that simple quality universals are always associated with quality dimensions and vice-versa, i.e.,
(34). $\forall x,y$ **assoc(x,y)** $\rightarrow$ **(QDim(x)** $\leftrightarrow$ **simpleQualityUniversal(y))**

and that complex quality universals are always associated with quality domains

(35). $\forall x,y$ **assoc(x,y)** $\rightarrow$
(QDom(x) $\leftrightarrow$ **ComplexQualityUniversal(y))**

Suppose that D is quality domain associated with a (complex) quality universal U, then D can be defined in terms of the cross-product of the quality dimensions associated with the quality universals characterizing U. Formally,

(36). $\forall U,x\ \mathrm{QDom}(x) \wedge \mathrm{assoc}(x,U) \rightarrow \exists y_1 \ldots y_n \exists z_1 \ldots z_n\ (x \subseteq y_1 \times \ldots \times y_n)$
$\bigwedge_{i \leq n} (\mathrm{assoc}(y_i,z_i) \wedge \mathrm{characterization}(U,z_i)) \wedge$
$(\forall w\ \mathrm{characterization}(U,w) \rightarrow \bigvee_{i \leq n} (w = z_i))$

We use predicate *ql(x,y)* to represent the formal relation between a quality individual y and its quale x:

(37). $\forall x,y$ **ql(x,y)** $\rightarrow$ **quale(x)** $\wedge$ **quality(y)**
(38). $\forall x,y$ **ql(x,y)** $\rightarrow \exists$**U (y::U)** $\wedge \exists$**D assoc(D,U)** $\wedge$ **(x** $\in$ **D)**

Relation *ql* is total functional for qualities

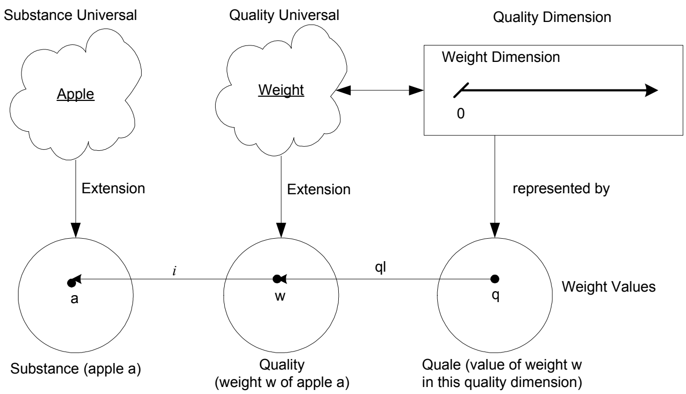
(39). $\forall x$ **quality(x)** $\rightarrow \exists! y$ **ql(y,x)**

Finally, we require every quale to be a member of a unique quality structure:

(40). quale(x) ↔ ∃!y QS(y) ∧ (x ∈ y)

We summarize the discussion of this section in the following way: suppose we have two distinct particular substantials *a* (a red apple) and *b* (a red car), and two qualities q_1 (particular color of *a*) and q_2 (particular color of *b*). When saying that *a* and *b* have the same color, we mean that their individual color qualities q_1 and q_2 are (numerically) different, however, they can both be mapped to the same point in the color quality domain, i.e., they have the same quale. The relations between a substantial, one of its qualities and the associated quale are summarized in figure 6.11.

Figure 6-11
Substantials, Qualities and Qualia



Quality universals can, in principle, form subsumption taxonomies as much as their substantials counterparts. This is very well reflected in Johnson's notions of *derterminable* and *determinate*. As we mentioned before, determinates can be understood as restrictions of determinables, providing a higher degree of specificity. For instance, *being colored* is a determinable and *being red* is a determinate from it. However, *being red* can also be considered a determinable for *being scarlet* (one of its determinates). In pace with (Heller et al., 2004), we choose to specify *quality universals* here only at the super-determinable level. Moreover, as with *GOL-Properties*, the *determinates* of our moment universals depend on a particular measurement systems and its associated conceptual space. For instance, an individual color quality is an instance of the (super-determinable) universal *Color*. In a given quality domain HSBColorDomain, the relation *ql(c,q)* holds between a quality q and a determinate c (a specific shade of a color in

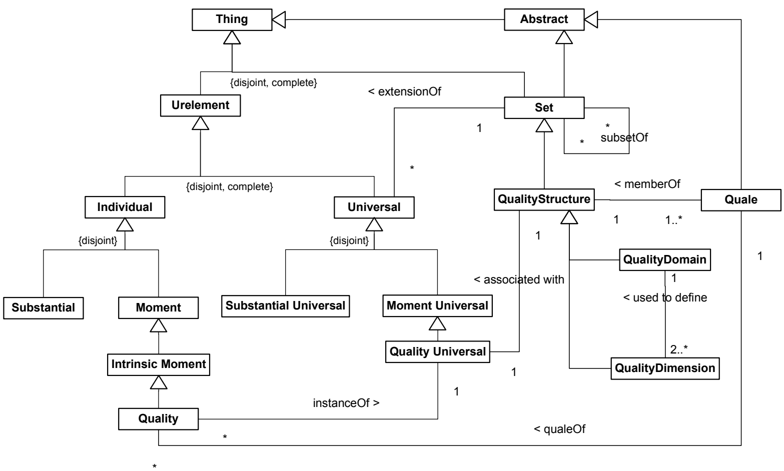
HSBColorDomain), meaning that the quale c is the color value of q in that quality domain.

This is far from disregarding the existence of genuine taxonomies of quality universals. We take here a *realist* stance on universals, in the sense that we believe that universals exist in reality independently of our capacity to know about them. However, we also believe that humans, as cognitive subjects, grasp universals by means of concepts that are in their minds and sometimes cannot capture the universals completely, but only as approximate views. Thus, we defend that, for the purpose of conceptual modeling of things, we must take explicitly into account the conceptual spaces that we use to build the concepts of universals employed in our representations.

Approximating the determinates that we have in our conceptual spaces to the real taxonomy of determinates related to a quality kind is the very task of science discovery, learning and conceptual development (Gardenfors, 2000). In summary, although we believe that every quality instantiates many quality universals related in a determinable/determinate taxonomy, what we mean by a *quality universal* here is only the super-determinate universal in that taxonomy. Therefore, here we consider that for each quality individual there is one and exactly one quality universal that it instantiates.

Figure 6.12 summarizes some important points that we have discussed in this section.

Figure 6-12 Qualities, Quality Universals and their associated Quality Dimensions, and Quality Dimensions as (non-empty) sets of Qualia



6.2.7 Relations and Relators

Relations are entities that glue together other entities. Every relation has a number of relata as arguments, which are connected or related by it. The number of a relation's arguments is called its arity. As much as an unary property such as being Red, properties of higher arities such as *being married-to*, *being heavier-than* are universals, since they can be predicated of a multitude of individuals. Relations can be classified according to the types of their relata.

There are relations between sets, between individuals, and between universals, but there are also cross-categorical relations, for example, between urelements and sets or between sets and universals.

We divide relations into two broad categories, called material and formal relations. *Formal relations* hold between two or more entities directly without any further intervening individual. Examples of formal relations are: 5 is *greater than* 3, this day is *part of* this month, and N is *subset of* Q, but also the relations of *instantiation* (::), *inherence* (i), *quale of a quality* (ql), *association* (assoc), *existential dependence* (ed), among others.

In principle, the category of formal relations includes those relations that form the mathematical superstructure of our framework. We name these relations here *basic formal relations* (Heller et al., 2004) or *internal relations* (Moore, 1919-1920). In this case, in conformance with (Armstrong, 1997; Schneider, 2002) we deem the tie (or nexus) between the relata as non-analyzable. However, we also classify as formal those domain relations that exhibit similar characteristics, i.e., those relations of *comparison* such as is taller than, is older than, knows more greek than. We name these relations *comparative formal relations*. As pointed out in (Mulligan & Smith, 1984), the entities that are immediate relata of such relations are not substantials but qualities.

For instance, the relation heavier-than between two atoms is a formal relation that holds directly as soon as the relata (atoms) are given. The truth-value of a predicate representing this relation depends solely on the atomic number (a quality) of each atom and the material content of heavier-than is as it were distributed between the two relata. Moreover, to quote Mulligan and Smith, "once the distribution has been effected, the two relata are seen to fall apart, in such a way that they no longer have anything specifically to do with each other but can serve equally as terms in a potentially infinite number of comparisons".

Material relations, conversely, have material structure on their own and include examples such as employments, kisses, enrollments, flight connections and commitments. The relata of a material relation are mediated by individuals that are called *relators*. Relators are individuals with the power of connecting entities; a flight connection, for example, founds a relator that connects airports, an enrollment is a relator that connects a student with an educational institution. The notion of relator (relational moment) is supported by several works in the philosophical literature (Heller et Herre, 2004; Loebe, 2003; Degen et al., 2001; Mulligan & Smith, 1986; Mulligan & Simons & Smith, 1984; Bacon, 1995; Simons, 1995; Smith & Mulligan, 1983) and, the position advocated here is that they play an important role in answering questions of the sort: what does it mean to say that John is married to Mary? Why is it true to say that Bill works for Company X but not for Company Y? To quote (once more) Mulligan & Smith's article "The relata of real material relations such as hittings and kissings, in contrast, cannot be made to fall apart in this way: Erna's hitting, r, is a hitting of Hans; it is not a hitting of anyone and everyone who happens to play a role as patient of a hitting qualitatively identical with r. Hence the relational core of such relations cannot be shown to be merely formal".

The distinction between formal and material relations in the quotes from Mulligan and Smith makes it analogous to another distinction among relations, namely the one between bonding and non-bonding relations as proposed by (Bunge, 1977)[53]. For Bunge, bonding relations are the ones that alter the history of the involved relata. For example, the individual histories of John and Mary are different because of the relation "John Kisses Mary". The same is not true for the relation "John is taller than Mary". One can have a world (as history) in which Mary does not exist or in which Mary is taller than John and both individual's history are exactly the same.

Perhaps a stronger and more general way to characterize the difference between formal and material relations is based on their foundation. However, before we proceed, there are some important notions that must be defined. We start with the notion of a *mode*.

Modes are intrinsic moments that are not directly related to quality structures. Gardenfors makes the following distinction between what he calls *concepts* and *properties* (Gardenfors, 2004, p.23): "Properties…form as special case of concepts. I define this distinction by saying that a *property* is based on *single domain*, while a *concept* may be based on *several domains*". We claim that only the moment universals that are conceptualized w.r.t. a single domain, i.e., quality universals, correspond to properties in Gardenfors sense. However, there are moments that as much as substantial universals can be conceptualized in terms of multiple separable quality dimensions. Examples include beliefs, desires, intentions, perceptions, symptoms, skills,

---

[53]The analogy between formal and material relations and bonding and non-bonding ones is only warranted for the case of relations between individuals. Abstract entities have no spatiotemporal qualities and, consequently, no history. However, here we only consider domain relations between particulars.

among many others. We term these entities *modes*. Like substantials, modes can bear other moments, and each of these moments can be qualities referring to separable quality dimensions. However, since they are moments, differently from substantials, modes inhere necessarily in some bearer. We, thus, define modes as follows:

Definition 6.14 (mode): A mode is an intrinsic moment individual which is not a quality.

(41). mode(x) $=_{def}$ intrinsicMoment(x) ∧ ¬quality(x)

■

A special type of mode that is of interest here is the so-called *externally dependent modes*. Externally dependent modes are individual modes that inhere in a single individual but that are existentially dependent on (possibly a multitude of) other individuals that are independent of their bearers. Take, for instance, *being the father of*. This is an example of a universal property, since it is clearly multiple instantiated. Suppose that John is the father of Paul. According to our view of universals, in this case, there is a particular instance x of *being the father of*, which bears relations of existential dependence to both John and Paul. However, x is not equally dependent on the two individuals, since x is a moment it must inhere in some individual, in this case, John. Of course, we can also imagine that under these circumstances, there is another extrinsic moment, instance of *being the son of* which conversely inheres in Paul but is also existentially dependent on John. Formally we have,

Definition 6.15 (Externally Dependent Mode): A mode x is externally dependent iff it is existentially dependent of an individual which is independent of its bearer. Formally,

(42). ExtDepMode(x) $=_{def}$ Mode(x) ∧ ∃y indep(y,β(x)) ∧ ed(x,y)

■

The *indep* relation defined in (13) is symmetric. The intention of (42) is to capture circumstances in which an object has a moment in virtue of its association to an external entity. By using the relation *indep* in (42) we can exclude several problematic cases, such as: (i) the moment *x* being dependent on its bearer; (ii) *x* being dependent on other moments of its bearer (e.g., color of an object can be considered dependent on the object's spatial extension); (iii) *x* being dependent on the essential parts or essential wholes of its bearer.

In the case of a material externally dependent moment *x* there is an individual *external* to its bearer (i.e., which is not one of its parts or intrinsic moments), which is the foundation of *x*. The notion of foundation can be seen as a type of *historical dependence* (Ferrario & Oltramari, 2004), in the way that, for instance, *being connected to* is founded in an individual *connection*, or *being kissed* is founded on individual *kiss, being punched by* is founded in an individual *punch*, *working at* is founded in a working contract. Since founding individuals are typically perdurants, we refrain from elaborating on the notion of foundation here.

Suppose that John is married to Mary. There are many externally dependent modes of John that depend on the existence of Mary, and that have the same foundation (e.g., a wedding event or a social contract between the parts). These are, for example, all responsibilities that John acquires by virtue of this foundation. Now, we can define an individual that bears all externally dependent modes of John that share the same dependencies and the same foundation. We term this particular a *qua individual* (Masolo et al., 2004, 2005; Odell & Bock, 1998). Qua individuals are, thus, treated here as a special type of *complex externally dependent modes*. Intuitively, a qua individual is "the way an object participates in a certain relation" (Loebe, 2003), and the name comes from considering an individual only w.r.t. certain aspects (e.g., John qua student; Mary qua musician) (Masolo et al., 2004).

The notion of qua individuals is ancient and comes at least from Aristotle[54]. For example in *On Interpretation* he says that someone might be good *qua* cobbler without being good. This problem, that was called by medieval philosophers the problem of *reduplicatio* or the problem of *qualification* (Poli, 1998), was faced also by Leibniz when he formulated the identity principle known as "Leibniz law" (see chapter 4), and by Brentano (Angelelli, 1967). More recently, in contemporary philosophy qua individuals have been employed in order to solve some problems related to theories of *constitution* (Fine, 1982) and theories of *action* (see Anscombe, 1979 and Fine, ibid.). Furthermore, they have been used in the literature to ascribe incompatible properties to the same individual x that result from x's participation in different relations. This is illustrated by the classical example *"Nixon qua quaker is a pacifist, while Nixon qua republican is not"* (Masolo et al., 2005). In an ontology that countenances the notion of qua individuals, this situation can be modelled by having two different modes *Nixon-qua-quaker* and *Nixon-qua-republican* inhering in the substantial Nixon. In this case, we take that whilst the former exemplifies *pacifism*, the latter does not.

---

[54] See for example (Szabó, 2003) and, for a more historical and deep account (Baek, 1982).

Our notion of *qua individual* is akin to what is termed a *Material Role* in (Heller et al., 2004) and (Loebe, 2003). Moreover, this notion resembles strongly the one of *roles* in (Wieringa & de Jonge & Spruit, 1995), with the difference that Wieringa and collegues do not discuss this type of external dependence of a role. We resume this discussion on roles in section 7.3, in which we elaborate on the relation between the concepts of role proposed by these authors and the one that we have proposed in chapter 4 as well as their interrelationships.

Finally, we can define an aggregate[55] of all qua individuals that share the same foundation, and name this individual a *relator*. Now, let x, y and z be three distinct individuals such that: (a) x is a relator; (b) y is a qua individual and y is part of x; (c) y inheres in z. In this case, we say that x *mediates* z, symbolized by *m*. Formally, we have that:

(43). $\forall x,y\ \mathbf{m(x,y)} \rightarrow \mathbf{relator(x)} \wedge \mathbf{Endurant(y)}$
(44). $\forall x\ \mathbf{Relator(x)} \rightarrow$
$\forall y\ (\mathbf{m(x,y)} \leftrightarrow (\exists z\ \mathbf{quaIndividual(z)} \wedge (z < x) \wedge i(z,y)))$

Additionally, we require that a relator mediates at least two distinct individuals, i.e.,

(45). $\forall x\ \mathbf{Relator(x)} \rightarrow \exists y,z\ (y \neq z \wedge \mathbf{m(x,y)} \wedge \mathbf{m(x,z)})$

In conformance with (Degen et al., 2001), a relator is considered here as a special type of *moment*. Thus, formally, according to definition 6.1, a relator must inhere in a unique individual, i.e., it must have a bearer (see formula 10). We therefore estipulate that the bearer of a relator r is the mereological sum[56] of the individuals that r mediates, i.e.,

(46). $\forall x\ \mathbf{Relator(x)} \rightarrow (\beta(x) = \sigma y\, \mathbf{m(x,y)})$

Finally, a relation of *mediation* analogous to that of characterization can be defined to hold between relator universals and those universals of their mediated entities:

---

[55] Relators are genuine integral wholes according to the criteria posed in section 5.4, since the relation of common foundation can be used as a suitable characterizing relation. A similar argument can be made in favour of the bearers of relators (see formula 46) in terms of the relation of common mediation.

[56] The notion of mereological sum is defined in chapter 5. In a nutshell, $\sigma x \varphi$ represents the individual x composed by all individuals that satisfy the predicate φ. In the case of formula (46), the bearer of a relator x is the individual y which is composed by all entities mediated by x.

Definition 6.16 (Mediation): The mediation relation holds between a universal U and a relator universal U_R iff every instance of U is mediated by an instance of U_R. Formally,

(47). $$\mathbf{mediation(U, U_R) =_{def} Universal(U) \land RelatorUniversal(U_R) \land \forall x\ (x::U \rightarrow \exists r\ r::U_R \land m(r,x))}$$

■

A relator universal is a universal whose instances are relators. Relator universals constitute the basis for defining material relations R whose instances are n-tuples of entities. In general, a material relation R can be defined by the following schema.

Definition 6.17 (Material and Formal Relations): Let $\phi(a_1,\ldots,a_n)$ denote a condition on the individuals $a_1,\ldots,a_n$

(48). $$\mathbf{[a_1\ldots a_n]::R(U_1\ldots U_n) \leftrightarrow \bigwedge_{i \leq n} a_i::U_i \land \phi\ (a_1\ldots a_n)}$$

A relation is called *material* if there is a relator universal U_R such that the condition ϕ is obtained from U_R as follows:

$$\mathbf{\phi(a_1\ldots a_n) \leftrightarrow \exists k\ (k::U_R \land_{i \leq n} m(k,a_i))).}$$

In this case, we say that the relation R is derived from the relator universal U_R, or symbolically, *derivation(R,* U_R*)*. Otherwise, if such a relator universal U_R does not exists, R is termed a formal relation.

■

An example of a ternary material relation is *purchFrom* corresponding to a relator universal *Purchase* whose instances are individual purchases. These individual purchases connect three individuals: a *person*, say *John*, an individual *good*, e.g. the book *Speech Acts by Searle*, and a *shop*, say *Amazon*. Thus,

[John, SpeechActsBySearle, Amazon]::$\mathbf{R_{purchFrom}}$**(Person, Good, Shop).**

Since *John::Person, SpeechActsBySearle::Good*, *Amazon::Shop*, and there is a specific purchase relator *p::Purchase* such that

$$\mathbf{m(p,John) \land m(p, SpeechActsBySearle) \land m(p,Amazon)}$$

We obtain the following definition for the triple *[a_1, a_2, a_3]* being a link of the relation universal *purchFrom* between *Person*, *Good* and *Shop*:

$[a_1,a_2,a_3]::R_{purchFrom}$(Person, Good, Shop) ↔ John::Person ∧ SpeechActsBySearle::Good ∧ Amazon::Shop ∧ ∃p (p::Purchase ∧ m(p,John) ∧ m(p, SpeechActsBySearle) ∧ m(p,Amazon)).

We can summarize this section as follows:

– we make a fundamental distinction between formal and material relations. Whilst the former hold directly between two entities without any further intervening individual, the latter are induced by mediating entities called relators. Moreover, material relations are founded by material entities in reality, typically perdurants, which are external to their relata. Comparative formal relations, in contrast, are founded in qualities which are intrinsic to the their relata and, hence, can be reduced to relations between these qualities;

– Relators are special types of (relational) moments, i.e., particularized relational properties. Relators are composed of certain externally dependent modes named qua individuals;

– Qua individuals are (potentially complex) externally dependent modes exemplifying all the properties that an individual has in the scope of a certain material relation;

Formal relations shall be treated here as classes of tuples, but which are defined intensionally, as opposed to extensionally. These entities and their interrelationships are exemplified in figure 6.13.

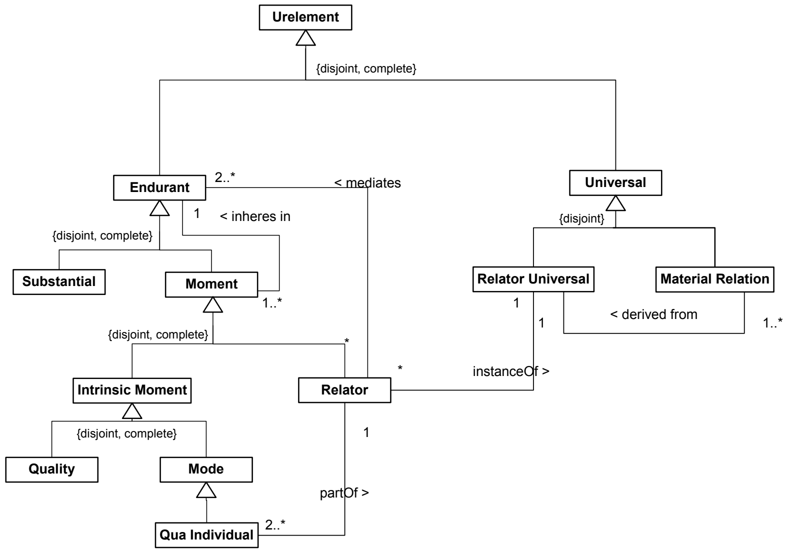


Figure 6-13 Modes, Qua Individuals and Material Relations

6.3 An Ontological Foundation for Conceptual Modeling most Basic Concepts

In this section, we employ the set of ontological categories proposed is section 6.2 to analyze and provide a foundation for some of the most basic constructs in conceptual modeling, namely, *classes*, *objects*, *attributes*, *attribute values* and *associations*. These modeling concepts are represented in practically all conceptual modeling languages. Thus, the conclusions drawn in what follows can be extended to all these languages. However, with the sole purpose of exemplification, we shall refer in the sequel to these concepts as they are represented by UML's modeling primitives. In the remaining of this section, we refer to the *OMG UML Superstructure Specification 2.0* (OMG, 2003c) when quoting text in italics. For simplicity, we write *CM-ontology* when we mean domain ontology in the form of a *conceptual model specification*. Whenever the context is clear, we omit the prefix 'UML' and simply say 'object', 'class', etc., instead of 'UML object', 'UML class', etc.

6.3.1 Classes and Objects

In a UML-based conceptual model specification, an *object* represents a particular instance of a class. It has identity and attribute values (OMG, 2001). A *"Class describes a set of Objects that share the same specification of features,* *constraints and semantics."* (p.86). *"The purpose of a class is to specify a classification of objects and to specify the features that characterize the structure and behaviour of those objects"*. (p.87). Moreover, a UML specification is concerned with describing the *intention* of a class, that is, the rules that define it. The run-time execution provides its *extension*, that is, its instances (OMG, 2001).

We may observe a direct correspondence between universals and classes of a certain kind, as stated in the following principle:

Principle 6.1: In a *CM-ontology*, any universal U of the domain that carries a determinate principle of identity for its instances may be represented as a concrete class C_U. Conversely, for all concrete classes (of a *CM-ontology*) whose instances are basic objects or links (representing individuals), there must be a corresponding universal in the domain.

There are some important points that are represented in this principle. Firstly, it states that only universals that carry a determinate principle of identity for their instances can be represented in a conceptual model as a concrete class (a class that can have direct instances). Therefore, as argued in chapter 4, non-sortal substantial universals must be represented as *abstract classes*. It is a general requirement in conceptual modeling languages that the represented instances must have a definite identity (Borgida, 1990). For this reason, in the category of substantials, what is meant by *object* in conceptual modeling coincides with our use of the term in figure 6.5, i.e., non-object substantials (amounts of matter) can only be represented in a conceptual model as *quantities* (see section 5.5.1).

Moreover, this principle represents an important divergence between the view proposed here and the BWW approach (see, for example, Wand & Storey & Weber, 1999). The proponents of the BWW approach claim that classes in a conceptual model of the domain should only be used to represent substantial universals. In particular, they deny that moment universals should be represented as classes. In our opinion, this claim is not only counterintuitive but also controversial from a metaphysical point of view. This is discussed in depth in section 6.5.1.

Most classes in a *CM-ontology* indeed represent *Substantial Universals*. Firstly, because conceptual models are typically used to model static aspects of a domain and, consequently, the universals represented are typically endurant universals. In addition, Substantials are prior to Moments not only from an existential but also from an identification point of view. For example, (Schneider, 2003b) claims that moments (tropes) are *identificationally dependent* on substantials (objects), i.e., while the latter can be "single out on their own", in order to identify a moment *m* of substantial *s*, one has to identify *s* first.

Principle 6.1, although important to establish the correspondence between conceptual modeling classes and universals, is not elaborated enough to account for the necessary distinctions in this category. For this reason, we defend that the class construct in conceptual modeling should be extended to account for all distinctions among the categories of *substantial* and *moment universals* proposed in chapter 4 of this thesis as well as in the remainder of this section.

6.3.2 Attributes and Data Types

Suppose that we have the situation illustrated in figure 6.11, i.e., a substantial universal *Apple* whose elementary specification contains the feature *Weight*. Thus, for an instance *a* of Apple there is an instance *w* of the quality universal Weight inhering in *a.* The intention of this universal could be represented by the following quality specification:

$$\forall a\ (a{::}Apple \rightarrow \exists w\ (w{::}Weight \land i(w,a)))$$

Associated with the quality universal Weight we have a quality dimension **WeightValue** and, hence, for every instance *w* of Weight there is a quale *c* denoting a particular weight value, i.e., a point in the weight quality dimension such that *ql(c,w)* holds.

We take the weight quality domain to be a one-dimensional structure isomorphic to the half-line of non-negative numbers, which can be represented by the set **WeightValue**. We assume that any quality domain could be represented as an Algebra A = (V,F), where V is the set of qualia and F a finite set of functions. However, we will not discuss the specifics of algebraic specifications and even less shall we discuss particular algebras. Firstly, because algebraic specification is a topic that has been extensively explored in the literature (see for example Goguen & Malcolm, 2000). Secondly, because here we are only interested in the ontological and cognitive foundations underlying these algebraic specifications. From a conceptual modeling perspective, knowing what should be represented is prior to particular mathematical forms of representation.

The mapping between a substantial *a* and its weight quale can be represented by the following function

$$\textbf{weight: Ext(Apple)} \rightarrow \textbf{WeightValue}$$

such that

$$weight(x) = y \mid \exists z\ z{::}Weight \land i(z,x) \land ql(y,z).$$

In general, let U be a substantial or moment universal and let $Q_1,\ldots,Q_n$ be a number of quality universals. Let E be an elementary specification capturing the intention of universal U:

$$\textbf{(49).}\ \forall x\ (x::U \rightarrow \exists q_1,\ldots,q_n \bigwedge_{i\leq n} (q_i::Q_i \wedge i(q_i,x))).$$

If D_i is a quality domain *associated with* Q_i, we can define the function Q_i**:** **Ext(U) → D_i** (named an *attribute function* for quality universal Q_i) such that for every x::U we have that

$$\textbf{(50).}\ Q_i(x) = y \mid y \in D_i \wedge \exists q::Q_i \wedge i(q,x) \wedge ql(y,q).$$

Let us suppose for now a situation in which every Q_i composing in the elementary specification of a universal U (50) is a simple quality universal i.e., Q_i is associated to a quality dimension. In this simplest case, the quality universals appearing in the elementary specification of *U* can be represented in a *CM-ontology* via their corresponding *attribute functions* and associated *quality dimensions* in the following manner:

Principle 6.2: Every attribute function derived from the elementary specification of the universal U may be represented as an attribute of the class C_U (representation of the universal U) in a *CM-ontology*; every *quality dimension* which is the co-domain of one these functions may be represented as data types of the corresponding attributes in this *CM-ontology*. Finally, relations constraining and informing the geometry of a quality dimension may be represented as constraints in the corresponding data type.

For example, in UML *"a data type is a special kind of classifier, similar to a class, whose instances are values (not objects)… A value does not have an identity, so two occurrences of the same value cannot be differentiated"* (p.95). A direct representation of Apple's elementary specification in UML according to principle 6.2 maps the attribute function **weight: Ext(Apple)→WeightValue** to an attribute weight with data type Weight(Kg)[57] in class Apple (figures 6.14.a-b)**.** In UML, specifically,

[57]For now, we shall assume a standard metric unit for each quality dimension (e.g., kilograms in this case). Different metric units affect the granularity but not the structure of a quality domain. For instance, if we consider Weight(Ton) instead of kilograms, the weight dimension is still isomorphic to the half-line of the positive numbers obeying the same ordering.

navigable end names (figure 6.14.b) are a natural alternative mechanism for representing attribute functions since they are semantically equivalent to attributes (OMG, ibid., p.82).

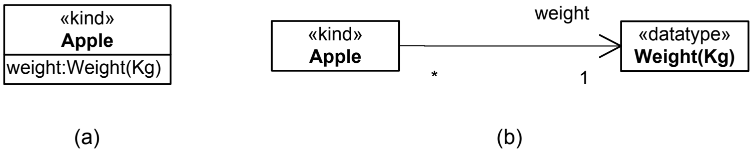


Figure 6-14 Alternative Representations of Attribute Functions in UML

Suppose now that we have the following extension of the elementary specification of the universal Apple represented in figure 6.11:

$$\forall a\ (a::Apple \rightarrow \exists c \exists w\ (c::Color \wedge i(c,a)) \wedge (w::Weight \wedge i(w,a)))$$

In order to model the relation between the quality c (color) and its quale, there are other issues to be considered. As previously mentioned, Color is a complex quality universal, and the quality domain associated with it is a three-dimension splinter (see figures 6.7 and 6.8) composed of quality dimensions hue, saturation and brightness. These dimensions can be considered to be *indirect qualities* universals exemplified in an apple *a*, i.e., there are quality individuals h, s, b which are instances of quality universals Hue, Saturation and Brightness, respectively, that inhere in the color quality c (which in turn inheres in substantial a). The intention of the quality universal Color could then be represented by the following specification:

$$\forall c\ (c::Color \rightarrow \exists h \exists s \exists b\ (h::Hue \wedge i(h,c)) \wedge (s::Saturation \wedge i(s,c)) \wedge (b::Brightness \wedge i(b,c)))$$

In this case, we can derive the following attribute functions from the features in this specification:

hue: Ext(Color) → HueValue;
saturation: Ext(Color) → SaturationValue;
brightness: Ext(Color) → BrightnessValue.
Together these functions map each quality of a color *c* to its corresponding quality dimension. One possibility for modeling this situation is a direct application of principle 6.2 to the Color universal quality specification. In this alternative, depicted in figure 6.15, the class Color directly represents the quality universal color and, its attributes the attribute functions *hue*, *saturation* and *brightness*.

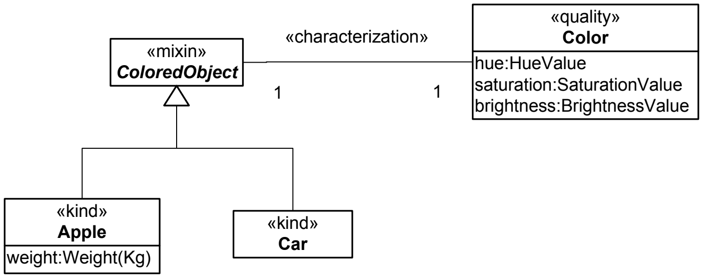


Figure 6-15 Representing Quality Universals and Indirect Qualities

Another modeling alternative is to use directly the construct of a data type to represent a quality domain and its constituent quality dimensions (figure 6.16). That is, we can define the quality domain associated with the universal Color as the set **ColorDomain** $\subset$ **HueValue** $\times$ **SaturationValue** $\times$ **BrightnessValue,** thus complying with (36). Then, we can define the following *attribute function* for the substantial universal Apple according to (50):

color: Ext(Apple) → ColorDomain

such that

$$\mathbf{color(x) = \{\langle h,s,b\rangle \in ColorDomain \mid \exists c::Color\ i(c,x) \land (h = hue(c)) \land (s = saturation(c)) \land (b = brightness(c))\}}$$

where hue, saturantion and brightness are the attribute functions previously defined.

Figure 6-16 Representing Qualia in Multi-Dimensional Quality Domain

In figure 6.16, we use the UML construct of a *structured datatype* to model the **ColorDomain**. In this representation, the *datatype fields* hue, saturation, brightness are placeholders for the coordinates of each of the (integral) quality dimensions forming the color domain. In this way the "instances"(members) of ColorDomain are quale vectors ⟨x,y,z⟩ where x ∈ HueValue, y ∈ SaturationValue and z ∈ BrightnessValue. The *navigable end name **color*** in the association between Apple and ColorValue represents the attribute function ***color*** described above**.**

The two forms of representation exemplified in figures 6.15 and 6.16 do not convey the same information, which we highlight by the use of different stereotypes. In figure 6.15, color objects are one-sidedly existentially dependent on the individuals they are related to via an *inherence* relation. These objects are *bonafide* individuals with a definite numerical identity. As discussed in section 6.2.3, the *characterization* relation between a quality universal and a substantial universal is mapped in the instance level to an inherence relation between the corresponding quality and substantial individuals. In figure 6.16, contrariwise, the members of the ColorDomain are *pure values* that represent points in a quality domain. These values can qualify a number of different objects but they exist independently of them in the sense that a color tuple is a part of quality domain even if no object "has that color".

Both representations are warranted, in the sense that ontologically consistent interpretations can be found in both cases, and which alternative shall be pragmatically more suitable is a matter of empirical investigation. Notwithstanding, we believe that some guidelines could be anticipated. In situations in which the moments of a moment all take their values (qualia) in a single quality domain, the latter alternative (shown in figure 6.16) should be preferred due to its compatibility with the modeling tradition in conceptual modeling and knowledge representation. This is certainly the case with complex quality universals. Additionally, since the conceptualization of these moments depends on the combined appreciation of all their quality dimensions, we claim that they should be mapped in an integral way to a quale vector in the corresponding n-dimensional quality domain.

We can generalize this idea as follows. Let E be an elementary specification capturing the intention of a *substantial or moment* universal U:

$$\forall \mathbf{x}\ (\mathbf{x::U} \rightarrow \exists \mathbf{q}_1,\ldots,\mathbf{q}_n\ \bigwedge_{i\leq n} (\mathbf{q}_i\mathbf{::Q}_i \wedge \boldsymbol{i}(\mathbf{q}_i,\mathbf{x}))).$$

And a quality specification capturing the intention of each *complex quality universal* universal Q_i in E:

$$\forall \mathbf{q}\ (\mathbf{q::Q}_i \rightarrow \exists \mathbf{k}_1,\ldots,\mathbf{k}_n\ \bigwedge_{j\leq n} (\mathbf{k}_j\mathbf{::K}_j \wedge \boldsymbol{i}(\mathbf{k}_j,\mathbf{q}))).$$

Let DO_i be the quality domain associated with Q_i. Thus, $DO_i \subseteq D_1 \times \ldots \times D_n$ such that *assoc*(K_j,D_j) and *characterization*(Q_i,K_j). Then we can define an attribute function

$$\mathbf{Q_i: Ext(U) \rightarrow DO_i}$$

such that

$$\mathbf{Q_i(x) = \langle d_1,\ldots,d_n\rangle \in DO_i \mid \exists q \exists k_1 \ldots k_n\ (q::Q_i) \wedge i(q,x)(\bigwedge_{j \leq n} (i(k_j,q) \wedge ql(d_j,k_j))}.$$

Or simply, if we take that (following formula 50) there are functions $\mathbf{K_j: Ext(Q_i) \rightarrow D_j}$ for all $\mathbf{K_j}$ above, then

$$\mathbf{Q_i(x) = \langle d_1,\ldots,d_n\rangle \in DO_i \mid \exists q\ (q::Q_i) \wedge i(q,x)(\bigwedge_{j \leq n}\ d_j = K_j(q))}$$

Quality domains are composed of integral dimensions. This means that the value of one dimension cannot be represented without representing the values of others. By representing the color quality domain in terms of a structured data type we can reinforce (via its constructor method) that its tuples will always have values for all the integral dimensions. Moreover, the representation of a quality domain should account not only for its quality dimensions but also for the constraints on the relation between them imposed by its structure. To mention another example, consider the Gregorian calendar as a quality domain (composed of the linear quality dimensions days, months and years) in which date qualities can be represented. The value of one dimension constrains the value of the others in a way that, for example, the points [31-April-2004] and [29-February-2003] do not belong to this quality structure. Once more, constraints represented on the constructor method of a data type can be used to restrict the possible tuples that can be instantiated.

In the sequel, we observe the following principle between quality domains and their representation in terms of data types:

Principle 6.3: Every quality dimension D associated to a quality universal Q may be represented as a datatype DT in a *CM-ontology*; Relations constraining and informing the geometry of a quality dimension D may be represented as operators in the corresponding datatype DT. A collection of integral dimension $D_1 \ldots D_n$ (represented by data types $DT_1 \ldots DT_n$) constituting a quality domain QD can be grouped in structured datatype W representing quality domain QD. In this case, every quality dimension D_i of QD may be represented by a field of W of type DT_i. Moreover, the relations between the dimensions D_i of QD may be represented by constraints relating the fields of data type W.

Principle 6.3 is a generalization of principle 6.2 in order to account for quality domains. In summary, every quality universal Q that is associated to a quality domain in an elementary specification of universal U can be represented in a conceptual model via attribute functions mapping instances of U to quale vectors in the n-dimensional domain associated with Q. The n-dimensional domains should be represented in a conceptual model as an n-valued structured data type.

Now, let us consider a case where one of the moment universals M that characterizes a universal U in its elementary specification is a *mode universal*. We defend here that these are the cases in which we want to explicitly represent a moment universal in a conceptual model. An example of such a situation is depicted in Figure 6.17, which models the relation between a Hospital, its Patients, and a number of symptoms reported by these patients. Suppose an individual patient John is suffering from headache and influenza. John's headache and influenza are moments inhering in John. Even if another patient, for example Paul, has a headache that is qualitatively indistinguishable from that of John, John's headache and Paul's headache are two different individuals. Instances of Symptoms can have moments themselves (such as duration and intensity) and can participate in relations of, for example, causation or precedence.

In figure 6.17, the moment universal Symptom is represented by a class construct decorated with the «mode» stereotype. The formal relation between Symptom and Patient is mapped to the *inherence* relation in the instance level, representing the existential dependence of a Symptom on a Patient. In other words, for an instance *s* of Symptom there must be a specific instance *p* of Patient associated with *s*, and in every situation that *s* exists *p* must exist and the inherence relation between the two must hold. This formal relation has a semantics that is outside the usual interpretation of the association construct in UML. According to its standard usage, the multiplicity 1 in the Patient end only demands that, in every situation, symptom *s* must be related to *an* instance of Patient. The inherence relation, however, requires *s* to be always related to the *one and the same instance* of Patient. The difference between these two sorts of requirements is analogous to those marking the difference between mandatory and essential part-whole relations as discussed in chapter 5, respectively.

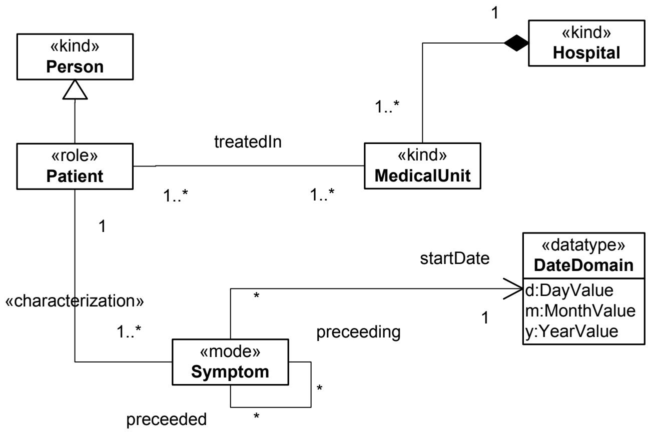


Figure 6-17
Representing Quality Universals and Domain Formal Relations

A mode universal such as Symptom in figure 6.17 can be seen as the ontological counterpart of the concept of *Weak entities types* in EER diagrams, which has been lost in the UML unification process. A *weak entity* can be defined as follows (Vigna, 2004):

"a *weak entity* is an entity that exists only if is related to a set of uniquely determined entities, which are called the *owners* of the *weak entity*. For instance, [the] weak entity type edition; each book has several editions, and certainly it is nonsense to speak about an edition if this does not happen in the context of a specific book…When an *entity* is deleted from a *schema instance*, all *owned weak entities* are deleted, too…For entities of type `W` to be owned by entities of type `X`, a requirement must be satisfied: there must be an *identification function* from `W` to `X` that specifies the *owner* of each *entity* of type `W`, that is, a relationship type going from `W` to `X` whose cardinality constraint impose its instances to be functions. Deletion of an entity of type `X` implies deletion of all related entities of `W`…*Weakness* is recursive. If `X` *owns* `Y` and `Y` *owns* `Z`, then `X` *(indirectly) owns* `Z`…as a special case, it must not happen that an *entity owns* itself".

If we perform suitable substitutions to the italicized terms in the above definition, it becomes directly applicable to the notion of moments adopted here: (i) "a *moment* is an entity that exists only if is related to a set of uniquely determined entities, which are called the *bearers* of the *moment"* (formulas 4 and 8 for the inherence relation). Moreover, the requirement that: (ii) when an *object* is deleted from a *conceptual model*, all *inhering moments* are deleted, too" must be satisfied in implementations of the corresponding model, since moments are existentially dependent on their bearers, they cannot exist without the latter. Also the requirement that (iii) "there must be an *identification function* from `W` to `X` that specifies the *bearer* of each *moment* of type `W`, that is, a relationship type going from `W` to `X` whose cardinality constraint impose its instances to be functions", is clearly satisfied by the inherence relation due to the non-migration principle defined for moments (9). Finally: (iv) "*indirect moment of* is recursive. If `X` *inheres* `Y` and `Y` *inheres* `Z`, then `X` *is an indirect moment of* `Z`....as a special case, it must not happen that a *moment inheres in* itself". Again, inherence is irreflexive (5) and although it is intransitive (7), a relation of *indirect moment of* can be defined to be transitive.

To summarize this section we can provide the following procedure to represent in conceptual modeling the elementary specification of universals and their associated quality universals and quality structures:

Take a substantial universal U with its associated elementary specification. For every moment universal Q characterizing U do:

1. If Q is a simple quality universal then principle 6.2 can be applied;
2. If Q is a complex quality universal then principle 6.3 can be applied;
3. If Q is a mode universal then it should be explicitly represented and should be related to U in a model via a *characterization* relation. Moreover, this procedure can be applied again to the elementary specification of Q.

6.3.3 Associations

In most conceptual modeling languages, n-ary relationships are taken to represent sets of n-tuples (e.g., OWL, LINGO, CCT, EER). In UML, the ER concept of a relationship type is called association. *"An association defines a semantic relationship that can occur between typed instances...An instance of an association is called a link...An association declares that there can be links between instances of the associated types. A link is a tuple with one value for each end of the association, where each value is an instance of the type of the end...An association describes a set of tuples whose values refer to typed instances."*(p.81).

The OMG UML Specification is somehow ambiguous in defining associations. An association is primarily considered to be a 'connection', but, in certain cases (whenever it has 'class-like properties'), an association may be a class: An association class is *"[a] model element that has both association and class properties. An AssociationClass can be seen as an association that also has class properties, or as a class that also has association properties. It not only connects a set of classifiers but also defines a set of features that belong to the relationship itself and not to any of the classifiers."*(p.118).

An association A between the classes $C_1,\ldots,C_n$ of a CM-ontology can, in principle, be understood in our framework as a relation (relational universal) *R* between the corresponding universals $U_1,\ldots,U_n$ whose extension consists of all tuples corresponding to the links of A. However, current conceptual modeling languages (including UML) do not distinguish between formal and material relations.

In figure 6.18, an example of a comparative formal relation is the relation of *precedence* between Symptoms. Precedence is a partial order relation between symptoms that depends only on the starting date of each of them.

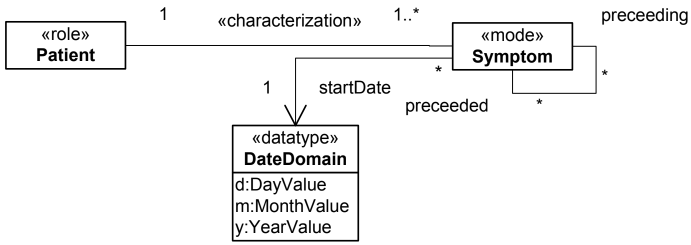


Figure 6-18 Example of a formal Relation between individual modes

It is common in conceptual modeling languages that a number of formal meta-properties (e.g., reflexivity, symmetry, transitivity) are defined for relationships (e.g., LINGO, OWL, RDF, F-LOGIC). In the specific case of *precedence*, these meta-properties are irreflexivity, anti-symmetry and transitivity, i.e., it is a strict *partial ordering* relation. Can we provide an explanation for these meta-properties?

As we have discussed in section 6.2.7, the immediate relata of formal comparative relations are not substantials but qualities. Take, for example, the relations of *taller than*, *heavier than* and *precedence*. All these relations can be reduced to relations between qualities:

- x is *taller than* y iff height(x) $>$ height(y);
- x is *heavier than* y iff weight(x) $>$ weight(y);
- x *preceeds* y iff startDate(x) $<$ startDate(y),

in which *height*, *weight* and *startDate* are attribute functions mapping the substantials x and y to the corresponding qualia. All three quality structures involved in these expressions have a linear structure *ordered* by the $<$

relation. By making this analysis explicit, it becomes evident that *precedence* is a partial order because the qualities founding this relation are associated with a *total ordered* quality dimension. In general, we can state that the meta-properties of a comparative *formal relation* R_F can be derived from the meta-properties of the relations between qualia associated with the qualities founding this relation R_F. This view is also shared by (Gardenfors, 2000). However, unlike Gardenfors, we do not claim that this holds for all types of relations.

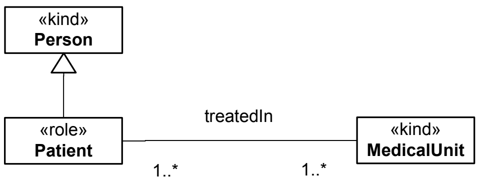
Take for instance the relation treatedIn between Patient and Medical Unit in figure 6.19 below. This relation requires the existence of a third entity, namely an individual Treatment mediating a particular Patient and a particular Medical Unit in order for the relation to hold. This latter case can be modeled in our framework as follows: Let treatedIn be a binary material association induced by a relator universal Treatment whose instances are individual treatments. These individual treatments connect two individuals: a patient, say John, and a MedicalUnit, say TraumaUnit#1. Thus,

$$\textbf{[John, TraumaUnit\#1]:}\mathbf{R_{treatedIn}}\textbf{(Person, MedicalUnit),}$$

since John::Person and TraumaUnit#1::MedicalUnit, and there is a specific treatment t::Treatment such that **(m(t,John) ∧ m(t,TraumaUnit#1)**, We obtain the definition for the tuple [a1,a2] being a link of the association treatedIn between Person and MedicalUnit:

$$\mathbf{[a_1,a_2]:R_{treatedIn}(Person, MedicalUnit) \leftrightarrow a_1::Person \wedge a_2::MedicalUnit \wedge \exists t(t::Treatment \wedge (m(t, a_1) \wedge m(t, a_2))).}$$

Figure 6-19 Example of a Material Relation



How can we represent a material relation in a conceptual modeling language such as UML? Let us follow for now this (tentative) principle: a material relation R_M of the domain may be represented in a CM-ontology by representing the relator universal associated with the relation as an association class. By applying this principle to the treateadIn relation aforementioned we obtain the model of figure 6.20.

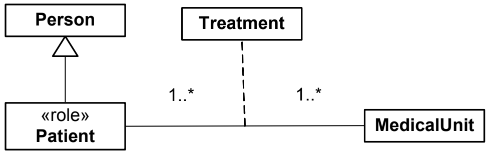


Figure 6-20 Representing a Relator Universal as a UML Association Class

There is a specific practical problem concerning the representation of *material relations* as standard associations that supports a modeling choice in the lines of what is proposed by this principle. This problem, mentioned in (Bock & Odell, 1997a), is caused by the fact that the standard notation collapses two different types of *multiplicity constraints*. For instance, in figure 6.20, the model represents that each Patient can be treated in *one-to-many* Medical Units and that each medical unit can treat *one-to-many* patients. However, this statement is ambiguous since many different interpretations can be given to it, including the following:

- a patient is related to only one treatment to which participate possibly several medical units;
- a patient can be related to several treatments to which only one single medical unit participates;
- a patient can be related to several treatments to which possibly several medical units participate;
- several patients can be related to a treatment to which several medical units participate, and a single patient can be related to several treatments.

The cardinality constraint that indicates how many patients (or medical units) can be related to one instance of Treatment is named *single-tuple* cardinality constraints. *Multiple-tuple* cardinality constraints restrict the number of treatments a patient (or medical unit) can be related to. By modeling the relator universal *Treatment* as an association class one can explicitly represent both types of constraints. A version of figure 6.20 adopting this principle is presented in figure 6.21.

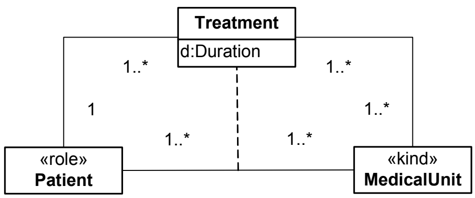


Figure 6-21 Explicit Representation of Single-Tuple and Multiple-Tuple cardinality constraints

This problem is not at all specific to this case. For another example of a situation where this problem arises see figure 6.22. In this case, the (material) relation statement is that: (a) a customer *purchases* one-to-many purchase items from one-to-many suppliers; (b) a supplier supplies one-to-many purchase items to one-to-many customers; (c) a purchase item can be bought by one-to-many customers from one-to-many suppliers.

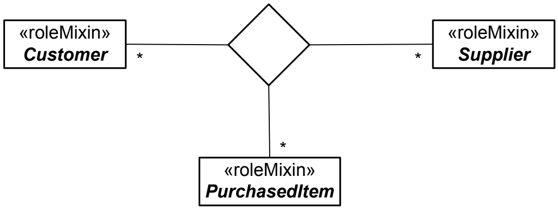


Figure 6-22 Example of a ternary material relation with ambiguous representation of cardinality constraints

PurchaseFrom is a material relation induced by the relator universal *Purchase*, whose instances are individual purchases. Therefore, we have that

$$[a_1, a_2, a_3]::R_{purchFrom}(\text{Customer, PurchaseItem, Supplier}) \leftrightarrow a_1::\text{Customer} \wedge a_2::\text{PurchaseItem} \wedge a_3::\text{Supplier} \wedge \exists p\ (p::\text{Purchase} \wedge \boldsymbol{m}(p,a_1) \wedge \boldsymbol{m}(p, a_2) \wedge \boldsymbol{m}(p, a_3))$$

In other words, for this relation to hold between a particular Customer, a particular PurchaseItem, and a particular Supplier, they must be mediated by the same Purchase instance. Once more, we can see that the specification in figure 6.22 collapses single-tuple and multiple-tuple cardinality constraints. For this reason, there several possible ways of interpreting this model, including the following:

- In a given purchase, a Customer participates by buying many items from many Suppliers and a customer can participate in several purchases;
- In a given purchase, many Customers participate by buying many items from many Suppliers, and a customer can participate in only one purchase;
- In given purchase, a Customer participates by buying many items from a Supplier, and a customer can participate in several purchases;
- In given purchase, many Customers participate by buying many items from a Supplier, and a customer can participate in several purchases;

By depicting the Purchase universal explicitly (such as in figure 6.23), we can make explicit the intended interpretation of the material PurchaseFrom relation, namely, that in a given purchase, a Customer buys many items from a Supplier. Both customer and supplier can participate in several purchases. Although a purchase can include several items, each item in this model is a unique exemplar and, hence, can only be sold once.

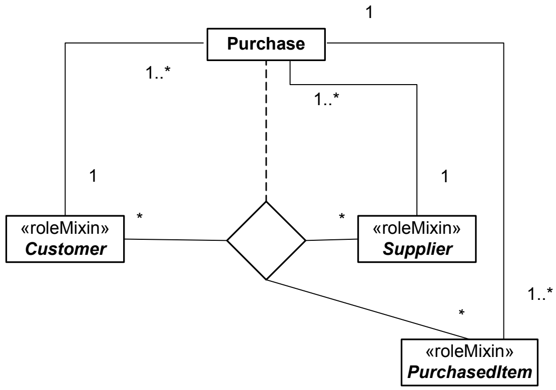


Figure 6-23 Example of a Material Relation with explicit representation of Single-Tuple and Multiple-Tuple cardinality constraints

This problem is specific to material relations. Formal relations are represented by sets of tuples, i.e., an instance of the relation is itself a tuple with predefined arity. In formal relations, cardinality constraints are always unambiguously interpreted as being *multiple-tuple*, since there is no point in specifying single-tuple cardinality constraints for a relation with predefined arity. Hence, formal relations can be suitably represented as standard UML associations. One should notice that the relations between Patient and Treatment, and Medical Unit and Treatment are formal relations between universals (*mediation*). This is important to block the infinite regress that arises if material relations were required to relate these entities. The same holds for the pairwise associations between Customer, Supplier and PurchaseItem, on one hand, and Purchase on the other.

At first sight, it seems to be satisfactory to represent a material relation by using an association class to model a relator universal that induces this relation. Nonetheless, the interpretation of this construct in the language is quite ambiguous w.r.t. defining what exactly counts as instances of an association class. We claim that the association class construct in UML exemplifies a case of *construct overload* in the technical sense discussed in chapter 2. This is to say that there are two distinct ontological concepts that are represented by this construct. To support this claim, we make use of the following (overloaded) semantic definition of the term as proposed by the pUML community: "an associaton class can have as instances either (a) a n-tuple of entities which classifiers are endpoints of the association; (b) a n+1-tuple containing the entities which classifiers are endpoints of the association plus an instance of the objectified association itself" (Breu et al., 1997).

Take as an illustration the association depicted in figure 6.21. In case (a), *TreatedIn* can be directly interpreted as a *relational universal*, whose instances are pairs *[a,b]*, whereas *a* is patient and *b* is medical unit. In this case, *[a,b]* is an instance of TreatedIn iff there is a relator Treatment connecting *a* and *b*. In interpretation (b), *TreatedIn* is what is named in (Guizzardi & Herre & Wagner, 2002b; Guizzardi & Wagner & Herre, 2004) a *Factual Universal*. In short, if the relator r connects (mediates) the entities $a_1,\ldots,a_n$ then this yields a new individual that is denoted by $\langle r: a_1,\ldots,a_n\rangle$. Individuals of this latter sort are called *material facts*. For every relator universal R there is a set of facts, denoted by *facts*(R), which is defined by the instances of R and their corresponding arguments. We assume the axiom that for every relator universal R there is a factual universal F(R) whose extension equals the set *facts*(R). Therefore, an instance of *TreatedIn* in this case could be the material fact $\langle t_1$: John, $\text{MedUnit}_{\#1}\rangle$, whereas John is a Patient, $\text{MedUnit}_{\#1}$ is a Medical Unit and t_1 is a treatment relator (founded in a treatment process).

The relation between relators, relations and factual universals can be defined as follows. Let R be a relator universal. The factual universal F = F(R) is the basis for the material relation universal Γ(F) whose instances are n-tuples of entities. In general, a relation universal Γ(F) can thus be defined by the following schema. Let $\phi(a_1,\ldots,a_n)$ denote a condition on the individuals $a_1,\ldots,a_n$

$$[a1\ldots an]::\Gamma(F)\ (U1\ldots Un) \leftrightarrow \bigwedge_{i \leq n} ai::Ui \wedge \phi(a1\ldots an)$$

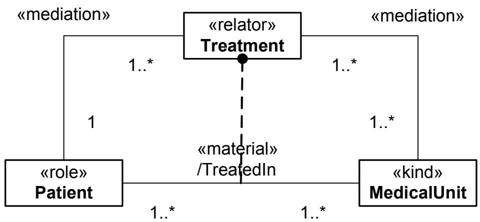
A relation is called *material* if there is a relator universal *R* such that the condition ϕ is obtained from R as follows: $\boldsymbol{\phi(a_1\ldots a_n) \leftrightarrow \exists k\ (k::R \wedge \langle k:a_1\ldots a_n\rangle::F(R))}$.

As a moment, a relator can bear other moments. For example, in figure 6.21, the temporal duration of a Treatment is a moment of the latter. For this reason, between the two aforementioned interpretations for association classes, we claim that interpretation (b) should be favored, since it allows for the explicit representation of relators and their properties. However, there is still one problem with this representation in UML. Suppose that treatment t_1 mediates the individuals John, and the medical units $\text{MedUnit}_{\#1}$ and $\text{MedUnit}_{\#2}$. In this case, we have as instances of Treatment both facts $\langle t_1$: John, $\text{MedUnit}_{\#1}\rangle$ and $\langle t_1$: John, $\text{MedUnit}_{\#2}\rangle$. However, this cannot be represented in such a manner in UML. In UML, t_1 is supposed to function as an object identifier for a unique tuple. Thus, if the fact $\langle t_1$: John, $\text{MedUnit}_{\#1}\rangle$ holds then $\langle t_1$: John, $\text{MedUnit}_{\#2}\rangle$ does not, or alternatively, John and $\text{MedUnit}_{\#2}$ must be mediated by another relator. These are, nonetheless, unsatisfactory solutions, since it is the very same relator Treatment that connects one patient to a number of different medical units.

We therefore propose to represent relator universals explicitly as in figure 6.24. This model explicitly distinguishes the two entities: relator universals are represented by the stereotyped class «relator»; material relations are represented by a derived UML association stereotyped as «material». The dashed line between a material relation and a relator universal represents that the former is derived from the latter (see *derived from* relation figure 6.13). To mark this difference to the similar graphic symbol in UML, we attach a black circle in the relator universal end of this relation. In this figure, a particular Treatment is existentially dependent on a single Patient and in a (immutable) group of medical units. This would mean in UML that for every association representing an existential dependency relation between a moment and the endurant(s) it depends on, the association end should be frozen in the side of the latter. This compound modeling construct should replace the ambiguous association class construct in UML.

Unlike in figure 6.21, the entities representing a relator universal (the stereotyped class that takes the place of an association class), and the material relation (the association itself) are distinct entities. In fact, the latter is completely derived from the former (see definition 6.16). For instance, the relator universal Treatment and the material relation TreatedIn represent distinct entities and can possibly have different cardinalities, since the same relator t_1 can connect both the entities in [John, $MedUnit_{\#1}$] and [John, $MedUnit_{\#2}$]. Nonetheless, the cardinality constraints of TreatedIn can be completely deduced from the existential dependency relations between Treatment and the universals whose instances are the relata of TreatedIn, namely, Patient and MedicalUnit.

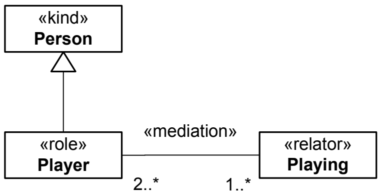
Figure 6-24
Specification with explicit representation of a Relator Universal, a Material Relation, a (formal) derivation relation between the two



This representation eliminates the construct overload present in figure 6.21, since all constructs here have a unique and unambiguous ontological interpretation. Although (derived) relations can be important representation constructs, serving as a base for the specification of *mappings* in design time (Bock & Odell, 1997b), it is important to recognize the conceptual and ontological primacy of their foundations. Comparative formal relations are completely founded on certain intrinsic moments, and material relations are founded on relators. Moreover, as discussed in depth in (Snoeck & Dedene, 1998), the explicit representation of relator universals and their corresponding existential dependency relations provides a suitable mechanism for consistency preservation between static and dynamic conceptual models.

Another benefit of this approach is that relator universals allow for the representation of anadic relations, i.e., relations whose arities are not fixed. Take for example the relation between some people that *play together*. For each instance of *playing* there may be a different number of people that participate in it (Loebe, 2003). This situation can modeled in the approach adopted here by explicitly representing of the anadic relator universal *Playing*.

Figure 6-25 Using Relator Universals to represent Anadic Relations



In figure 6.25, the relator *Playing* connects individuals that play the same role. In (Loebe, 2003), this is named *intensional symmetry* (in contrast to the more common extensional symmetry that holds for formal relations). In general, a material relation R_M is intensionally symmetric iff the relator universal U_R from which R_M is derived is such that: every instance of U_R mediates only entities that instantiate the same role universal. It is important to emphasize that both comparative relations and material relations, as derived entities, have their meta-properties also derived from their foundations. In the case of formal relations, they are derived from the meta-properties of relations among qualia in the underlying conceptual space. In the case of material relations, they are derived from their founding relators and mediated entities. According to formula (45), relators must connect distinct entities. As a consequence, it should be clear that, there cannot be intensionally reflexive binary material relations.

The benefits of this approach are even more evident in the case of n-ary relations with n > 2. Take the UML representation of a ternary relation in figure 6.22. In this specification, we are forced to represent the minimum cardinality of zero for all association ends. As explained in the UML specification (p.82): *"For n-ary associations, the lower multiplicity of an end is typically 0. If the lower multiplicity for an end of an n-ary association of 1 (or more) implies that one link (or more) must exist for every possible combination of values for the other ends"*. As recognized by the UML specification itself, n-ary associations in which there are tuples for every possible combination of the cross-product of the extension of the involved classes are atypically. Thus, in the majority of cases, the UML notation for n-ary associations completely looses the ability of representing real minimum cardinality constraints. Furthermore, as empirically demonstrated in (Bodard et al., 2001), conceptual models without optional properties (minimum cardinality constraints of zero) lead to better performance in problem-solving tasks that require a deeper-level understanding of the represented domain.

Finally, in the same way as qualities, relators can have their own inhering moments (e.g., Duration, as a quality associated to the universal Treatment, in figure 6.21) but also be mediated by other relators such as, for instance, a relator universal *Payment* whose instances connect particular Treatments and Payers.

The results of this section can be summarized in the following principle regarding the representation of formal and material relations in a *CM-ontology*:

Principle 6.4: In a *CM-ontology*, any formal relation universal R_F of the domain may be directly represented as a standard association whose links represent the tuples in the extension of R_F. Conversely, a material relation R_M of the domain may be represented in a *CM-ontology* by a complex construct composed of: (i) *CM-class* stereotyped as «relator» representing the *relator universal*. The relator universal is associated to CM-Classes representing mediated entities via associations stereotyped as «mediation»; (ii) a standard association stereotyped as «material» representing a material relation whose links represent the tuples in the extension of R_M; (iii) a dashed line with a black circle in one of the ends representing the formal relation of derivation between (i) and (ii), in which the black circle lies in the association end of the relator universal.

6.4 Related Work

The approach found in the literature that is closest to the one presented here is the so-called BWW approach presented in (Shanks & Tansley & Weber, 2003; Evermann and Wand, 2001a,b; Wand & Storey & Weber, 1999; Weber, 1997; Wand & Weber, 1995, 1993, 1990, 1989). In these articles, the authors report their results in mapping common constructs of conceptual modeling to an upper level ontology. Their approach is based on the BWW ontology, a framework created by Wand and Weber on the basis of the original metaphysical theory developed by Mario Bunge in (Bunge, 1977, 1979).

In this section we compare the foundation ontology proposed here with BWW in terms of their theories and of their corresponding mapping approaches. Additionally, we discuss the approach proposed in (Veres & Hitchman, 2002; Veres & Mansson, 2005). This approach applies a psychologically and linguistically well-founded ontology as well as empirical evidence to criticize the general assumptions of the BWW approach, and in particular, its proposals for the representation of relational properties.

6.4.1 Things and Substantials

The concepts of *substantial* here and of *thing* in BWW are both based on the Aristotelian idea of substantial, i.e.,

1. an essence which makes a thing what it is;
2. that which remains the same through changes;
3. that which can exist by itself, i.e., which does not need a 'subject' in order to exist.

In BWW, a thing is defined as a substantial individual with all its substantial properties: "a thing is what is the totality of its substantial properties" (Bunge, 1977, p.111). As a consequence, in BWW, there are no bare individuals, i.e., things without properties: a thing has one or more substantial properties, even if we, as cognitive subjects, are not or cannot be aware of them. Humans get in contact with the properties of things exclusively via the thing's attributes, i.e. via a chosen representational view of its properties. This is far from saying that Bunge himself embraces the so-called *Bundle of Universals* perspective. In fact, he explicitly rejects this theory and, instead, holds a position that can be better identified with a *substance-attribute* view (Armstrong, 1989).

In short, in the former type of theory, particulars are taken as *bundles of universals*, i.e., as aggregates of properties which themselves are repeatable abstract entities. The most important exemplar of this type of theory was proposed by Russel in (Russel, 1948). A direct consequence of this theory is that two different things cannot have the same properties, where properties are universals. It is important not to confound this principle with another principle discussed in chapter 4 named the Leibniz's law, also know as the principle of the (P1) *indiscernability of identicals*. Principle (P1) states that if x is identical to y then whatever property x has then y has as well, or formally

$$\forall x,y\ (x = y) \rightarrow \forall U\ (x::U \leftrightarrow y::U)$$

The principle is implied by the bundle theory of universals is named the (P2) *Identity of indiscernibles* and can be formally stated as follows

$$\forall x,y \forall U\ (x::U \leftrightarrow y::U) \rightarrow (x = y)$$

that is, if two individuals instantiate the same universals then they are identical. Whilst principle (P1) is universally accepted among philosophers, (P2) is matter of great controversy, and there is at least the logical possibility that (P2) fails to be the case (Armstrong, 1989).

Another problem with this bundle theory is that it makes universals the substance of reality, in the sense that everything is constructed out of universals. How can be the case that the concrete reality is made solely by these abstract entities? These problems (among others) are discussed in great detail in (Armstrong, 1989, chap.4). In fact, among the theories that countenance the existence of properties, Armstrong considers the bundle theory of universals to be the weakest from a philosophical point of view.

The *substance-attribute* view makes an explicit distinction between a thing and the properties that the thing happens to have. As a consequence, the theory countenances the existence for every individual of a propertyless *substratum, particularized essence* or *bare particular*. The notion of substratum is strongly associated with the British empiricist philosopher John Locke (Armstrong, 1989) and due to its mysterious nature it has been the target of some criticism throughout history. Nonetheless, as a "theoretical fiction" (Bunge, 1977, p.57) it solves some of the philosophical problems existing in the *bundle of universals* theories.

In summary, Bunge is a *realist w.r.t. universals*, i.e., he claims that universals exist in reality independent of our knowledge of their existence. However, he denies the existence of *particularized properties*, i.e., moments. The denial of property instances puts BWW in a singular position among the foundational ontologies developed in the realm of computer science (e.g., Schneider, 2003b, 2002; Heller et al., 2004; Masolo et al., 2003a; Neuhaus & Grenon & Smith, 2004).

In principle, it seems that a thing in BWW could be directly associated to our concept of substance. However, there are some important differences between the two. Whilst a BWW-thing can be thought as a substratum instantiating a number of properties (as repeatable abstract entities), our substantials are particulars that bear other particularized properties (i.e., moments), or to borrow Simons' phrase, "particulars in particular clothing" (Simons, 1994).

Although we do not make any ontological commitment w.r.t. the nature of our *substratum*, by adopting a four-category ontology, if necessary, we can dispense with a substratum of a mysterious nature. In this case, we can take a view such as the one of Simons' *Nuclear Theory*. This theory proposes that the accidental properties instances of an individual are held together by their mutual existential dependency to a nucleus. This nucleus, in turn, is composed by a number of essential properties that form what Husserl names a *foundational set*, i.e., a closure system under the relation of existential dependency. This approach has the advantages of the substance-attribute view, without having to accept its problems, since the nucleus is akin to a substratum, only not a mysterious one. In BWW, contrariwise, the mysterious substratum cannot be eliminated without putting the theory into a *Bundle of Universals* group. We claim that this flexibility puts an ontology in which moments are recognized in a better position than one in which they are not.

6.4.2 Properties and Moments

Despite the differences, an important commonality between a BWW-thing and our notion of substance is that, as the former, the latter has a non-empty appearance, i.e. every substance bears at least one moment. The converse is also true, i.e., in both approaches a property exists only in connection with its bearer. In BWW, a property whose existence depends only on a single thing is called an *intrinsic property*. A property that depends on two or more things is called a *mutual property*. These concepts are analogous to our notions of *intrinsic* and *relational moment universals*. Nevertheless, once more, in our approach properties are instantiated. Thus, our intrinsic properties can be better defined as universals whose instances inhere in a single individual, while relational properties are universals whose instances mediate multiple individuals.

In BWW, only things possess properties. As a consequence, a property cannot have properties. This dictum leads to the following modeling principle: "Associations should not be modeled as classes" (Rule 7 in Evermann & Wand, 2001b). Here, in contrast, according to principle 6.1 we propose that any first-order universal can be represented as a conceptual modeling class. This issue marks an important divergence between the view proposed here and the BWW approach.

The BWW authors claim that classes in a conceptual model of a domain should only be used to represent substantial universals. In particular, they deny that universals whose instances are particularized properties (i.e., moments) should be represented as classes. This claim is not only perceived as counterintuitive by conceptual modeling practitioners (as shown by Veres & Hitchman, 2002; Hitchman, 2003; Veres & Mansson, 2005), but it is also controversial from a metaphysical point of view.

Bunge denies that there properties of properties, i.e., higher-order properties. Since classes can have attributes representing properties, (Wand & Storey & Weber, 1999; Weber, 1997) claim that properties should not be represented as classes, since in this case they could have attributes ascribed to them. We think there are some problems with this argumentation.

First, claiming that there are no higher-order universals is itself quite controversial. (Armstrong, 1989), for instance, who embraces scientific realism as a theory of universals, claims that higher-order properties are *necessary* to represent the concept of a *law*. For Armstrong, a law such as Newton's F = MA describes a second-order relation between the three universals involved. Strangely enough, Bunge also defines the concept of a *Law* (quite a central notion is his approach) as a relation between properties, which then makes it a second-order relation (Bunge, 1977, p.77). The view that there are, in fact, material higher-order universals is also shared by other approaches (e.g., Degen et al., 2001; Heller & Herre, 2004). Even simple higher-order relations between universals such as "*Redness is more like Orange than it is like yellow"* cannot be dealt with in the current version of the BWW framework. As discussed in section 6.2.6, in the approach presented here, if one wants to dispense with higher-order properties, this relation can be expressed in terms of relations between quality regions (abstract individuals) in a conceptual space.

The second problem with Wand & Weber's argumentation is that, even if one denies the existence of higher-order properties, it is not necessary to proscribe the representation of properties as classes. Alternative, one can simply proscribe the representation of attributes in classes representing properties[58].

Nonetheless, if one subscribes to Bunge's theory, there is a much stronger reason to argue against the representation of non-substantial universals as classes. Since Bunge denies the existence of *particularized properties*, one could simply state that properties should not be represented as classes because they should not be allowed to have instances. As discussed in section 6.2.3, there is philosophical, cognitive and linguistic support in the literature for taking an ontological view of individuals in which moments are accepted (Schneider, 2003b, 2002; Heller et al., 2004; Masolo et al., 2003a; Neuhaus & Grenon & Smith, 2004; Lowe, 2001). Moreover, even if both ontological choices were deemed equivalent, there are cognitive and pragmatic reasons for defending the acceptance of property instances and, hence, in favor of accepting also the representation of non-substantial universals as conceptual modeling types. As demonstrated in section 6.3.3, the explicit representation of relator universals (relational properties) allows for the explicit representation of

[58]Since the authors take attributes to represent substantial properties, we assume here that the restriction is meant only for this type of attributes. That merely formal higher-order attributes (predicates) can always be created for universals is beyond dispute.

single-tuple and multiple-tuple cardinality constraints in associations. Additionally, as discussed in the same section, it enables the possibility of consistency preservation between static and dynamic conceptual models of a domain. Finally, in our approach, properties of properties such as the *hue of a certain color* or the *graveness of a certain symptom* can be modeled as first-order inherence relations between moments, and this is possible exactly because we countenance particularized properties.

To provide one more example of the importance of relators in conceptual modeling, suppose the situation in which one wants to model that *students* are *enrolled* in *universities* starting in a certain date. Following the proscription of mutual properties being modeled as entity types, (Wand, Storey, and Weber 1999) propose the following model for this situation (figure 6.26)[59].

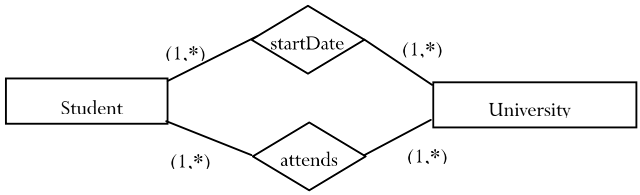


Figure 6-26 An alternative modelling of "properties of properties" according to the BWW approach (from Wand & Storey & Weber, 1999).

We claim that it is rather counterintuitive to think about a model of this situation in these terms. According to Wand, Storey & Weber, relationships representing mutual properties are equivalent to n-ary attribute predicates. In this case, what is startDate supposed to stand for? Is it a binary predicate that holds, for example, for John and the University of Twente, like in *startDate (John, UT)*? This seems to be an absurd conclusion. Thus, startDate should at least be a ternary predicate applied to, for instance, *startDate (John, UT, 14-2-2004).* Now, suppose that there are many predicates like this one relating a student and a university. For example, the *start-date of writing the thesis*, the *start-date of receiving a research grant*, etc. We believe that, in this case, the authors would propose to differentiate the startDate depicted in figure 6.26 by naming it *startDateofEnrollment*. But does not this move make it obvious that *startDate* is actually a property the enrollment? In our approach, this can be explicitly modeled such as in figure 6.26.

[59]The BWW model is shown in the ER notion according to the original models presented in (Wand, Storey, and Weber 1999).

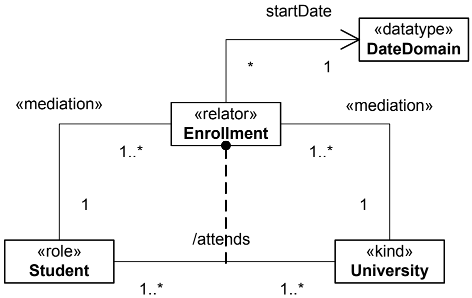


Figure 6-27 The representation of "properties of properties" according to our approach

The model of figure 6.27 makes an explicit distinction between a closed-linked relation between student and university and an indirect relation between student and start date. In what follows we discuss cognitive and linguistic justifications for highlighting this distinction.

6.4.3 Conceptual Structures and a Linguistic Analysis of n-ary relationships

In a series of papers, Veres and colleagues (Veres & Hitchman, 2002; Veres & Mansson, 2005; Hitchman, 2003) offer a detailed analysis and criticism of the general assumptions of the BWW approach. More specifically, in (Veres & Mansson, 2005), they provide empirical evidence to support a case against the BWW treatment of associations.

They argued that claims about the intuitiveness of modeling languages constructed by the principles provided by the BWW approach are based on a hidden assumption about psychology. That is, "the assumption behind the claim that an 'ontologically correct' modeling language will be intuitive is that people's cognition is also 'ontologically correct' at some level: if our models represent reality the way it 'really is', then people will find this a natural representation" (Veres & Mansson, 2005). The criticism cannot be against ontology *per se*, since the authors themselves state that they "describe an ontology of conceptual structure" or "psychologically motivated ontology" for the same purpose, but against the use (for this purpose) of a revisionary ontology (such as Bunge's) that lacks a linguistic and cognitive foundation. Moreover, they observe that the human cognitive architecture imposes fundamental constraints on our representations and that *some* prescriptions on the representation of entities should also be based on psychology

In contrast, the authors propose the use of a psychologically motivated ontology based on the theory of conceptual structures developed by the linguist Ray Jackendoff (Jackendoff 1983, 1990, 1997) as a foundation for guiding conceptual modeling decisions. It is important to emphasize that conceptual structures themselves are not language dependent, and they are not determined by language use. Quite the contrary; language evolved as a way to externalize the pre existing content of cognition for the purpose of communication (Pinker & Bloom, 1990). The idea is that we gain insight into concepts by observing the correspondences between the structural properties of languages, on one side, and conceptual structures on another. This idea is very much conformant with approach developed here (see discussions on chapters 2 to 4 of this thesis).

In (Veres & Mansson, 2005), the authors propose an interesting example that considers the linguistic (syntactic) distinction between *complements* and *adjuncts*. Observe the following sentences:

1. Sarah robs the 7-11 in New York.
2. Adam robs in Washington in February.

The basic premise in both sentences is that someone commits a robbery. However, if one constructs a syntactic tree for these sentences it becomes manifest that *robs* and *7-11* in sentence 1 are more closely linked than are *robs* and *Washington* in sentence 2. *7-11* is a *complement* of the verb *robs*, whereas *in Washington* is a less closely linked *adjunct*. Simple transformations on these sentences can show that a verb is quite selective in the complements it will take, but insensitive to its adjuncts. An interesting point observed by the authors is that this distinction reflects a distinction in the underlying conceptual structure: following a lexical/conceptual correspondence rule from (Jackendoff, 1990) it can be shown that the complement is directly linked to the verb in the conceptual structure. Adjuncts, conversely, are mapped into some lower position in the recursive definition.

Now, suppose a conceptual model containing the entities *Person, Location, Establishment, Time_period.* How should the situations in (1) and (2) be modeled? Both situations describe a ternary relationship between three entities: *person-establishment-location* or *person-location-time_period*. And, as (Wand, Storey, and Weber 1999) suggest, all n-ary relationships should be modeled alike. However, as Veres and colleagues demonstrate, the two scenarios have subtly different underlying conceptual representations.

Figure 6.28 and 6.29 depicts two alternative models of situations (1) according to the BWW approach and to ours, respectively. First of all, the use of a relator universal in our model eliminates the ambiguity caused by the collapse of multiple-tuple and single-tuple cardinality constraints in 6.28. Additionally, since it is not the case that there are tuples for all the combinations of instances of Thief, Establishment and Location, in figure 6.28, the minimum cardinality constraints for all association ends connected to these classes must be specified equal to zero. This problem disappears in figure 6.29.

Notice that the specification of figure 6.28 contradicts the BWW rule that proscribes the representation of optional properties (Weber, 1997). In other words, the BWW model represented in this figure is inconsistent even w.r.t. to the rules prescribed by the BWW approach. Moreover, this will be the case for almost all n-ary relations with n > 2.

Finally, the approach taken in figure 6.29 allows for the explicit representation of the relations between Thief (Person) and Establishment (a material relation mediated by a relator robbery) and the indirect relation between Thief and Location. Notice that *robs* can be hardly said to be a real ternary relation in the first place, since Location and Establishment are strongly coupled. In fact, once the establishment is fixed the location of the robbery is derived from the location of the establishment.

Figure 6-28 The representation of the sentence *Person robs Establishment in Location* according to the BWW approach

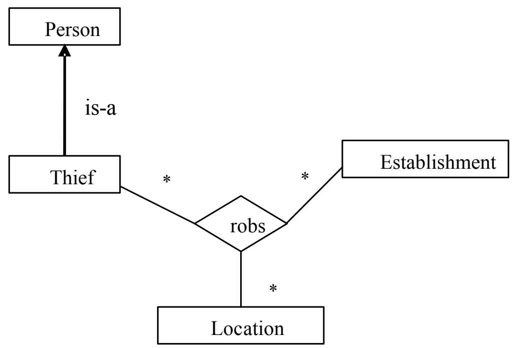
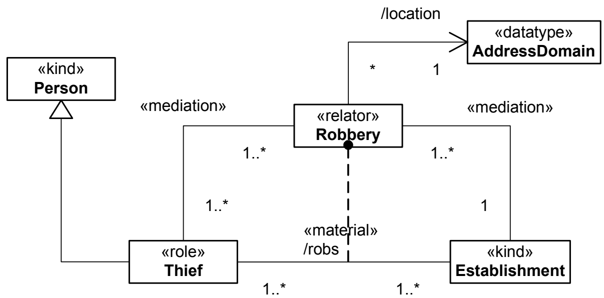


Figure 6-29 The representation of the sentence *Person robs Establishment in Location* according to our approach



Despite what has been discussed so far, the claim that "the practice of representing domain attribute and relationship types as UML classes is *ontologically incorrect*" made by the BWW authors is, in our view, not-warranted. At maximum, one can say that this practice produces conceptual models that are non-conformant with the BWW ontology. However, models obeying this practice would conform to several other philosophically sound foundational ontologies in computer science (e.g., Heller et al., 2004; Masolo et al., 2003; Schneider, 2002; Neuhaus & Grenon & Smith, 2004, Veres & Mansson, 2005). In fact, we believe that the evidence shown here should support not only a case in favor of our ontological choices, but also a case against the suitability of Bunge's ontology as a proper ontological foundation for conceptual modeling.

6.4.4 Natural Kinds and Substantial Universals

In BWW, the definition of a class is based on the notion of the scope of a property. A scope s of a property P is a function assigning to each property that exists in a domain a set of things from that domain, i.e., s(P) is the set of things in the domain that possess property P. A class is then defined as the scope of a property.

If we have a non-empty set P of properties, the intersection of the scopes of all members of P is called a kind. Finally, a kind whose properties satisfy certain *laws* is called a *natural kind*.

A model that describes things with common properties is named a *functional schema.* A functional schema comprises a finite sequence of functions $F = \langle F_1..F_n \rangle$, such that each function F_i (named an *attribute*) represents a property shared by the members of the class described by the functional schema. For every attribute F_i there is a co-domain V_i of values. Evermann and Wand claim that a CM-type cannot be mapped to any of the BWW concepts of class, kind or natural kind, because the latter are defined extensionally (as special sets), while the former is defined intensionally and has an extension at run-time. They, therefore, propose that a CM-type is equivalent to a functional schema of a natural kind.

As we discuss in section 4.5.1, the notion of a natural kind is equivalent to our *substance sortal*, or simply, a *kind*. In that section, we argue that a kind is only one of the many types of substantial universals that are needed for conceptual modeling. Clearly, some of the classifiers that are used in the examples provided by the BWW authors (e.g., student in figure 6.26) are not natural kinds according to this definition.

Evermann and Wand propose that the conceptual modeling representations of Natural Kinds (i.e., functional schemas) should be accompanied of a constraints specification that restricts the possible values that the attributes of a substantial thing can assume. This idea has its origins in Bunge's ontology itself. Bunge defines a function F(t) as the *state function* of the thing, such that $F(t') = \langle F_1(t')..F_n(t') \rangle$ is said to represent the *state of a* *thing* at time t'. The set $V_1 \times \ldots \times V_n$ (i.e., the Cartesian product of all co-domains) is termed the *state space of a thing*. Now, since the properties of instances of a natural kind are lawfully related, it is not the case that the coordinates of state vectors can vary freely. The subset of $V_1 \times \ldots \times V_n$ constrained by the laws of the natural kind being described is named by Bunge the *lawful state space* of a thing. In other words, the lawful state space associated with a natural kind defines all possible states that instances of this kind can assume.

In our approach the *lawful state spaces* are the realist counterpart of conceptual spaces associated with *substance sortals*. As in the BWW, we claim that the conceptual model representation a universal U should be constrained by the laws relating the properties of U, or in a less than ideal case, by the known structure of the conceptual space associated with U. Nonetheless, unlike in BWW, we acknowledge that the value domains V_i can themselves be multidimensional, exhibiting a constrained structure that occurs in the definition of quality domains associated with possibly different substantial universals. For this reason, we believe the explicit representation of quality domains (associated with quality universals) as datatypes not only provides a further degree of structuring on lawful state spaces, but it also allows for a potential reuse of specifications of a subset of its constraints.

6.5 Final Considerations

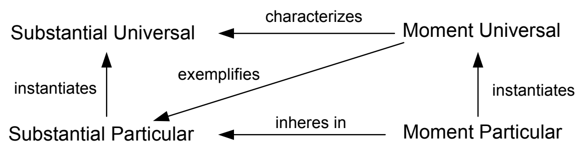
In this chapter, we provide ontological foundations for some of the most basic conceptual modeling constructs, namely, classes, attributes and relationships. These are fundamental constructs in the sense that they are present in practically all conceptual modeling languages (e.g., UML, ER, LINGO, CCT), and in particular, in the so-called *semantic web languages*. However, in these languages, these constructs are understood only on a superficial level, and typically interpreted as sets of elements, or as the extensions of certain modeling *predicates*. According to (Wand & Storey & Weber, 1997) the relationship construct seems to be particularly problematic. Despite of being perceived as very important in practice and widespread in the literature, empirical evidence shows that the use of this construct is often problematical as a way of communicating meaning in an application domain. For example, in studying the process of logical database design, (Batra & Hoffler & Bostrom, 1990, p.137) conclude "that the most commonly occurring errors pertain to the connectivity of relationships."

In pace with Wand and colleagues, we believe that this is mainly due to the lack of consensus and imprecise definitions of the meaning of these categories. As a consequence, "users of conceptual modeling languages are frequently confused about whether to show an association between things via a relationship, an entity, or an attribute." (Wand & Storey & Weber, 1999, p.494). In fact, an inspection in the conceptual modeling literature shows that these constructs receive a mere superficial characterization in most conceptual modeling languages, such as that "associations represent a semantic link between objects" (OMG, 2003c). Thus, in practice, a relationship is simply taken to stand for any n-ary predicate in natural language.

In this chapter, we have shown that taking entity and relationship universals as sets does not provide any explanation for what it means for something to have a certain property (e.g., a certain atomic number), or for multiple things to be related (e.g., John being married to Mary). For this reason, any philosophically and cognitively well-founded ontology must countenance a number of more fundamental urelements that should complement the set-theoretical ontologies underlying existing conceptual modeling languages. To put it in a different way, *Barrichello's Ferrari is red* because it bears its particular redness, and *Peter is kissing Sarah* because there is a particular kiss in which both of them participate.

The foundational ontology developed in this chapter is centred in an Aristotelian ontological square depicted in figure 6.30. Ordinary mesoscopic objects of our every-day experience belong to the category of substantial. Examples are an apple, a cat, my car, Queen Beatrix, the Dutch part of the North Sea, etc. These objects are characterized by certain particular qualities. For instance, an apple *x* has a certain particular color *y*, which is a quality of this apple, and numerically distinct from the particular colors of all other apples. Because *x* bears y, which is an instance of a universal Y, *x* is said to exemplify Y, in the way, for example, that Monica Bellucci is said to exemplify beauty, or that Stephen Hawking is said to exemplify intelligence. Additionally, if every x exemplifies a number of (moment) universals $Y_1 \ldots Y_n$, we say that these universals characterize the (substantial) universal X that x instantiates[60].

Figure 6-30 The Ontological Square



As we have discussed along the chapter, there is strong support in the cognitive and philosophical literature for accepting the existence of particularized properties and particularized relations. Actually, in agreement

[60]This is far from saying that the *meaning* of a substantial universals correspond to an enumeration of properties common to all its instances (see discussion in Keil, 1992).

with (Schneider, 2002), we believe that moments are indeed "the immediate objects of everyday experience". From an ontological point of view, the acceptance of these entities enables the construction of a foundational ontology in which minimum metaphysical commitments are made. In section 6.4.1, we have discussed that it allows for an ontology that can dispense both with the controversial "identity of the indiscernibles" principle, and with a mysterious substratum or bare particular. However, another philosophical advantage can be pointed out in favour of this approach. In section 6.2.5, we are intentionally neutral w.r.t. to the nature of universals. However, in an ontology that accepts moments, there is open the possibility of conceiving universals as special sorts of particulars, such as, for example, *resemblance structures* (Armstrong, 1989; Schneider, 2003b). The advantage of such a view is that, if being a first-order particular is to be part of a resemblance structure then the same can be said for higher-order ones. In fact, in this case, higher-order instantiation could be explained in terms of mereological relations between resemblance structures. For instance, if the first-order universal *Eagle* is thought of as a particular then not only it can bear its own moments (e.g., *life expectancy*), which are not exemplified in any particular eagle, but it easily can be thought of as a instance (part of) the higher-order universal (resemblance structure) *Bird species*. There is support in the philosophy of biology literature for conceptualizing biological species as individuals, or more precisely as integral wholes unified by the characterizing relation of common ancestry (Erenefsky, 2004; Milikan, 1998). Although we shall not entertain this possibility here, it should be investigated in future extensions of our ontological framework. This idea, in principle, could be used to provide an ontological foundation for the important (but only vaguely explained) concept of *Powertype* in conceptual modeling.

One formal relation that plays a paramount role here is *existential dependency*. This simple notion, which has been formally defined in chapter 5 allow us to precisely characterize: (i) the distinction between substantial and moments; (ii) inherence (mediation) relation between moments and substantials. In other words, once we have a formal notion of existential dependence, we can differentiate which particulars in the domain are *substantials* and which are *moments*. Based on the multiplicity of entities a particular depends on, we can distinguish between *intrinsic* and *relational moments*. With qualities we can explain *comparative formal relations*, and with relators, the *material* ones. In particular, by employing (explicitly represented) relators, we can provide not only an ontologically well-founded interpretation for the (otherwise problematic) relationship construct, but also one that can accommodate more subtle linguistic distinctions. Additionally, from a conceptual point of view, we can produce models that are free of cardinality constraint ambiguities. Furthermore, with the notions of modes and relators we can define *qua individuals*. Finally, with moments we can explain exemplification, and with exemplification characterization. In summary, practically the whole core of the urelement segment of our foundational ontology is based on the primitive and unambiguous notion of *existential dependency*.

In this chapter, we have also been intentionally neutral to whether substantials and moments should be interpreted as *continuants* or as *snapshot entities* (i.e., momentary states of continuants)[61]. In the latter case, moment continuants can be thought as logical constructions from world-bounded moment snapshots, in the same way that in L_1 (see section 4.4.1) substantials are considered as logical constructions from world-bounded substantial snapshots. We name these moment counterparts of substantial individual concepts as *moment persistents* (after Heller et al., 2004). This interpretation can provide an interesting explanation for some linguistic cases regarding properties. Take the following example from (Masolo et al., 2003a):

1. This rose is red.
2. Red is a color.
3. This rose has a color.
4. The color of this rose turned to brown in one week.
5. The rose's color is changing.
6. Red is opposite to green and close to brown.

In (1), the predicate red is applied to an individual rose. As an adjective, red is a characterizing universal of the substantial universal rose, i.e., a mixin. In (2), red is intended as a noun, i.e., red is taken as abstract particular, a quality region in a quality domain (another abstract particular). The same hold for (6), in which a particular shade on red can be interpreted as a region of a quality domain that, due to the structure of that domain, bears certain relations to some other regions (brown and green). Sentence (3) can be interpreted as a case of inherence. Specially, when taken together with (4) and (5). In sentences (4) and (5) one is not speaking of a shade of color, neither of a characterizing universals but of something that changes while maintaining its identity. In DOLCE, this is explained by having a quality color that can change with time. This position is at odds with most approaches in the literature that take even super-determinate moment universals to be rigid. In other words, if a particular is an instance of redness, it cannot cease to be so, ergo, it cannot become and instance of brownness. Therefore, in most theories of particularized

[61] See discussion in section 4.4.

properties, these changes are explained by having one particular quality replaced by another. However, in this case, there is no quality there that persists while keeping its identity. Now, if individuals are interpreted as snapshots then we can have:

(i) in world w_1 a moment x_1 instance of color c_1 inhering in r_1 (rose snapshot);
(ii) in world w_2 a moment x_2 instance of color c_2 inhering in r_2 (rose snapshot);
(iii) r_1 and r_2 are states of the same continuant rose because there is an individual concept R (in the extensional of the universal rose) such that $R(w_1) = r_1$ and $R(w_2) = r_2$;
(iv) x_1 and x_2 are states of the same moment continuant *Color* because there is an *moment persistent*[62] C such that $C(w_1) = x_1$ and $C(w_2) = x_2$.

In this case, C is exactly what persists in (4) and (5) while maintaining its identity, and which is capable of changes in a genuine sense.

Finally, besides the concrete urelements that constitute our ontology, we explicitly take into account the conceptual measurement structures in which particular qualities are perceived (and conceived). By employing the theory of conceptual spaces, we can provide a theoretical foundation for the conceptual modeling notion of *attribute values* and *attribute value domains*. Traditionally, in conceptual modeling, value domains are taken for granted. In general, they are considered by taking primitive datatypes as representing familiar mathematical sets (e.g., natural, integer, real, Boolean) and the focus has been almost uniquely on mathematical specification techniques. Once more, we defend that understanding what should be represented is prior to delving on specific ways of specifying it. Whatever constraints should be specified for a datatype must reflect the geometry and topology of the *quality structure* underlying this datatype. Moreover, as discussed in section 6.3.3, by understanding the structure of a certain quality domain, we can derive meta-properties (e.g., reflexivity, asymmetry, transitivity) for the comparative formal relations that are based on this domain.

We emphasize that there are other benefits for conceptual modeling in adopting a notion such as the one of conceptual spaces. For example, in (Gerlst & Pribbenow, 1995), besides their classification of parts as *quantities*, *elements* or *components* (see discussion on section 5.5), the authors

[62] In the case that a substance bears only one moment of each kind, C can be logically constructed by taking in each world the unique individual x_i that instantiates the super-determinable universal Color. We aimed at generality here to account for situations in which this is not the case, such as in the case of qua individuals.

propose an orthogonal criterion of classification, which is independent of the inherent compositional structure of the entity considered as a whole. The authors name a *Portion* as an aggregate constructed by selecting certain parts of an integral object, according to certain internal properties of this object. For example,

a. the red parts of a painting;
b. the exciting parts of a story;
c. the reddish parts of an object;
d. the dark (colored) parts of a picture;
e. the tall animals in the group.

The resulting *portions* are those parts of the whole that provide the requested value of the relevant property. However, some sentences like (c) and (d) make it clear that the construction of portions with respect to a specific property must be rely on the underlying structure in which a certain property takes it values. Sentence (e) actually exemplifies a situation where the use of *contrast classes* in conceptual spaces can allow for interesting cases of non-monotomic reasoning (Gardenfors 2000; 2004). Since, for instance, although every squirrel is an animal, a tall squirrel is not a tall animal.

By explicitly representing conceptual spaces and their constituent domains and dimensions, one can analyze the same object in different measurement structures. In particular, one can define transformations and projections between conceptual spaces that facilitate knowledge sharing and semantic interoperability. Moreover, one can allow for *context-aware reasoning*, by providing context matrices that emphasize certain dimensions in detriment of others in certain circumstances. In (Raubal, 2004), the author demonstrates how conceptual spaces have been used to model façades of buildings as landmarks for a wayfinding service in Vienna. First, different conceptual spaces are associated with the same landmarks for different purposes. For example, in a database system that stores and manipulates information about the landmarks, the color domain assumed is the RGB model, and a *cultural importance* dimension of the landmarks is considered. For the wayfinding service, a color model based on human perception (HSB, see figures 6.7 and 6.8) is used instead, and the cultural importance dimension is not taken into account. Finally, *weight matrices* are used to highlight different dimensions in different contexts. As discussed by Raubal, people select different landmarks for wayfinding during the day and at night. Thus, while in a *day context*, the best landmark to be used by the service could be a façade with the most contrasting color, in a night context, it can be simply the highest or widest landmark.

Chapter 7

Roles

In this chapter, we employ some of the results of our previous chapter to describe the modeling concept of roles.

Roles represent a fundamental notion for our conceptualization of reality. This notion has received much attention both in philosophical investigation (van Leeuwen, 1991; Wiggins, 2001; Loebe, 2003; Masolo et al., 2004, 2005) and in the conceptual modeling literature in topics such as object-oriented modeling (Bock & Odell, 1998), agent-oriented modeling (Odell & Parunak & Fliescher, 2003; Odell & Nodine & Levy, 2005), and organizational modeling (Dignum, 2004), among others. In a comprehensive study on this topic, Friedrich Steimman (2000b) defends that the role concept naturally complements those of *objects* and *relationships*, standing on the same level of importance. However, Steimann also recognizes that *"the role concept, although equally fundamental, has long not received the widespread attention it deserved"*, and that *"although there appears to be a general awareness that roles are an important modelling concept, until now no consensus has been reached as to how roles should be represented or integrated into the established modeling frameworks"* (ibid., p.84). The last statement can be verified by inspecting the diversity and incompatibility of the several conceptualizations of roles currently co-existing in the literature (Wieringa & Spruit & de Jong, 1995; Loebe, 2003; Steimann, 2000a,b, 2001; Bock & Odell, 1998).

Recently, not only has the interest in roles grown continuously, but also has the interest in finding a common ground on which the different notions of roles can be judged and reconciled. In fact, a significant part of the work presented in this chapter originated in a material that we prepared for an invited presentation in a workshop dedicated to discussing different conceptualizations of this concept (see Guizzardi, 2004, but also Masolo et al., 2005).

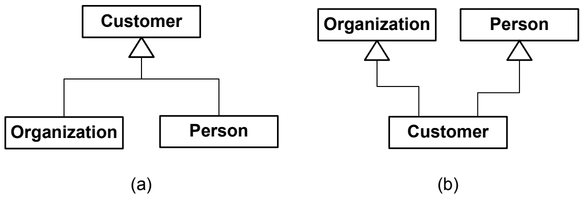
This chapter also exemplifies the usefulness of the categories proposed in our foundational ontology. Firstly, by employing the categories and postulates of the theory of universals constructed in chapter 4, we propose an *ontological design pattern* that captures a solution to a recurrent and much discussed problem in role modeling (section 7.1). Moreover, with some definitions offered in chapter 5, we investigate the link between some of the formal meta-properties defined for part-whole relations and those meta-properties by which roles are characterized (section 7.2). Furthermore, by borrowing some results from chapters 4 and 6, we manage to harmonize some different conceptions of roles that are used in the literature (section 7.3). Finally, by building on an existing theory of transitivity of linguistic functional parthood relations, and on the role related notion of *qua individuals* discussed in section 7.3, we propose a number of visual patterns that can be used as a methodological support for the identification of the scope of transitivity for the most common type of part-whole relations in conceptual modeling (section 7.4).

Section 7.5 concludes the chapter by presenting some final considerations.

7.1 An Ontological Design Pattern for Role Modeling

Figure 7.1 below exemplifies a recurrent problematic case in the literature of role modeling (Steimman, 2001), which is termed here the problem of role with multiple disjoint *allowed types* (Bock & Odell, 1998). Suppose a conceptualization in which the social concept of Customer should be represented. A Customer, in this conceptualization, is assumed to be a role played by an entity in the context of *purchasing* relation to a Supplier. Moreover, in this conceptualization, not only persons but also organizations can be customers. Notice that, in this case, customer is indeed a role according to the definition of chapter 4, i.e., it is anti-rigid and relationally dependent universal. The problem is how to model the relationship between the role type Customer and its allowed types Person and Organization? This problem is also mentioned in (van Belle, 1999): *"how would one model the customer entity conceptually? The Customer as a supertype of Organisation and Person? The Customer as a subtype of Organisation and Person? The Customer as a relationship between or Organisation and (Organization or Person)?"* The first two alternatives are presented in figures 7.1.a and 7.1.b, respectively.

Figure 7-1 Problems with modeling roles with multiple allowed types



In the model of figure 7.1.a, the role *Customer* is defined as a supertype of *Person* and *Organization*. This modeling choice violates postulate 4.3 (see chapter 4) and produces an ontologically incorrect conceptual model. Firstly, not all persons are customers, i.e., it is not the case that the extension of *Person* is necessarily included in the extension of *Customer*. Moreover, an instance of *Person* is not necessarily a *Customer*. Both arguments are equally valid for *Organization*.

In a series of papers (Steimann, 2000a,b, 2001), Steimann discusses the difficulties in specifying admissible types for Roles that can be filled by instances of disjoint types. As a conclusion, the author claims that the solution to this problem lies in the separation of role and type hierarchies, which would lead to a radical and counterintuitive revision to the metamodel of most current conceptual modeling languages, or to put in UML terms, a counterintuitive *heavyweight extension* to the language (OMG, 2003b). In the sequel we show that this claim is not warranted. Moreover, we propose a *design pattern* based on the profile introduced in chapter 4 that can be used as an ontologically correct solution to this recurrent problem. Finally, this solution has a smaller impact to conceptual modeling metamodels than the one proposed by the author, since it does not demand radical heavyweight extensions to the language.

In the example above, *Customer* has in its extension individuals that belong to different kinds and, thus, that obey different principles of identity. Customer is hence a dispersive or transsortal type (a non-sortal) and, by definition, cannot supply a principle of identity for its instances. Since an (determinate) individual must obey one and only one principle of identity (postulates 4.1 and 4.2 in chapter 4), every instance of Customer must be an instance of one of its subtypes (forming a partition) that carry that principle of identity. For example, we can define the sortals *PrivateCustomer* and *CorporateCustomer* as subtypes of *Customer*. These sortals, in turn, carry the (incompatible) principles of identity supplied by the kinds *Person* and *Organization*, respectively. In summary, if x is a *Customer* (abstract class) then x must be an instance of exactly one of its subtypes (e.g., *PrivateCustomer*) that carries the principle of identity supplied by an appropriate substance sortal (e.g., *Person*).

Figure 7.2 shows how this solution can be incorporated in a conceptual modeling design pattern. In this picture the abstract class A is the *role mixin* that covers different role types (e.g., Customer, Participant). Classes B and C are the disjoint subclasses of A that can have direct instances, representing the sortal roles that carry the principles of identity that govern the individuals that fall in their extension. Classes D and E are the ultimate substance sortals (kinds) that supply the principles of identity carried by B and C, respectively. The association r represents the common specialization condition of B and C, which is represented in A. Finally, class F represents a type that A is *relationally dependent* of.

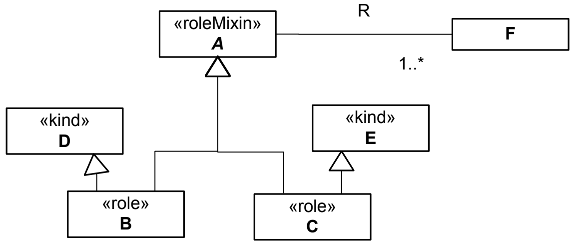


Figure 7-2 An ontological design pattern for the problem of specifying roles with multiple disjoint allowed types

An application of this pattern is illustrated in figure 7.3 in which it is used to produce ontologically correct versions of two stereotypical models of this situation (one of them being the model depicted in figure 7.1). In both cases, the universal the *role mixin* depends of, and the association representing the specialization condition are omitted for the sake of brevity.

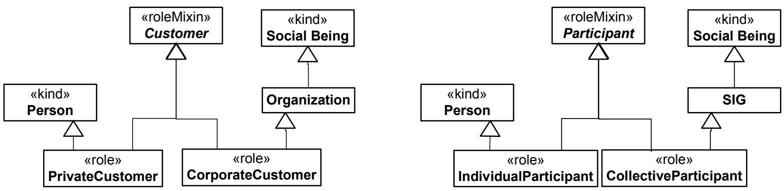


Figure 7-3 Ontologically correct models of roles with disjoint allowed types obtained by the application of the design pattern of figure 7.2

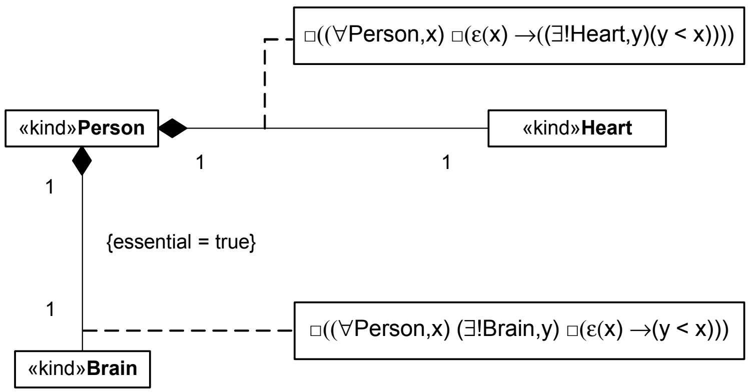
7.2 Parts of Roles

In chapter 5, we have presented a division of parthood relations w.r.t. ontological dependence containing three possible subtypes:

- *essential parts*: characterized by existential dependence from the whole to a part;
- *mandatory parts*: characterized by generic constant dependence from the whole to the universal a part instantiates;
- *contingent parts*.

Let us take the human body as an example. The relation between an individual Human Body and an individual Human Brain is an example of (i). In contrast, the relation between a Human Body and a Human Heart is an example of (ii). In the former case, a Body *x* depends specifically on the existence of a particular Brain y, i.e., for every instance of Human Body *x* there is a unique Brain *y* such that in every world that *x* exists, *y* is part of *x*. In the latter case, contrariwise, the Body *x* depends on the existence of *any* instance of the Heart universal, not on a specific one. In other words, for every Human Body *x* and in every world that *x* exists, there is a Human Heart *y* that, in that world, is part of *x*. These two situations are depicted in figure 7.4 together with their corresponding modal logics formalizations. For the sake of simplicity, we formalize in this case only the axioms w.r.t. the relation from the whole to the part. All other axioms are omitted. Moreover, one should remember that we are dealing only with non-trivial entities here, i.e., we dispense with entities that are necessarily existent or necessarily non-existent.

Figure 7-4
Representation of essential and mandatory parthood in a conceptual specification of the human body



In all examples used in chapter 5, the universals representing wholes are *kind universals* (see chapter 4). Let us now investigate how these different types of necessary parthood relations can be used to characterize other non-rigid universals, such as *Roles*. Suppose, for instance, the situation depicted in figure 7.5. The figure illustrates the relation between a Boxer and one of his hands. What the picture attempts at representing is the statement that "every boxer must have a hand". This relation is certainly not one of mandatory parthood, since it is not the case that a Boxer depends generically on the universal hand but specifically on one particular hand[63]. It thus appears to be the case that this relation is one of essential parthood.

[63] We are here not considering the possibility of hand transplants. Once more, the point of the argumentation is not the specific example.

However, this is not true either. If a hand were to be considered an essential part of a particular boxer then the corresponding formula represented in figure 7.5 should be valid. To show that this is not the case, suppose the following: let John be a boxer in world w and let x be John's hand in w. What the formula in figure 7.5 states is that in every world w' in which John exists, x must be part of John in w'. This formula is clearly falsifiable. One just have to imagine a world w'', in which John exists without being a boxer and without having x as his hand (supposed that x has been tragically amputated in w''). This problem arises from the ambiguity of the word "must" in "every boxer must have a hand". Intuitively, the situation that this model intended to express is the valid statement that "For every Person *x*, there is a hand *y*, such that *in every world that x is a Boxer*, *y* is a hand of *x*".

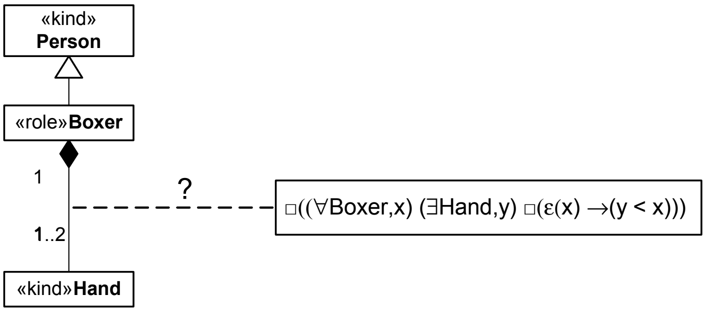


Figure 7-5 Problems in the representation of specifically dependent parts for anti-rigid universals

In the example of figure 7.5, Boxer cannot have essential properties and, in particular, cannot have essential parts, since it is an anti-rigid universal. In other words, if "to be a boxer" is consider as a property, it is not an essential property itself of any individual.

However, this situation can be understood in terms of the philosophical distinction between *de re* and *de dicto* modality. Take the following two sentences:

(i) The queen of the Netherlands is necessarily queen
(ii) The number of planets in the solar system is necessarily odd

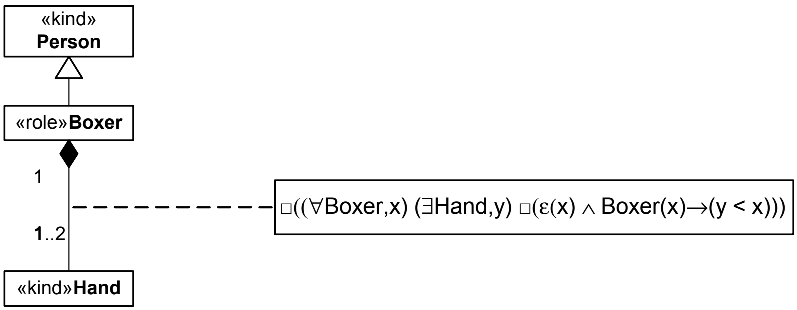
In the *de re* reading, the first sentence expresses that a certain individual (Beatrix) is necessarily queen. This is clearly false, since we can conceive a different world in which Beatrix decides to abdicate the throne. However, in the *de dicto* reading the sentence simply expresses that it is necessarily true that in any circumstance whoever is the Dutch queen is a queen. The second sentence works in the converse manner. In the *de re* reading the sentence (ii) expresses that a certain number (9) is necessarily odd. This is indeed necessarily true. The *de dicto* reading of the sentence however is false. It is not necessarily the case that the number of planets in the solar system is odd. We can imagine a counterfactual situation in which the solar system has, for instance, 8 or 10 planets. The Latin expressions *de re* represents a modality which refers to a property of the thing itself (*res*), whereas de dicto represents a modality that refers to an expression (*dictum*). This is made explicit in the logical rendering of the possible readings of these two expressions:

(iii-a) *de re* (false): $\forall x\ QueenOfTheNetherlands(x) \rightarrow \Box(Queen(x))$
(iii-b) *de dicto* (true): $\Box(\forall x\ QueenOfTheNetherlands(x) \rightarrow Queen(x))$

(iv-a) *de re* (true): $\forall x\ NumberOfPlanets(x) \rightarrow \Box(Odd(x))$
(iv-b) *de dicto* (false): $\Box(\forall x\ NumberOfPlanets(x) \rightarrow Odd(x))$

Take now the expression “every boxer has necessarily a hand”. Once more, this expression is true only in one of the readings, namely, the *de dicto* reading. Whilst it is the case that the expression “In any circumstance, whoever is boxer has at least one hand” is necessarily true, it is false that “If someone is a boxer than he has at least a hand in every possible circumstance”. Figure 7.6, expresses a correct representation of this situation in the *de dicto* modality.

Figure 7-6 Correct representation of specifically dependent parts of anti-rigid universals



We now have expressed three different types of dependency relations between wholes and parts:

– specific dependence with *de re* modality;
– generic dependence with *de re* modality;
– specific dependence with *de dicto* modality.

The remaining option is, of course, conceivable, i.e., generic dependence with *de dicto* modality. This situation can be captured by the following formula (v), in which A represents the (anti-rigid) whole and B represents the part. In this formula, the predicate B is used as what we term here a *guard predicate*. Intuitively, this predicate "selects" those worlds, in which the parthood relation must hold. The same holds for the predicate Boxer in figure 7.6.

(v) $\Box(\forall A,x\ \Box(\varepsilon(x) \wedge A(x) \rightarrow \exists y\ B(y) \wedge (y < x)))$

We have seen that essential properties, i.e., specific dependence expressed in terms of the *de re* modality, can only be expressed for rigid universals. For anti-rigid universals (e.g. roles), only the corresponding *de dicto* modality can be applied. Nonetheless, it is also true that for every *de re* statement regarding an individual *x*, we can express a corresponding *de dicto* one, by using as guard predicate the substance sortal that x instantiates. For instance, if it is true that "The number of planets in the solar system (9) is essentially odd" then it is also true that "In any circumstance, if 9 is a number then 9 is odd". We therefore could rephrase the formulas in figure 7.4 as follows:

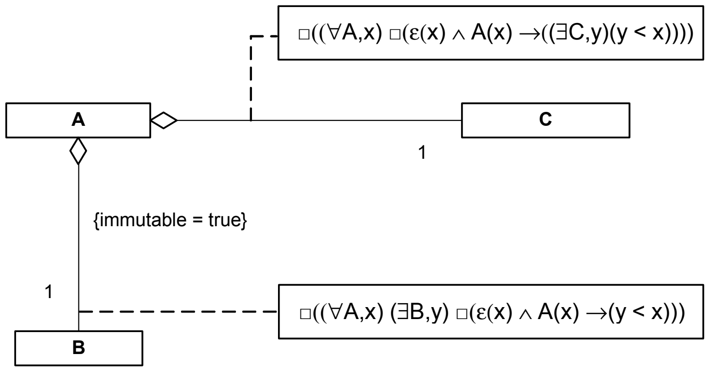
(vi) $\Box((\forall \text{Person},x)(\exists!\text{Heart},y)\ \Box(\varepsilon(x) \wedge \text{person}(x) \rightarrow (y < x)))$
(vii) $\Box((\forall \text{Person},x)\ \Box(\varepsilon(x) \wedge \text{person}(x) \rightarrow (\exists!\text{Heart},y)(y < x)))$

Since Person is a substance sortal (rigid universal), everything that is person is necessarily a person (see chapter 4). In other words, the predicate *person* is modally constant, and for every object selected by the universal quantifier, *person* must be true for this object in every possible world. Consequently, (vi) and (vii) are logically equivalent to their counterparts in figure 7.4.

In order to achieve a uniform axiomatization, we therefore propose the following formula schemas depicted in figure 7.7, which must hold irrespective of the whole universals being rigid or anti-rigid sortals. If the universal A is rigid then A(x) is necessarily true (if true) and the antecedent $(\varepsilon(x) \wedge A(x))$ can be expressed only by $(\varepsilon(x))$. In this case, the B's are truly essential parts of A's.

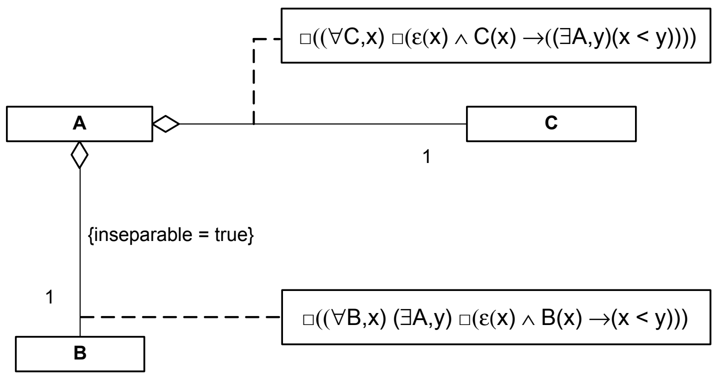
From now on, we refrain from using the term *essential parts* for the cases in which a mere *de dicto* modality is expressed. Therefore, for the case of *specific dependence* from instances of roles (and role mixins) to theirs part we adopt the term *immutable part* instead. Of course, every essential part is also immutable.

Figure 7-7 General representation for Immutable and Mandatory parts



Generalization axioms analogous to those in figure 7.7 can be produced for the case of inseparable and mandatory wholes. Figure 7.8 depicts a representation of inseparable parts and mandatory wholes, in which guard predicates are included to produce generalizations of the axioms in definitions 6.14 and 6.15 that are suitable for the cases of both rigid and anti-rigid universals.

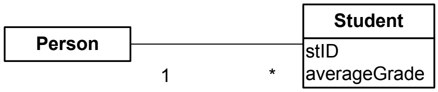
Figure 7-8 General representation for Inseparable Parts and Mandatory Wholes



7.3 Harmonizing different notions of Roles in Conceptual Modeling

In chapter 4, we have discussed the related work of (Wieringa, de Jong and Spruit, 1995), which discusses the need for elaborating on the distinctions among the types of universals used in conceptual modeling. Wieringa and colleagues propose three classifier categories: *static classes*, *dynamic classes* and *roles*. The first two of these correspond to our categories of kinds (and subkinds) and phases, respectively. However, differently from our proposal, in their approach a role universal is not a phased-sortal. Conversely, their roles are *rigid* universals whose instances are said to be *played by* instances of ordinary (static and dynamic) types. The *played by* relation (also termed *inheritance by delegation* by the authors) between a role instance *r* and an object *o* implies that r is *existentially dependent* on *o*. This means that r can only be played by o, and that r can only exist when played by o. However, in contrast, *o* can possibly be associated via the *play by* relation to many instances of the role class (and to many different role classes). Moreover, role universals are responsible for supplying a principle of identity for its instances, which is different from the one supplied by the universals instantiated by their players. Figure 7.9 depicts an example of ordinary and role universals according to the authors.

Figure 7-9 Example with Role and Role player universals



An inspection of the role literature shows, however, that most authors conceive role universals in a way which is akin to the notion proposed in chapter 4, i.e., as substantial universals. This includes authors both in philosophy (Wiggins, 2001; van Leeuwen, 1991) and in conceptual modeling in computer science (Bock & Odell, 1998; Essink & Erhart, 1991; Jungclaus et al., 1991; Sowa, 1984, 1988). Moreover, several authors share the view that the identity of a role instance is supplied by a universal subsuming the role type that it instantiates (i.e., the role's allowed type) (Gottlob & Schrefl & Rock, 1996; Kristensen, 1995; Albano et al., 1993; Richardson & Schwartz, 1991). Finally, there are authors that explicitly share both views see (Guarino & Welty, 2004, 2002a,b, 2000; Steimann, 2001, 2000a,b). In fact, in an extensive study about the topic of roles in the conceptual modeling literature, (Steimann, 2000b) deems the approach of Wieringa and colleagues to be a singular case in which the identity of role instances is not supplied by a universal subsuming the role type they instantiate.

The motivation for such a view proposed by Wieringa and colleagues lies in a philosophical problem known as *The Counting Problem* (Gupta, 1980). Consider the following argument:

KLM served four thousand passengers in 2004
Every passenger is a person
Ergo, KLM served four thousand persons in 2004

Thus, as Wieringa et al. write: *"if we count persons, we may count 1000, but if we count passengers, we may count 4000. The reason for this difference is that if we* *count things we must identify those things, so that we can say which things are the same and which are different. But in order to identify them, we must classify them."*

Although, we appreciate and share the view of connecting *counting with identity* and *identity with classification* (see chapter 4), we do not agree with the conclusion the authors make of this example, namely, that since person and passenger do not share a principle of counting then they must not share a principle of identity either. Since, as we discuss in chapter 4, a principle of identity can only be supplied by a rigid universal. This must be the foundation of the authors' conclusion that a role universal therefore must be a rigid universal.

Why do we think the conclusions made by the authors are not warranted? To start with, in line with (van Leeuwen, 1991), we defend that the counting problem is actually a fallacy. Take the argument posed by its defenders: "The person that boarded flight KL124 on April 22nd, 2004 is a different passenger from the person who boarded flight KL256 on November 19th, 2004, but the two passengers are the same person". We do not agree that it can be correctly said that *the two passengers are the same person*, or, alternatively, that a single *person is distinct passengers* (at different times), if we are truthful to our commonsense use of the *common noun* passenger. However, let us suppose that this is the case, i.e., the person and passenger obey different principles of identity. In this situation, the second premise of the argument is no longer valid, i.e., one cannot say anymore that *every passenger is a person* in a reading in which the copula "is" is interpreted as a relation of identity. This is because, due to Leibniz Law, identity holds necessarily and that identical things necessarily share all their properties (principle of indiscernability of the identicals). As a consequence, we would arrive at the invalid conclusion that every person has all the properties of a passenger in all situations he/she is a person. Moreover, since identity is an equivalence relation, we would have that

"passenger *x* on flight KL124" is identical to person y
"passenger *z* on flight KL256" is identical to person y
Ergo, "passenger *x* on flight KL124" is identical to "passenger *z* on flight KL256"

This conclusion contradicts the initial premise that the two passengers were different. Therefore, if we have the second premise interpreted in the strong reading, one must conclude that passenger carries the same principle of identity as person and, hence, that "passenger *x* on flight KL124" and "passenger *z* on flight KL256" are indeed numerically the same.

In this case, though, the first premise ceases to be true, i.e., one can no longer say that "KLM served two million passengers in 2004".We must conclude then that the second premise should have a weaker reading in which the copula does no represent a relation of identity but one of *coincidence*. But, if this interpretation is taken the whole argument is clearly invalid, since the conclusion cannot be expected to follow from the premises.

In summary, the conclusion that different principles of identity must be supplied by role types and the types instantiated by their players cannot follow from this argument. Furthermore, as pointed out by (Putnam, 1994), if the principle of identity of passengers were not supplied by a unique substantial sortal subsuming passenger (see chapter 4), and if passenger were a rigid classifier whose identity would be tied to the context of particular flight, we would not be able to make sense of sentences such as "The passenger came on flight KL109 and she continued on flight 311".

Despite disagreeing with the conclusions, we think there is an important truth highlighted by the argument of Wieringa and colleagues. If not instances of passengers, what is one counting when stating that "KLM served two million passengers in 2004"? Let us analyze the concept of role proposed by Wieringa et al.:

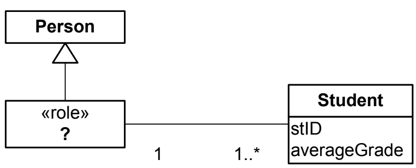
1. a role universal is a rigid classifier;
2. role instances are (one-sidely) existentially dependent of a unique object, which is said to *play* the role;
3. objects *play* these roles only contingently, i.e., the *play* relation is only a contingent relation for the player. As a consequence, ceasing to play the role does not alter the identity of the player object.

A recent work that has a concept of role similar to the one of Wieringa et al. is (Loebe, 2003). However, Loebe's roles are not only *existentially dependent* on their players, but they also depend on the existence of another entity (distinct from their players), in the way, for instance, that *being a student* depends on the existence of a education institution, or *being a husband* depends on the existence of a wife, *being an employee* depends on the existence of an employer, etc. This feature of roles is recognized in our analysis in chapter 4. In fact, it is generally accepted in the literature that roles only exist in a certain context, or in the scope of a certain relation (Masolo et al., 2004, 2005; Guarino & Welty, 2004, 2002a,b, 2000; Steimann, 2001, 2000a,b; Bock & Odell, 1998; Chu & Zhang, 1997; Elmasri & Weeldreyer & Hevner, 1985; Sowa, 1984). Thus, Loebe's notion of roles agrees with that of Wieringa et al. in the points (a), (b) and (c) above, but it also characterizes role instances as existentially dependent on each other.

The concept of role in Wieringa et al. and Loebe is equivalent to our notion of *qua individuals* discussed in chapter 6, since we can interpret their *play by* relation as a sort of inherence. Both relations represent a one-side monadic existential dependence relation. Thus we can say that, like their notion of Roles, our qua individuals (special types of externally dependent modes) are: instances of a rigid classifier (a); one-side existentially dependent of objects, which are related to their "players" via a contingent sort of existential dependence relation (b). Furthermore, a qua individual is a complex of externally dependent modes (e.g., in figure 7.9, *student id*, *average grade*[64]), which, by definition, depends also on the existence of another object extrinsic to its bearer (player). Thus, as in Loebe's concept of roles, asides from the inherence (play) relationship with its bearer (player), our *qua individuals* stand in parthood relationship with a unique relator in the scope of a material relation. Since relators consist of at least two distinct qua individuals, we conclude that the qua individuals composing a relator are existentially dependent on each other.

Now, how can we relate this notion of roles as qua individuals with the one proposed in chapter 4? Let us revisit the example depicted in figure 7.9 above. To start with, a point that can be argued against this model is the representation of optional cardinality constraints. In fact, since no restriction is defined for the allowed type of a role classifier, optional cardinalities must be represented in both Wieringa's and Loebe's approaches. As argued in, for instance, (Weber, 1997; Wand & Storey & Weber, 1999), from an ontological standpoint, there is no such a thing as an optional property and, hence, the representation of optional cardinality leads to unsound models (in the technical sense of chapter 2), with undesirable consequences in terms of clarity. Moreover, as empirically demonstrated in (Bodard et al., 2001), conceptual models without optional properties lead to better performance in problem-solving tasks that require a deeper-level understanding of the represented domain. To put it simply, not all persons bear a student moment, but only those persons that, for example, are enrolled in an educational institution. We can then define a restriction of the universal Person, whose instances are exactly those individuals that bear a student moment, i.e., that are enrolled in an educational institution (see figure 7.10).

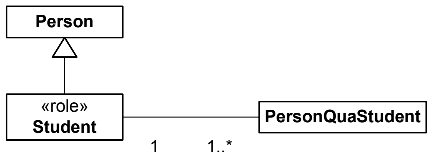
Figure 7-10 A Role universal, its allowed type and an exemplification relation to a qua individual universal



[64] To see that, for example, having a particular *student id* is an externally dependent moment, the reader should imagine a person that is registered in different departments of a university, having a different student id for each department.

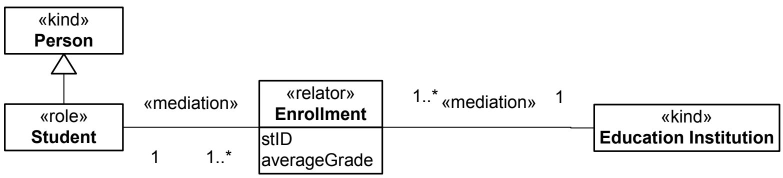
Now, the role universal in figure 7.10 is exactly what we mean by a role in chapter 4 and it is the one idea of role that accurately corresponds to the commonsense use of roles in ordinary language. For this reason, we propose to use the role name for the role universal and to create a new name for the *qua individual* universal (see figure 7.11). This choice is also supported by Aristotle's analysis of substantial and moment universals: student as a universal is *said of subject* (*de subjecto dici*) not exemplified in a subject (*in subjecto est*). That is why the general term Student (Passenger, Employee, etc.) belongs to the grammatical category of *count nouns*, not to the category of *adjectives*[65].

Figure 7-11 A Role universal, its allowed type and an exemplification relation to a qua individual universal (revised version)



Although an improvement of figure 7.10, figure 7.11 is still incomplete in the sense that it does not express the additional dependence relation that a *qua individual* has with other objects extrinsic to its bearer (see definition of qua individual in chapter 6). This problem is solved in figure 7.12, in which relators (as aggregates of qua individuals) are represented explicitly and in which the externally dependent moments of a qua individual are represented as *resultant moments*[66] of the relator. In this figure, the associations between Student and Enrollment and between Education Institution and Enrollment stand for formal relations of mediation (see chapter 6).

Figure 7-12 A role universal, its a allowed type and its associated relator universal



Now, let us return to the "counting problem" previously discussed:

[65] Etymologically the English word *noun* comes from the latin word *substantivus*, meaning expressing *substance*. The original form is still preserved in latin languages such as Portuguese (substantivo) and Italian (sostantivo), as well as in the English word *substantive*, which is a less familiar synonym for noun. Conversely, one of the meanings of *adjective* in English is "not standing by itself, dependent" (Merriam-Webster, 2004).

[66] Resultant properties of an object are properties that a whole inherits from one of its parts (Bunge, 1977, p.97).

500 students graduated from the University of Twente in 2004
Every student is a person
Ergo, 500 persons graduated from the University of Twente in 2004

In this argument, if the first premise is true than the word *student* refers to the mode *Person qua student*. The counting of these entities in a given situation is equal to the cardinality of the extension of the PersonQuaStudent universal in figure 7.11 (i.e., #*ext*(PersonQuaStudent)) or the cardinality of the extension of the Enrollment classifier in figure 7.12 (i.e., #*ext*(Enrollment)), since there is always a 1-1 correspondence between relators and their composing qua individuals. However, if this interpretation for student is assumed, the second premise is simply false, since the relation between a student and a person would be one of inherence, not one of identity. Alternatively, if the word student is interpreted (in the more natural way) as in figure 7.11, then the counting of students is equal to the cardinality of the extension of the Student universal in this figure (i.e., #*ext*(Student)). Though, in this case, premise one is not necessarily true.

In both cases, the alleged "counting problem" disappears. Nonetheless, with the model of figure 7.12 we are still able to represent for both kinds of entities (roles and qua individuals) and their respective counting in an unambiguous manner. Additionally, this solution is able to make explicit and harmonize the two diverse senses of Role which have been used in the conceptual modeling literature. Finally, as we demonstrate in section 6.3.3, there are other important conceptual modeling benefits of this type of representation proposed.

In the same manner that qua individuals are employed to characterize roles in this section, one can think about other types of qua individuals that characterize other types of sortal universals. In this spirit, we can think about a *phase qua-Individual* as (potentially complex) *intrinsic* moments that a substantial bears when in a given phase, or a *substantial qua-Individual*[67] as a potential complex of essential moments, which thus inhere necessarily in an individual. Any attribution (non-sortal) is, in fact, founded on a characterization relation to what can be conceived as some sort of qua individual universal (in the limit case any moment can be thought as a simple qua individual). However, in this thesis, when using the term *qua individual* we mean a *relational qua-Individual*, i.e., a special type of externally

[67] In the Aristotelian tradition there is always a unique universal that answers the question of "what a thing is?" According to Aristotle, seeing an object as an instance of this universal is equivalent to seeing the *individual qua itself* (van Leeuwen, 1991). This universal is named here a kind (or substance sortal).

dependent mode that a substantial has in the context of a certain material relationship.

Finally, we can refine the characteristic *relational dependence* defined for roles in chapter 4 by explicitly relating the notions of role and qua individuals discussed in this section. As discussed in section 4.1, roles are defined as specializations of a substance sortal according to a relational specialization condition φ. Here, this specialization condition can be further analyzed as a material relation being derived from a certain relator universal U_R (see definition 6.17). Consequently, we can state that a role universal bears always a mediation relation to a relator universal.

(1). $\forall x\ \mathbf{Role(x)} \rightarrow \exists y\ \mathbf{RelatorUniversal(y)} \wedge \mathbf{mediation(x,y)}$

Or alternatively, that a role universal is always characterized by a qua individual universal:

(2). $\forall x\ \mathbf{Role(x)} \rightarrow \exists y\ \mathbf{QuaIndividualUniversal(y)} \wedge \mathbf{characterization(x,y)}$

As a consequence of formula (1), we have that, in the UML profile defined in chapter 8 of this thesis, a UML class stereotyped as «role» must always be connected to an association end of a «mediation» relation (see section 8.2.2).

7.4 *Qua Individuals* and the problem of transitivity in mandatory parthood relations between functional complexes

As we have discussed in chapter 5, the parts of a complex object have in common the fact that they all posses a functional link with the complex. In other words, they all contribute to the functionality (or the behavior) of the complex. Moreover, parthood between complexes also represents a case of *functional dependence*. In section 5.6, we have addressed the case of *specific functional dependence* between complexes, exemplified, in a given conceptualization, by the relation between an individual person and the specific brain of that person.

In figure 7.13, we represent a parthood relation between two complexes. However, the functional dependence that is implied by the parthood relation in this example is not one of specific dependence. Conversely, this type of relationship between the universals Heart and Body is what is named *Generic Functional Dependence* between two universals in (Vieu & Aurnague, 2005). This relationship can be defined as follows:

(3). $GFD(X,Y) \equiv \Box(\forall x\ (x::X) \land F(x,X) \rightarrow \exists y\ \neg(y = x) \land (y::Y) \land F(y,Y))$

Figure 7-13 A mandatory parthood relationship between two Complex Object Universals

The predicate F(x,X) in formula (3) has the meaning *x functions as an X*. In Vieu and Aurnague's theory, it is not necessary for an X that it functions as an X. So for instance, although Engine is a rigid concept, it is not the case that in every circumstance an engine functions as an engine. We thus can think of a universal X^F which is a restriction of X according to the restriction condition expressed by the predicate F(x,X), so that every X^F is a X functioning as a X. We name the universal X^F a functional restriction of X. Notice that X^F in this case is an anti-rigid universal which can be characterized by the qua individual q_X. This qua individual, in turn, stands for the moments bearing in an X's while functioning as such, or the particular behaviour of an X while functioning as an X. For instance, an engine x can have the property of emitting a certain number of decibels or being able to perform certain tasks only when functioning as an engine.

In figure 7.14, we can create restrictions of the universals Heart and Body to the universals $Heart^F$ *(FunctioningHeart)* and $Body^F$*(FunctioningBody)*. In this picture, the arrow with the hollow head represents subsumption. The symbols ::, *i* and *ed* represent instantiation, inherence and existential dependence, respectively. Whenever a heart functions as such, i.e., whenever it instantiates the universal *FunctioningHeart*, there is a qua individual q_h that inheres in it. *Mutatis Mutandis*, the same goes to Body and *FunctioningBody* in this picture. As represented in this picture, the qua individuals q_h and q_b are existentially dependent on each other. In this case, $ed(q_h,q_b)$ can be interpreted as "the heart functioning behavior existentially depends on the body functioning behavior". In this model the converse also holds, i.e., that $ed(q_b,q_h)$, or that "the body functioning behaviour existentially depends on the heart functioning behavior". Additionally, according to our model, a heart functioning q_h must inhere a heart h. Likewise, a body functioning q_b must inhere a body b. From this we have that whenever a heart h functions as a heart (i.e., $i(q_h,h)$) there must exist a body functioning behavior q_b (from $ed(q_h,q_b)$), which in turn, inheres a body b (i.e., $i(q_b,b)$). In other words, whenever a heart h functions as a heart, there must be a body b functioning as a body. Again, from the model of figure 7.14 we can derive the converse information, namely, that whenever a body b functions as a body, there must be a heart h functioning as a heart.

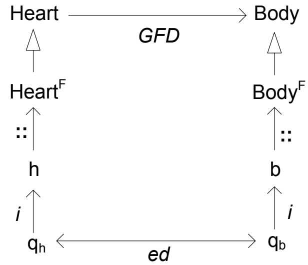


Figure 7-14
Representation of universals with Generic Functional Dependence and their Functional Restrictions

By definition of the relational qua individuals we are considering here (see chapter 6), q_h and q_b in figure 7.14 are externally dependent modes that share the same foundation and therefore can be said to compose a relator r that, in turn, can be said to mediate the instances of *FunctioningHeart* and *FunctioningBody*. This idea is depicted in figure 7.15. The symbols m and $<$ in this picture represent the mediation relation and the proper parthood relation, respectively.

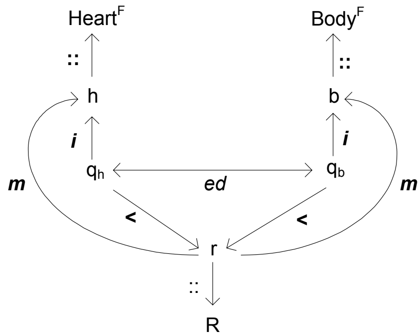


Figure 7-15
Representation of the relator instance composed of two functional qua individuals

As discussed in chapter 6, the relator universal R of which the relator r in figure 7.15 is an instance, can be said to induce the material relation φ_R between the universals FunctioningHeart and FunctioningBody. We shall define here the more general binary predicate $\varphi(x,y) \equiv \exists r\ m(r,x) \wedge m(r,y)$. In other words, $\varphi(x,y)$ holds iff there is a relator r which mediates these two individuals. More naturally, in this case, we can say that φ hold of x and y of type X and Y iff x to function as an X is depends on y functioning as a Y, and vice-versa. Notice that the functional restriction FunctioningHeart (FunctioningBody) is indeed not only an anti-rigid universal but also a *relationally dependent* one and, consequently, it conforms to the formulas (1) and (2) previously discussed: a FunctioningHeart is a Heart functioning as a

Heart in relation to a Body functioning as a Body, and vice-versa. To put in different terms, these functional restrictions of universals are types of Roles.

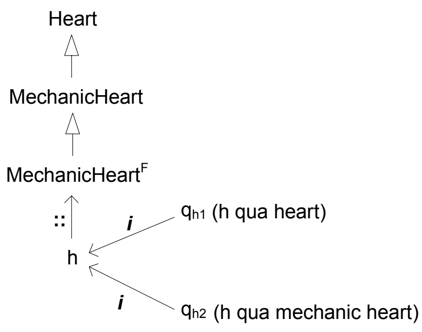
The predicate φ to hold for instances of functional restrictions X^F and Y^F requires the presence of a relator r to mediate these instances. This requires that the functional qua individuals inhering in the mediated instances of X^F and Y^F share a genuine foundation. The formula (3) of generic functional dependence between X and Y can then be better expressed as:

(4). $GFD(X,Y) \equiv \Box(\forall x\ (x::X) \land F(x,X) \rightarrow \exists y\ (y::Y) \land F(y,Y) \land \varphi(x,y))$

Notice that the predicate φ eliminates the possibility that formula (4) is trivially satisfied by its consequent being necessarily true. Another possibility for this formula being trivially satisfied is if its antecedent were to be necessarily false. This would imply that the $(\neg\Diamond\exists x\ (x::X) \land F(x,X))$ is true, which in turn implies that $(\neg\Diamond\exists x\ x::X^F)$. However, this amounts at stating that the universal X^F cannot be possibly instantiated. By adopting an Aristotelian theory of universals in chapter 6 and, therefore, rejecting the possibility of platonic universals, we require that every universal in the theory must be possibly instantiated. We conclude then that formula (4) cannot be trivially satisfied. Moreover, as discussed in chapter 6, a relator must mediate at least two distinct individuals. As a consequence, we have that $\varphi(x,y)$ implies $\neg(y = x)$, rendering this condition superfluous in the consequent of formula (4).

Suppose that the universal X is a specialization of another universal A. Then not only every X is an A but whenever an X functions as such it also functions as an A (Vieu & Aurnague, ibid.). For example, suppose that X and A are the universals MechanicHeart and Heart, respectively. Whenever a MechanicHeart functions as a MechanicHeart, it also functions as a Heart, or alternatively, whenever a MechanicHeart bears the behaviour (or properties) of a functioning MechanicHeart, then it also bears the properties of a functioning Heart. This is illustrated in figure 7.16.

Figure 7-16
Propagation of Functioning to the Supertype



We thus have that

(5). $(F(x,X) \wedge Subtype(X,Y)) \rightarrow F(x,Y)$

Moreover, we assume the following to hold:

(6). $F(x,X) \rightarrow \varepsilon(x)$

That is to say, if x is functioning as an X in a given world then x exists in that world. As previously mentioned, we assume that concrete individuals are not necessarily existents. Therefore, we conclude that an individual X cannot necessarily function as an X either. In other words, a functional restriction of a universal is indeed an anti-rigid universal.

Suppose the situation depicted in figure 7.17.

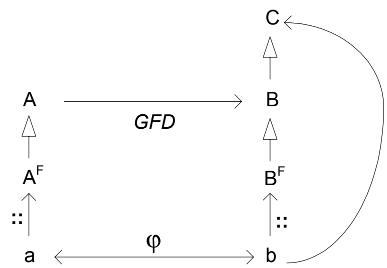


Figure 7-17
Propagation of General Functional Dependence to the Supertype

The universal A is generally functionally dependent on universal B. Thus, for every instance *a* of A that functions as such there is an instance *b* of B functioning as a B. Moreover, the predicate φ holds for *a* and *b*. Now, since b is also a C and, due to (5), b also functions as a C. Hence, we have that whenever an instance *a* of A functions as such there is an instance b of C that functions as a C. Since φ(a,b), we can derive that GFD(A,C). Thus, we have that the following is always true:

(7). $GFD(X,Y) \wedge Subtype(Y,Z) \rightarrow GFD(X,Z)$

Now, suppose the situation depicted in figure 7.18.

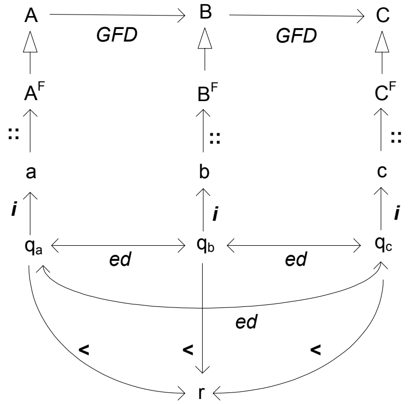


Figure 7-18 Transitivity of General Functional Dependence

In this model, every instance *a* of A functioning as an A bears a particular $q_a$behaviour. The qua individual q_a is existentially dependent on the qua individual q_b, i.e., on the behaviour of a b functioning as a B. However, this model also represents that if b functions as a B (bears q_b) there is a c functioning as a C, i.e., bearing a C behavior q_c. Due to transitivity of existential dependence, we have that q_a is existential dependent also on q_c. Additionally, q_a and q_b share the same foundation and so do q_b and q_c. Thus, q_a and q_c also must share the same foundation. In other words, whatever is responsible for creating q_a and q_b must also be responsible for creating q_c. By definition (see chapter 6), a relator is an aggregation of qua individuals that share the same foundation. We can then define a relator r which consists of q_a, $q_b$and q_c. Consequently, we have that φ(a,b), φ(b,c) and φ(a,c). Now, we have that for every instance *a* of A functioning as an A, there is an instance of c functioning as a C. Since φ(a,c), we then have that GFD(A,C). This argument shows that the following is always true:

(8). $GFD(X,Y) \wedge GFD(Y,Z) \rightarrow GFD(X,Z)$

Although formula (4) defines the notion of general dependence, we need in addition to establish that a functional dependence link holds precisely between two individual entities x and y:

(9). $IFD(x,X,y,Y) \equiv GFD(X,Y) \wedge x::X \wedge y::Y \wedge (F(x,X) \rightarrow F(y,Y))$

This predicate termed *individual functional dependence* states that if an individual x::X is *individually functionally dependent* of another individual y::Y then: (i) there is a generic functional dependence between their types; (ii) x and y are classified as those given types in that world; (iii) for x to function as a X in that world, then y must function as a Y.

An example of individual functional dependence is one between a particular heart h and a particular body b in figures 7.14 and 7.15. As previously discussed, there is a generic functional dependence between the universals Heart and Body, and if in a given circumstance a heart h functions as a heart there is a body b that functions as a body in that circumstance.

Let us now return to the example of figure 7.13 of a mandatory parthood relation between the universals Heart and Body. In this model, a particular heart h is not only functionally dependent of a body b in a given world, but h is also part of b. This type of the parthood relation is termed in (Vieu & Aurnague, 2005) *direct functional parthood of type 1*:

Definition 7.1 (Direct Functional Part of type 1): An individual x instance of X is a direct functional part of type 1 of an individual y of type Y (symbolized as $d_1(x,X,y,Y)$) iff x is a part of y and x is individually functionally dependent of y. Formally, $d_1(x,X,y,Y) \equiv ((x < y) \wedge IFD(x,X,y,Y))$.

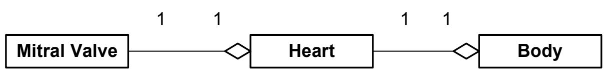
■

Examples of d_1 include cuff-sleeve, stem-plant, carburetor-engine, finger-hand, hand-arm, arm-body, hand-body, heart-body, heart-circulatory system.

In conformance with the findings of (Vieu & Aurnague, 2005), we propose that a mandatory parthood relation between two *functional complexes*[68] (such as the one depicted in figure 7.13) should be interpreted as a case of *direct functional parthood.* In this specific case, the model implies that: $\Box(\forall x\ x{::}Heart \rightarrow (\varepsilon(x) \rightarrow \exists y\ y{::}Body \wedge d_1(x,Heart,y,Body)))$.

Now, suppose that we have a model such as the one represented of figure 7.19.

Figure 7-19 Example of direct functional parts of type 1



In this case, both the relationships between Heart and Body, and between Mitral Valve and Heart, are mapped in the instance level to cases of *direct functional parthood(1)*, i.e.,

(i) $\Box(\forall x\ x{::}Heart \rightarrow (\varepsilon(x) \rightarrow \exists y\ y{::}Body \wedge d_1(x,Heart,y,Body)))$

[68] See definition of a functional complex in chapter 5.

(ii) $\Box(\forall x\ x::MitralValve \rightarrow (\varepsilon(x) \rightarrow \exists y\ y::Heart$
$\wedge\ d_1(x,MitralValve,y,Heart)))$

The important question at this point is: from (i) and (ii), can we derive formula (iii) below?

(iii) $\Box(\forall x\ x::MitralValve \rightarrow (\varepsilon(x) \rightarrow \exists y\ y::Body$
$\wedge\ d_1(x,MitralValve,y, Body)))$

We here adopt the so-called *mereological continuism* (Simons, 1987), which advocates that parthood only holds between existents. Since d_1 implies parthood, we conclude that d_1 also holds exclusively between existents. Thus, if in every world w that x::X exists there is a y::Y such $d_1(x,X,y,Y)$ in w then in every world w in which x::X exists there is a y::Y which also exists in w. Additionally, if in every world w that y::Y exists there is a z::X such that $d_1(y,Y,z,Z)$ in w then in every world w in which y::Y exists there is a z::Z which also exists in w. From this we conclude that in every world w that x::X exists there are y::Y and z::Z that also exist in w and both $d_1(x,X,y,Y)$ and $d_1(y,Y,z,Z)$ are true in that world. Consequently, we have that (iii) follows from (i) and (ii) iff d_1 is transitive. Thus, the question above can be rephrased as: is direct functional parthood(1) a transitive relation?

In the sequel we demonstrate that this is indeed the case. The abbreviations in the proofs are: (a) TFP (transitivity of formal parthood); (b) TLI (transitivity of the logical implication); (c) EC (Elimination of the Disjunction), and (d) IC (Introduction of the Disjunction).

(T1) Theorem 1: $d_1(x,X,y,Y) \wedge d_1(y,Y,z,Z) \rightarrow d_1(x,X,z,Z)$
<u>Proof:</u>

1. $d_1(x,X,y,Y)$	T1
2. $d_1(y,Y,z,Z)$	T1
3. $(x < y) \wedge IFD(x,X,y,Y)$	1, Definition 7.1
4. $GFD(X,Y) \wedge x::X \wedge y::Y \wedge (F(x,X) \rightarrow F(y,Y))$	3, (8)
5. $(y < z) \wedge IFD(y,Y,z,Z)$	2, Definition 7.1
6. $GFD(Y,Z) \wedge y::Y \wedge z::Z \wedge (F(y,Y) \rightarrow F(z,Z))$	5, (8)
7. $(x < z)$	3,5, TFP
8. $GFD(X,Z)$	4,6, (7)
9. $(x::X) \wedge (z::Z)$	4,6, EC
10. $(F(x,X) \rightarrow F(z,Z))$	4,6, TLI
11. $GFD(X,Z) \wedge (x::X) \wedge (z::Z)\ \ (F(x,X) \rightarrow F(z,Z))$	8,9,10, IC

12. IFD(x,X,z,Z) 11, (8)
13. (x < z) ∧ IFD(x,X,z,Z) 7,12, IC
14. d_1(x,X,z,Z) 13, Definition 7.1

□

We can generalize this result for any chain of direct functional dependence in a model. Another example of such case is depicted in figure 7.20.

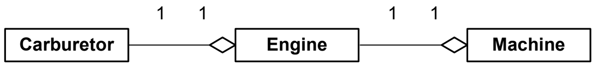


Figure 7-20 Examples of direct functional parts of type 1; transitivity always hold across parthood relations of this type

In models such 7.13 and 7.20, the mandatory parthood relation represents functional dependence in both directions. Take for instance figure 7.20. As discussed in chapter 5, the minimum cardinality constraint of 1 in the Engine association end of the aggregation relation between Carburator and Engine implies that every instance of Carburator necessitates an Engine to function as a Carburator. Likewise, the minimum cardinality constraint of 1 in the Carburator association end of that relation implies that every Engine necessitates a Carburator to function as an Engine. (Vieu & Aurnague, 2005) name this type of functional parthood in which *x is part of y* but *y as Y is individually functionally dependent on x as an X direct functional parthood (2)*:

Definition 7.2 (Direct Functional Part of type 2): An individual x instance of X is a direct functional part of type 2 of an individual y of type Y (symbolized as d_2(x,X,y,Y)) iff x is a part of y and y is individually functionally dependent of x. Formally, d_2(x,X,y,Y) ≡ (x < y) ∧ IFD(y,Y,x,X).

■

Examples of d_2 include wall-house, engine-car, electron-atom, atom-molecule, finger-hand, hand-arm, cell-heart, and feather-canary.

In the sequel, we prove that d_2 is also transitive.

(T2) Theorem 2: d_2(x,X,y,Y) ∧ d_2(y,Y,z,Z) → d_2(x,X,z,Z)
Proof:

(1). d_2(x,X,y,Y) T2
(2). d_2(y,Y,z,Z) T2
(3). (x < y) ∧ IFD(y,Y,x,X) 1, Definition 7.2
(4). GFD(Y,X) ∧ y::Y ∧ x::X ∧ (F(y,Y) → F(x,X)) 3, (8)

(5). $(y < z) \wedge IFD(z,Z,y,Y)$	2, Definition 7.2
(6). $GFD(Z,Y) \wedge z{::}Z \wedge y{::}Y \wedge (F(z,Z) \rightarrow F(y,Y))$	5, (8)
(7). $(x < z)$	3,5, TFP
(8). $GFD(Z,X)$	4,6, (7)
(9). $(z{::}Z) \wedge (x{::}X)$	4,6, EC
(10). $(F(z,Z) \rightarrow F(x,X))$	4,6, TLI
(11). $GFD(Z,X) \wedge (z{::}Z) \wedge (x{::}X)\ \ (F(z,Z) \rightarrow F(x,X))$	8,9,10, IC
(12). $IFD(z,Z,x,X)$	11, (8)
(13). $(x < z) \wedge IFD(z,Z,x,X)$	7,12,IC
(14). $d_2(x,X,z,Z)$	13, Definition 7.2

□

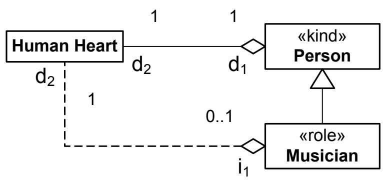
Whenever in a conceptual model we have a representation of a mandatory parthood relation between complex objects such as in figures 7.13 and 7.20, we have both a case of d_1 and a case of d_2. In particular, the model of figure 7.20 implies both formulas below:

(i) $\Box(\forall x\ x{::}Carburator \rightarrow (\varepsilon(x) \rightarrow \exists y\ y{::}Engine \wedge d_1(x,Carburator,y,Engine)))$
(ii) $\Box(\forall x\ x{::}\ Engine \rightarrow (\varepsilon(x) \rightarrow \exists y\ y{::}\ Carburator \wedge d_2(y,Carburator,x,Engine)))$

Since both d_1 and d_2 are transitive, we maintain that transitivity holds within any chain of direct functional dependence relations in a conceptual model.

Now, take for instance the relationship depicted in figure 7.21 below.

Figure 7-21 Example of an indirect functional parthood of type 1 (from Human Heart to Musician)



Every human heart necessitates a person, and every person necessitates a human heart, i.e., both d_1 and d_2 hold between direct instances of human heart and person. Moreover, every musician is a person. So, as any person, a musician necessitates a human heart, i.e., d_2 holds also between instances of human heart and musician. However, it is not the case that a *direct functional dependence* holds between human heart and musician. A human heart necessitates a person, but this person does not have to be a musician (this is made evident by the cardinality 0..1 of the inherited relation between these two universals). This type of relationship is termed *indirect* *functional parthood (1)* in (Vieu & Aurnague, 2005) and it is defined as follows:

Definition 7.3 (Indirect Functional Part of type 1): $i_1(x,X,y,Y) \equiv (x < y) \wedge IIFD(x,X,y,Y)$. IIFD(x,X,y,Y) is the relation of *individual indirect functional dependence* and is defined as

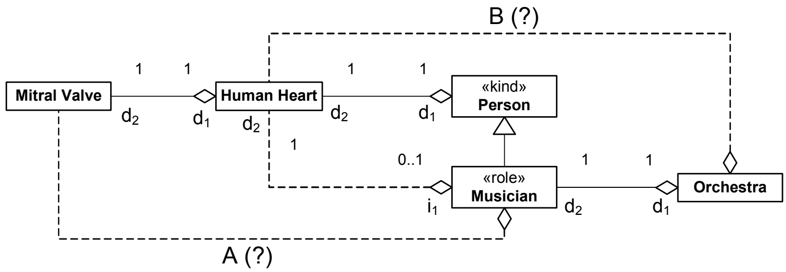
(9). $IIFD(x,X,y,Y) \equiv y::Y \wedge \exists Z\ (Subtype(Y,Z) \wedge IFD(x,X,y,Z))$.

■

To put it in a simple way, x as an X is individually indirect functional dependent of y as a Y iff for x to function as an X, y must function as a Z, whereas Z is a more general universal (subsuming that Y) that y instantiates. Examples of i_1 include handle-door (with "movable entity" for type subsuming "door"), door-house (with "wall, enclosure or building" subsuming "house"), engine-car (with "machine" subsuming "car"), brick-wall (with "construction" subsuming "wall"), valve-carburetor (with "fluid-holding device" subsuming carburetor), cell-heart (with "organ" subsuming "heart"), feather-canary (with "bird" subsuming "canary").

Now, take the model depicted in figure 7.22 below. There are two potential parthood relations **A** and **B**. The relation **A** between Mitral Valve and Musician holds iff transitivity holds across (Mitral Valve $\xrightarrow{d1}$ Human Heart) and (Human Heart $\xrightarrow{i1}$ Musician), since in the other reading of these relations, i.e., (Mitral Valve $\xrightarrow{d2}$ Human Heart) and (Human Heart $\xrightarrow{d1}$ Musician), transitivity is already guaranteed by theorem (T2). To put it baldly, relation **A** is transitive in this case iff $\mathbf{d_1(x,X,y,Y) \wedge i_1(y,Y,z,Z) \rightarrow i_1(x,X,z,Z)}$ is a theorem. Likewise, relation **B** is transitive in this case iff $\mathbf{i_1(x,X,y,Y) \wedge d_1(y,Y,z,Z) \rightarrow i_1(x,X,z,Z)}$ is a theorem.

Figure 7-22 Two candidate parthood relations due to transitivity.



As we show in the sequel, $\mathbf{d_1(x,X,y,Y) \wedge i_1(y,Y,z,Z) \rightarrow d_1(x,X,z,Z) \vee i_1(x,X,z,Z)}$ is a theorem (T3) while $\mathbf{i_1(x,X,y,Y) \wedge d_1(y,Y,z,Z) \rightarrow d_1(x,X,z,Z) \vee i_1(x,X,z,Z)}$ is not. Therefore, whilst **A** is a case of indirect functional parthood between Mitral Valve and Musician, relation **B** is not warranted and, hence, must not exist in figure 7.22.

(T3) Theorem 3: $\mathbf{d_1(x,X,y,Y) \wedge i_1(y,Y,z,Z) \rightarrow i_1(x,X,z,Z)}$

Proof:

(1). $d_1(x,X,y,Y)$	T3
(2). $i_1(y,Y,z,Z)$	T3
(3). $(x < y) \wedge IFD(x,X,y,Y)$	1, Definition 7.1
(4). $GFD(X,Y) \wedge x::X \wedge y::Y \wedge (F(x,X) \rightarrow F(y,Y))$	3, (8)
(5). $(y < z) \wedge IIFD(y,Y,z,Z)$	2, Definition 7.3
(6). $z::Z \wedge \exists W\ (Subtype(Z,W) \wedge IFD(y,Y,z,W))$	5, (9)
(7). $GFD(Y,W) \wedge y::Y \wedge z::W \wedge (F(y,Y) \rightarrow F(z,W))$	6, (8)
(8). $(x < z)$	3,5,TFP
(9). $GFD(X,W)$	4,7, (7)
(10). $(x::X) \wedge (z::W)$	4,7, EC
(11). $(F(x,X) \rightarrow F(z,W))$	4,7,TLI
(12). $GFD(X,W) \wedge (x::X) \wedge (z::W)\ \ (F(x,X) \rightarrow F(z,W))$	9,10,11,IC
(13). $IFD(x,X,z,W)$	12, (8)
(14). $z::Z \wedge \exists W\ (Subtype(Z,W) \wedge IFD(x,X,z,W)$	6,13,IC
(15). $IIFD(x,X,z,Z)$	14, (9)
(16). $(x < z) \wedge IIFD(x,X,z,Z)$	8,15, IC
(17). $i_1(x,X,z,Z)$	16, Definition 7.3

□

Let us now modify the model of figure 7.22 to depict a more realistic conceptualization. In this modified specification (figure 7.23) we have that every Blood Pump is part of a Circulatory System and necessitates a Circulatory System in order to work as such (d_1). Likewise, every Circulatory System has as part a Blood Pump and necessitates the latter to work as such (d_2). As any Blood Pump, a Biological Heart is part of a Circulatory System and necessitates a Circulatory System to work as such, i.e., *direct functional dependence (2)* is inherited by Biological Heart from the subsuming universal. The same obviously holds for Artificial Heart. However, it is not the case that a Circulatory System is directly functionally dependent of a Biological Heart specifically. To put it in an alternative way, a Circulatory System, in order to function as such, relies on the behavior of a Blood Pump, but this behavior does not have to be afforded in the specific way a Biological Heart does. In (Vieu & Aurnague, 2005), this type of relationship between Biological Heart and Circulatory System is termed *indirect functional parthood (2)* and it is defined as follows:

Definition 7.4 (Indirect Functional Part of type 2): $i_2(x,X,y,Y) \equiv (x < y) \wedge IIFD(y,Y,x,X)$.

■

Examples of i_2 include heart-circulatory system (with "blood pump" subsuming "heart"), brick-wall (with "construction material" subsuming "brick").

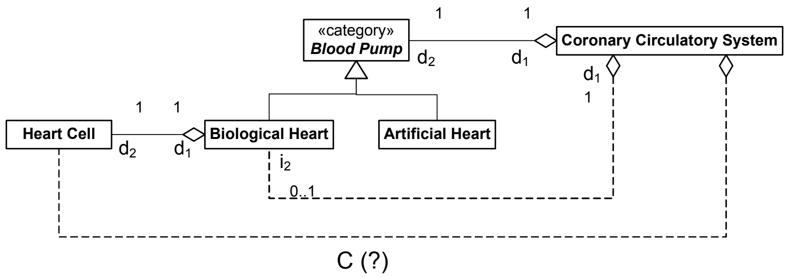


Figure 7-23 Example of an indirect functional parthood of type 2 (from Biological Heart to Coronary Circulatory System) and of a candidate parthood relationship (C) due to transitivity

Once more, we have the question: does transitivity hold across (Heart Cell $\xrightarrow{d2}$ Biological Heart) and (Biological Heart $\xrightarrow{i2}$ Coronary Circulatory System)? In the other reading we have (Heart Cell Valve $\xrightarrow{d1}$ Biological Heart) and (Biological Heart $\xrightarrow{d1}$ Coronary Circulatory System), thus, relation **C** is warranted iff the question above is answered affirmatively. The answer in this case is negative, since $\mathbf{d_2(x,X,y,Y) \wedge i_2(y,Y,z,Z) \rightarrow d_2(x,X,z,Z) \vee i_2(x,X,z,Z)}$ cannot be shown to be a theorem in this theory. However, the following is a theorem:

(T4) Theorem 4: $\mathbf{i_2(x,X,y,Y) \wedge d_2(y,Y,z,Z) \rightarrow i_2(x,X,z,Z)}$
Proof:

(15).	$i_2(x,X,y,Y)$	T4
(16).	$d_2(y,Y,z,Z)$	T4
(17).	$(x < y) \wedge IIFD(y,Y,x,X)$	1, Definition 7.4
(18).	$(y < z) \wedge IFD(z,Z,y,Y)$	2, Definition 7.2
(19).	$x::X \wedge \exists W\ (Subtype(X,W) \wedge IFD(y,Y,x,W))$	3, (9)
(20).	$GFD(Y,W) \wedge y::Y \wedge x::W \wedge (F(y,Y) \rightarrow F(x,W))$	5, (8)

(21).	$GFD(Z,Y) \land z::Z \land y::Y \land (F(z,Z) \rightarrow F(y,Y))$	4, (8)
(22).	$(x < z)$	3,4, TFP
(23).	$GFD(Z,W)$	6,7, (7)
(24).	$(z::Z) \land (x::W)$	6,7,EC
(25).	$(F(z,Z) \rightarrow F(x,W))$	6,7,TLI
(26).	$GFD(Z,W) \land (z::Z) \land (x::W) \;\; (F(z,Z) \rightarrow F(x,W))$	9,10,11,IC
(27).	$IFD(z,Z,x,W)$	12,(8)
(28).	$x::X \land \exists W\, (Subtype(X,W) \land IFD(z,Z,x,W))$	5,13,IC
(29).	$IIFD(z,Z,x,X)$	14,(9)
(30).	$(x < z) \land IIFD(z,Z,x,X)$	8,15,IC
(31).	$i_2(x,X,z,Z)$	16, Definition 7.4

□

Due to this theorem we have that the relation D between Biological Heart and Circulatory System (depicted in figure 7.24 below) is warranted, since transitivity holds across (Biological Heart $\xrightarrow{i2}$ Coronary Circulatory System) and (Coronary Circulatory System $\xrightarrow{d2}$ Circulatory System) in this case.

Figure 7-24 Example of an indirect functional parthood of type 2 due to transitivity (from Biological Heart to Circulatory System)

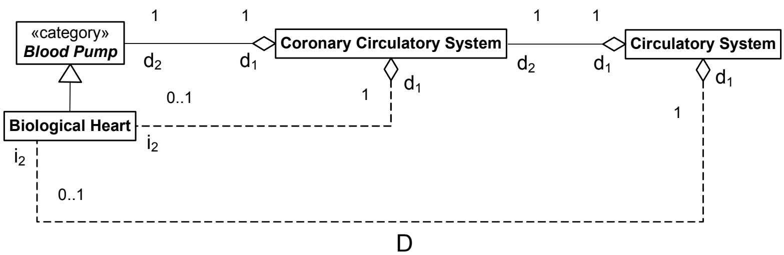


Figure 7.25 below summarizes the results of this section in the following manner: Mandatory parthood between complex objects (functional parthood) is a non-transitive relation (i.e., transitive in certain cases and intransitive in others). Transitivity can be guaranteed for these relations only in cases where the patterns of figures (7.25.a-c) occur.

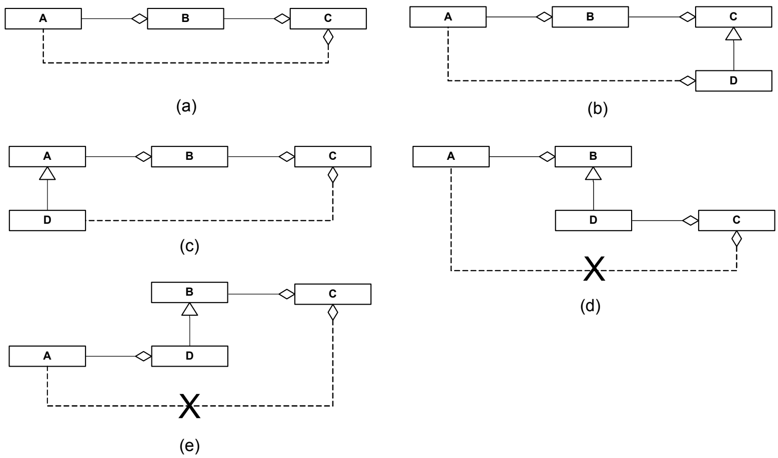


Figure 7-25 The patterns of figures (a), (b) and (c) represent cases in which a derived transitive parthood relation can be inferred. Intransitive cases are shown in figures (d) and (e)

7.5 Final Considerations

In this chapter, we build upon some of the results developed in chapters 4, 5 and 6 of this thesis to provide a foundation for the modeling concept of roles.

First, by employing the categories from the theory of universals presented in chapter 4 and its postulates, we were able to propose a design pattern to target a recurrent problem in role modeling discussed in the literature. We believe that the definition of design patterns capturing standard solutions for ontological modeling problems contribute greatly to the task of defining sound engineering tools for conceptual modeling. Nonetheless, the investigation of ontological design patterns is still in its infancy, and very few examples exist in the literature. Two examples are the *whole-part* design pattern introduced in (Guizzardi & Falbo & Pereira Filho, 2002) and the Inflammation pattern proposed in (Gangemi & Catenacci & Battaglia, 2004).

In chapter 5, we present a number of *modal* meta-properties characterizing part-whole relations. Here, we use these meta-properties to investigate their intertwining with anti-rigid universals and, in particular, with role universals standing as wholes in parthood relationships. We use the distinction between *de re* and *de dicto* modality in the literature of philosophical logic to characterize the different formal properties of the relations of specific dependence parthood from the whole to the part, depending whether the whole universal is rigid or anti-rigid.

By using the concept of role (as anti-rigid and relationally dependent sortals) introduced in chapter 4, and the concept of *qua individuals* (externally dependent modes) introduced in chapter 6, we manage to reconcile two incompatible views of roles present in the literature, as well as to present a modeling solution that retains the benefits of both approaches.

Finally, in chapter 5, we have discussed the so-called problem of fallacious transitivity in part-whole relations. In that chapter, we address this problem for three types of conceptual parthood, namely, the *subQuantityOf*, *subCollectionOf* and *memberOf* relations. However, the remaining relation (*componentOf*) is the one which is most commonly represented in conceptual models, since most individuals represented in these models are functional complexes. As discussed in that chapter, transitivity among *componentOf* relations only holds in certain contexts, but the definition of these contexts typically demands a substantial knowledge of the modeling domain and of the characterizing relations that unify its entities. With the intent to provide some methodological tools for helping the modeler in this task, we propose a number of visual patterns that can be used to delimit these contexts in conceptual class diagrams. The patterns are elaborated by using the concepts of relators, qua individuals and existential dependency relations discussed in chapter 6, but chiefly by building on the pioneering theory of transitivity of linguistic functional parthood relations proposed by (Vieu & Aurnague, 2005).

Chapter 8

A Case Study on Ontology-Based Evaluation and Re-Design

In this chapter, we conduct two complementary case studies of the techniques developed in this thesis.

Firstly, in order to demonstrate the language evaluation framework proposed in chapter 2, we systematically analyze and re-design the Unified Modeling Language (UML) as a conceptual modeling language. To achieve this objective, we employ the *foundational ontology* proposed throughout chapters 4 to 7 of this thesis as a basis for the evaluation of the current UML 2.0 metamodel, and for the definition of the real-world semantics of the ontologically well-founded version of UML that results from this re-designing process. This process is conducted in three complementary sections: section 8.1 uses the results mainly from chapter 4 to address the UML metamodel elements of *class* and *class taxonomies*; section 8.2 uses the results mainly from chapter 6, taking a broader view on UML classifiers (which include, besides classes, also associations, datatypes and interfaces), but also analyzing the UML *Property* metaclass (e.g., attributes and association ends); Finally, section 8.3 uses the results mainly from chapter 5 to address the representation of meronymic relations (e.g., composition and aggregation) in the UML metamodel. We employ the same structure in each of these three subsections: each subsection contains two parts; in the first part we always present a relevant fragment of the UML metamodel; in a second part we do three things, namely, (a) analyze this fragment in terms of a suitable part of our foundational ontology; (b) propose a revised version of the UML metamodel in conformance with the foundation ontology; (c) propose a modelling profile that implements the revised metamodel.

Secondly, once the language has been re-designed, in section 8.5 we demonstrate its adequacy in tackling some recurrent representation problems as well as some problems of semantic interoperability in the modeling and integration of *lightweight ontologies* to be used for context-aware service platforms.

Section 8.5 presents some final considerations for this chapter.

8.1 Classes and Generalization

8.1.1 The UML Metamodel

In this section we briefly present the UML metamodel, focusing on static elements and, in particular, those modeling primitives used in the construction of UML class diagrams. From now on, we refer to the *UML 2.0 Superstructure Specification* (OMG, 2003c) when quoting in italics the definition of elements of the UML metamodel.

Figure 8.1 below depicts a fragment of the UML metamodel featuring the metaclasses which are more relevant for the purpose of this discussion. In particular, we focus our discussion on those meta-constructs highlighted in this figure, namely, *Classifier*, *Class*, *Generalization* and *GeneralizationSet*.

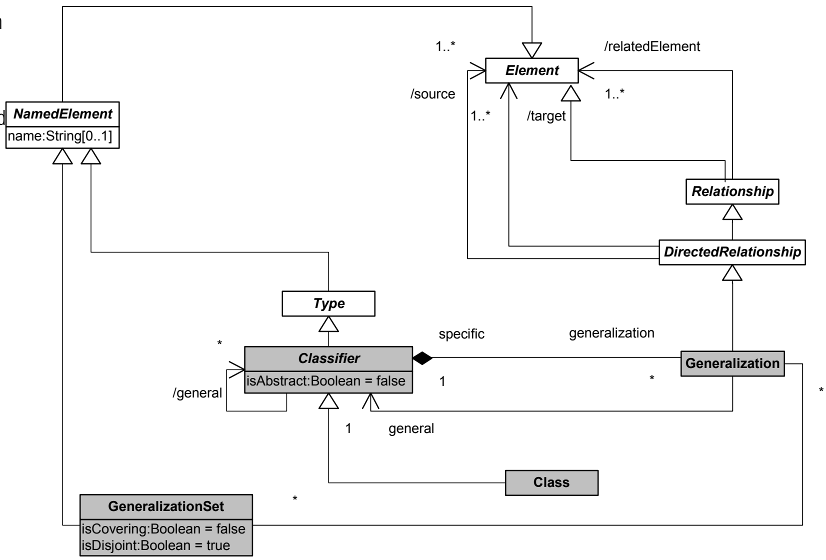


Figure 8-1 Excerpt from the UML metamodel featuring the metaclasses *Classifier*, *Class*, *Generalization* and *GeneralizationSet*

In the UML superstructure specification, a classifier is defined as follows: *"A classifier is a classification of instances according to their features. [It] may participate in generalization relationships with other Classifiers [, in which case] an instance of a specific Classifier is also an (indirect) instance of each of the general Classifiers. Any constraint applying to instances of the general classifier also applies to* *instances of the specific classifier. A Classifier defines a type. Type conformance between generalizable Classifiers is defined so that a Classifier conforms to itself and to all of its ancestors in the generalization hierarchy. [It] can specify a generalization hierarchy by referencing its general classifiers. If [the classifier's attribute isAbstract =] true, the Classifier does not provide a complete declaration and can typically not be instantiated. An abstract classifier is intended to be used by other classifiers, e.g. as the target of general metarelationships or generalization relationships. [The] default value is false."*

A classifier in UML is a general abstract metaclass that subsumes other metaclasses such as *Class*, *Association*, *Interface* and *Datatype*. Its main purpose is to describe the general aspects of property (attribute) description and generalization hierarchies (taxonomies) for its subclasses. As recognized in the UML specification, among its different subtypes *"class is the most widely used classifier"*. A class is a type of classifier that *"describes a set of Objects sharing a collection of Features, including Operations, Attributes and Methods, that are common to the set of Objects… [Its purpose] is to specify a classification of objects and to specify the features that characterize the structure and behavior of those objects"*.

In UML, a generalization is shown as a line with a hollow triangle as an arrowhead between the symbols representing the involved classifiers. The arrowhead points to the symbol representing the general classifier. A generalization relationship may be connected to a *GeneralizationSet*. This can be depicted graphically in one the following ways:

1. By placing the name of the GeneralizationSet explicitly in an adjacent position to that of the generalization symbol. Therefore, all Generalization relationships with the same GeneralizationSet name are part of the same GeneralizationSet (figure 8.2.a);

2. By drawing two or more lines (ends of the generalization line symbol) that converge to a single arrowhead. This symbolizes that all the specific classifiers belong to the same GeneralizationSet (figure 8.2.b);

3. By drawing a dashed line across those lines with separate arrowheads that are meant to be part of the same set. This dashed line designates the GeneralizationSet (figure 8.2.c).

In cases (b) and (c), the representation of the GeneralizationSet name is optional.

A GeneralizationSet defines a particular set of Generalization relationships that describe the way in which a specific Classifier (or superclass) may be partitioned. For example, in figures 8.2.a, 8.2.b and 8.2.c below, a GeneralizationSet named *gender* defines the partitioning of class Person in the two subclasses: MalePerson and FemalePerson.

However, in the UML specification, the term *partition* is not used necessarily in the mathematical sense. This is only the case if both attributes *isCovering* and *isDisjoint* of a GeneralizationSet have the *true* value. In the case of the aforementioned example, if (*isCovering* = *true*) then every instance of Person must be either an instance of MalePerson or FemalePerson, i.e., in every world the extensions of the subclasses that are part of the generalization set exhaust the extension of their superclass. If (*isDisjoint* = *true*) then there is no instance of Person who can be both and instance of MalePerson and of FemalePerson, i.e., within a world, the extensions of the subclasses that are part of the generalization set are necessarily mutually disjoint. The default interpretation in UML is that (*isDisjoint* = *true*) but (*isCovering* = *false*).

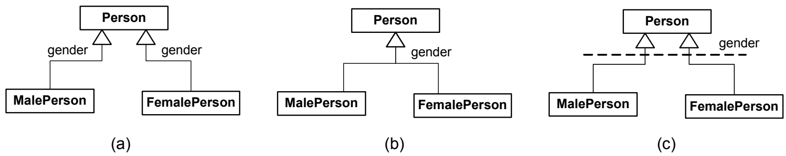


Figure 8-2 Alternative representations for a GeneralizationSet

8.1.2 Ontological Interpretation and Re-Design

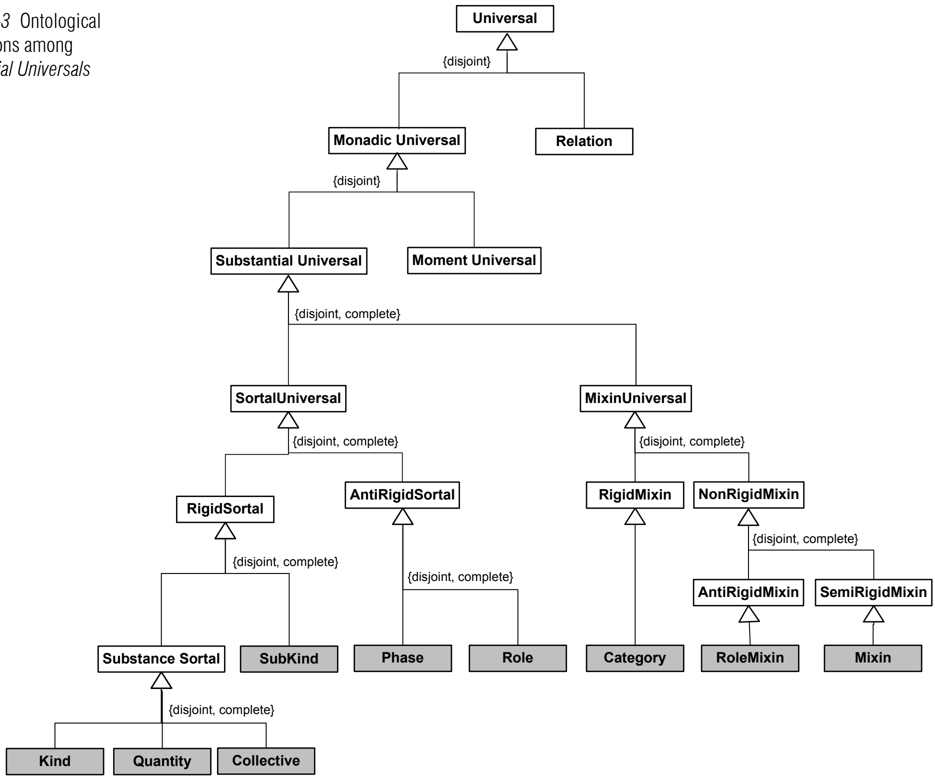
In this section we focus in a special sense of the UML metaclass *Class*. By class hereby we mean the notion of a first-order class, as opposed to *powertypes*, and one whose instances are single objects, as opposed to *association classes*, whose instances are tuples of objects.

In this sense, the ontological interpretation of a UML Class is that of a *monadic universal* (see figure 8.3). Class diagrams are intended to represent the static structure of a domain. For this reason, we take here that classes should always be interpreted as representing *endurant universals*. In addition, since substantials are prior to Moments from an identification point of view, i.e., the latter are *identificationally dependent* on the former (Schneider, 2003b), it is typically the case that most classes in a class diagram should be interpreted as substantial universals. In fact, we prescribe that a non-stereotyped class in a class diagram should be interpreted as a substantial universal.

Classes are therefore the representation of general terms in a description of a domain. A general requirement in conceptual modeling languages is that the represented instances must have a definite identity (Borgida, 1990). For this reason, classes in a class diagram representing substantial universals should always be interpreted as *Object universals*.

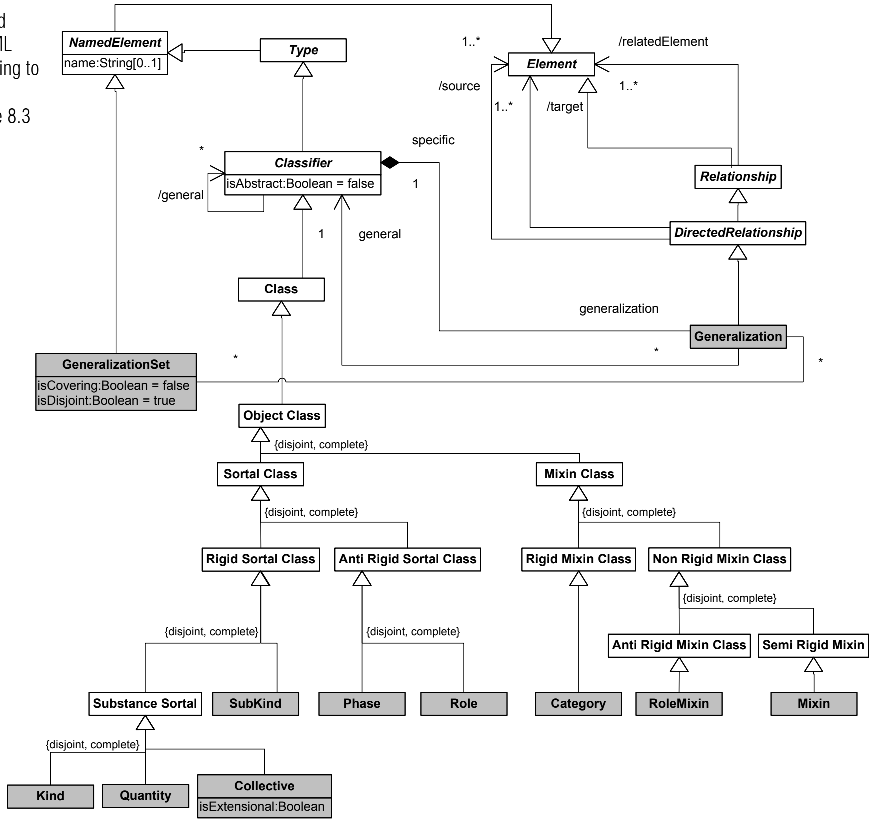
In chapter 4 of this thesis, we present a theory that proposes a number of ontological distinctions that refine the category of *Object Universal*. One of the sorts of object universal is a substance sortal, or a *Kind*. As discussed in depth in chapter 5 of this thesis, within the category of *kinds*, we can make three further types distinctions, based on the ontological categories of their instances. These distinctions are *quantity* kind, *collective* kind and *functional complex* kind. Functional complex kinds are certainly the most commonly represented kinds in conceptual specifications. For this reason, for now on, we simply write *kind* when referring to a functional complex kind. These ontological distinctions are depicted in figure 8.3 below.

Figure 8-3 Ontological Distinctions among *Substantial Universals*



If we make a representation mapping from the ontology of figure 8.3 to the UML metamodel, we can map the category of *Monadic Universal* to the UML element of a *Class*. However, by carrying on this process, we realize that in UML there are no modeling constructs that represent the ontological categories specializing *Object Universal* in this figure. In other words, there are ontological concepts prescribed by our reference ontology that are not represented by any modeling construct in the language. This amounts to a case of ***ontological incompleteness*** at the modeling language level (see section 2.3.4).

Figure 8-4 Revised fragment of the UML metamodel according to the ontological categories of figure 8.3



In order to remedy this problem, we propose a first extension to UML that represents finer-grained distinctions between different types of classes. We, thus, define extensions of the UML metaclass *Class* to represents each of the leaf ontological categories specializing *substantial universal* in figure 8.3. The extended UML metamodel must also include a number of constraints (derived from the postulates of the theories presented in chapters 4 and 5) that restrict the ways the introduced elements can be related. The goal is to have a metamodel such that all syntactically correct specifications according to this metamodel have logical models that are *intended world structures* of the conceptualizations they are supposed to represent (see section 3.4.1).

In figure 8.4 above we present an extension of the UML meta-model in order to represent the ontological concepts discussed in this section. The concrete metaclasses in this figure (highlighted in grey) represent the leaf ontological distinctions among the category of *Object Universals* prescribed by our theory.

In the sequel, we define a *profile* that implements the metaclasses of this (extended) UML metamodel, as well as their interrelationships and contraints. By using this profile, the concrete *Object classes* in figure 8.4 are represented in conceptual models as stereotyped classes representing each of the considered ontological distinctions. Likewise, the admissible relations between these ontological categories, derived from the postulates of our theory, are represented in the profile as syntactical constraints governing the admissible relations between the corresponding stereotyped classes. This modeling profile is presented in table 8.1 below.

Table 8-1 Ontologically Well-Founded UML modelling profile according to the ontological categories of figure 8.3

Metaclass	Description
Substance Sortal	*Substance Sortal* is an abstract metaclass that represents the general properties of all *substance sortals*, i.e., rigid, relationally independent object universals that supply a principle of identity for their instances. Substance Sortal has no concrete syntax. Thus, symbolic representations are defined by each of its concrete subclasses.
Constraints	
1. Every substantial object represented in a conceptual model using this profile must be an instance of a substance sortal, directly or indirectly. This means that every concrete element of this profile used in a class diagram (isAbstract = false) must include in its general collection one class stereotyped as either «kind», «quantity» or «collective». This constraint is a refinement of postulate 4.1 in chapter 4; 2. A substantial object represented in a conceptual model using this profile cannot be an instance of more than one ultimate substance sortal. This means that any stereotyped class in this profile used in a class diagram must not include in its general collection more than one substance sortal class. Moreover, a substance sortal must also not include another substance sortal nor a « subkind » in its general collection, i.e., a substance sortal cannot have as a supertype a member of {«kind», «subkind», «quantity», «collective»}. This constraint is a refinement of postulate 4.2 in chapter 4; 3. A Class representing a rigid substantial universal cannot be a subclass of a Class representing an anti-rigid universal. Thus, a substance sortal cannot have as a supertype (must not include in its general collection) a member of {«phase», «role», «roleMixin»}. This constraint is a result of postulate 4.3 in chapter 4.	
Stereotype	**Description**
«kind» **A**	A «kind» represents a *substance sortal* whose instances are *functional complexes*. Examples include instances of Natural Kinds (such as Person, Dog, Tree) and of artifacts (Chair, Car, Television).

Stereotype	Description
«quantity» **A**	A «quantity» represents a substance sortal whose instances are *quantities*. Examples are those stuff universals that are typically referred in natural language by mass general terms (e.g., Gold, Water, Sand, Clay).

Stereotype	Description
«collective» **A**	A «collective» represents a substance sortal whose instances are *collectives*, i.e., they are collections of complexes that have a uniform structure. Examples include a deck of cards, a forest, a group of people, a pile of bricks. Collectives can typically relate to complexes via a constitution relation. For example, a pile of bricks that *constitutes* a wall, a group of people that *constitutes* a football team. In this case, the collectives typically have an extensional principle of identity, in contrast to the complexes they constitute. For instance, The Beatles was in a given world w constituted by the collective {John, Paul, George, Pete} and in another world w' constituted by the collective {John, Paul, George, Ringo}. The replacement of Pete Best by Ringo Star does not alter the identity of the *band*, but creates a numerically different *group of people*.

Constraints

1. A collective can be extensional. In this case the meta-attribute *isExtensional* defined in figure 8.4 is equal to *True*. This means that all its parts are essential and the change (or destruction) of any of its parts terminates the existence of the collective. We use the symbol {extensional} to represent an extensional collective.

Stereotype	Description
«subkind» **A** **A**	A «subkind» is a rigid, relationally independent restriction of a substance sortal that carries the principle of identity supplied by it. An example could be the subkind MalePerson of the kind Person. In general, the stereotype «subkind» can be omitted in conceptual models without loss of clarity.

Constraints

1. Following postulate 4.3, a «subkind» cannot have as a supertype (must not include in its general collection) a member of {«phase», «role», «roleMixin»}.

Stereotype	Description
«phase» **A**	A «phase» represents the phased-sortals *phase,* i.e. *anti-rigid* and *relationally independent* universals defined as part of a partition of a substance sortal. For instance, ⟨Catterpillar, Butterfly⟩ partitions the kind Lepdopterum.

Constraints

1. Phases are anti-rigid universals and, thus, a «phase» cannot appear in a conceptual model as a supertype of a rigid universal (postulate 4.3);

2. The phases $\{P_1 \ldots P_n\}$ that form a phase-partition of a substance sortal S are represented in a class diagram as a disjoint and complete generalization set. In other words, a GeneralizationSet with (isCovering = true) and (isDisjoint = true) is used in a representation mapping as the representation for the ontological concept of a phase-partition.

Stereotype	Description
«role» **A**	A «role» represents a phased-sortal role, i.e. anti-rigid and relationally dependent universal. For instance, the role student is played by an instance of the kind Person.

Constraints

1. Roles anti-rigid universals and, thus, a «role» cannot appear in a conceptual model as a supertype of a rigid universal (postulate 4.3);

2. Let X be a class stereotyped as «role» and r be an association representing X's restriction condition. Then, $\#X.r \geq 1$. This constraint is elaborated in the next section, after additional elements of the UML metamodel are considered.

Metaclass	Description
Mixin Class	*Mixin Class* is an abstract metaclass that represents the general properties of all *mixins*, i.e., *non-sortals* (or dispersive universals). Mixin Class has no concrete syntax. Thus, symbolic representations are defined by each of its concrete subclasses.

Constraints

1. Following postulate 4.4 we have that a class representing a non-sortal universal cannot be a subclass of a class representing a Sortal. As a consequence of this postulate we have that a mixin class cannot have as a supertype (must not include in its general collection) a member of {«kind», «quantity», «collective», «subkind», «phase», «role»}.

2. Moreover, as a consequence of postulate 4.1, a non-sortal cannot have direct instances. Therefore, a mixin class must always be depicted as an abstract class (isAbstract = true).

Stereotype	Description
«category» ***A***	A «category» represents a rigid and relationally independent *mixin*, i.e., a dispersive universal that aggregates essential properties which are common to different substance sortals. For example, the category RationalEntity as a generalization of Person and IntelligentAgent.

Constraints
1. As a consequence of this postulate and of postulate 4.3, we have that a «category» cannot have a «roleMixin» as a supertype. In other words, together with condition 1 for all mixins we have that a «category» can only be subsumed by another «category» or a «mixin».

Stereotype	Description
«roleMixin» ***A***	A «roleMixin» represents an anti-rigid and externally dependent *non-sortal*, i.e., a dispersive universal that aggregates properties which are common to different roles. In includes formal roles such as *whole* and *part*, and *initiatior* and *responder*.
Constraints	
1. Let X be a class stereotyped as «roleMixin» and r be an association representing X's restriction condition. Then, #X.r ≥ 1. The latter constraint is elaborated in the next section, after additional elements of the UML metamodel are introduced.	

Stereotype	Description
«mixin» ***A***	A «mixin» represents properties which are essential to some of its instances and accidental to others (semi-rigidity). An example is the mixin Seatable, which represents a property that can be considered essential to the kinds Chair and Stool, but accidental to Crate, Paper Box or Rock.
Constraints	
1. Due to postulates 4.3, we have that a «mixin» cannot have a «roleMixin» as a supertype.	

8.2 Classifiers and Properties

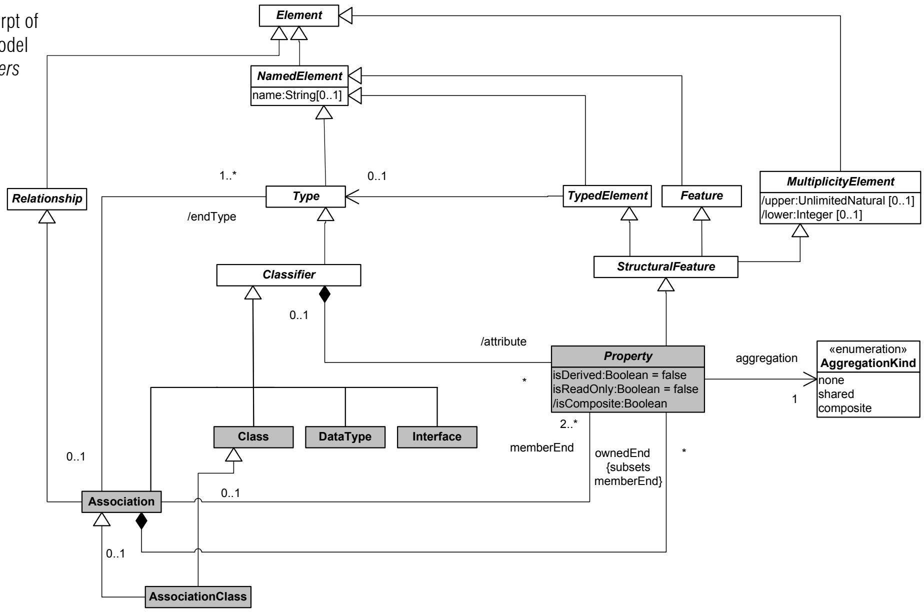
8.2.1 The UML Metamodel

Classifier is an abstract metaclass in the UML metamodel aimed at describing some general aspects that are common to all its subclasses, such as the mechanisms for participating in generalization hierarchies and for properties description.

A classifier can posses a number of *features*. A Feature represents some (behavioral or structural) characteristic that is common to all possible instances of its featuring classifiers. A *structural feature* is special type of feature that specializes both *typed element* and *multiplicity element*. As a consequence, it can impose a number of constraints on the range and type of the values that can be assigned to a given feature of its associated classifiers at instantiation time.

Structural feature is an abstract metaclass. Its most important subclass for the present discussion is *property*. When a property is owned by a *class*, an *interface* or *data type* it represents an *attribute* of that classifier. In this case it relates an instance of the classifier to a value of the type of the attribute. For instance, the property *color* can be defined as an attribute of class *Apple*. This property can have a name (since it is a named element), a type (e.g., the dataType *colorValue* as an enumeration of the colors black, white, blue, yellow, red, green), and a multiplicity (e.g., one-to-one). Therefore, every instance of apple must instantiate exactly one value of type *colorValue* for the property color.

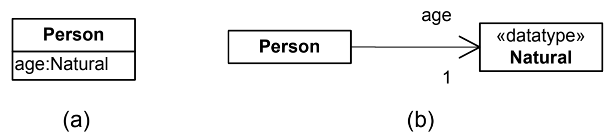
Figure 8-5 Excerpt of the UML metamodel featuring *classifiers* and *Properties*



A property is qualified according to two meta-attributes, namely, *isReadOnly* and *isDerived*. If the first attribute has the value *true* it means that the value of the property cannot be changed once it has been assigned an initial value. The second attribute indicates whether the property is derived, i.e., whether its value can be computed from other information in the model. For instance, a description of the class *Person* can contain the attribute *data of birth*, and the derivable attribute *age*, which is derived from the former for each given circumstance. The default value of both meta-attributes is *false*.

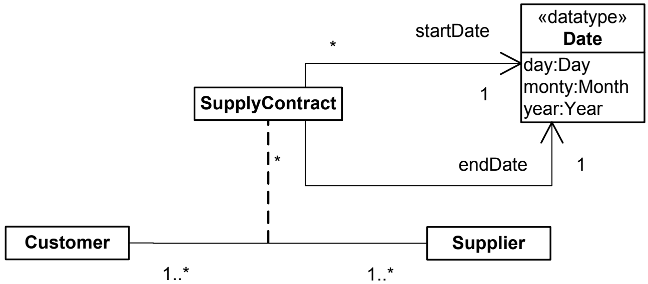
When related to an association, a property represents an *association end* of that association. An association describes a set of tuples whose values refer to typed instances. An instance of an association is called a link. Every association must have at least two association ends represented by properties, each of which is connected to the type of that property, which becomes therefore the type of the association end. Associations can be type reflexive, i.e., more than one association end of the same association may have the same type. In figure 8.5, *association* has two different relations with the metaclass property: one via *memberEnd* and another via *ownedEnd*. As represented in this model, the extension of the second one is a subset of that of the first. The first relation represents the association ends that are in fact owned by one of the associated classifiers. These are termed *navigable ends* and they are semantically equivalent to attributes, thus, merely representing an alternative notation. In contrast, in the association connected to ownedEnd, the properties represent type and cardinality restrictions for the possible tuples that instantiate that association. Figure 8.6 below exemplifies a navigable end use as an alternative notation for a classifier's attribute.

Figure 8-6 Navigable End as an alternative notation for representing an *Attribute*



An association may be refined to have its own set of attributes, i.e., attributes that do not belong to any of the connected classifiers but rather to the association itself. Such an association is called an *association class*. An association class is both a kind of association and a kind of a class. As an association it can connect a set of classifiers; as class it can have its own attributes and participate in other associations. Graphically, it is shown in a class diagram as a class symbol attached to the association path by a dashed line. Despite of being graphically distinct, the association path and the association class symbol represent the same underlying model element, which has a single name. An example of an association class is depicted in figure 8.7 below.

Figure 8-7 Exemple of an *Association Class* and its owned attributes



In the example above, *Date* is an illustration of a *data type*. According to the UML specification *"a data type is a type whose values have no identity (i.e., they are pure values)…so two occurrences of the same value cannot be differentiated. A DataType defines a kind of classifier in which operations are all pure functions (i.e., they can return data values but they cannot change [them). They are usually] used for specification of the type of an attribute. [Additionally, DataTypes] may have an algebra and operations defined outside of UML, for example, mathematically. [Finally, as any classifier,] a DataType may also contain attributes to support the modeling of structured data types"*.

The remaining type of classifier to be discussed is the metaclass *Interface*. An interface is a declaration of a coherent set of features and obligations. It can be seen as a kind of contract that partitions and characterizes groups of properties that must be fulfilled by any instance of a classifier that implements that interface. The obligations that may be associated with an interface are in the form of various kinds of constraints (such as pre- and postconditions) or protocol specifications, which may impose ordering restrictions on interactions through the interface. Since interfaces are merely declarations, they are non instantiable model elements. That is to say that an interface cannot have direct instances, but only via their implementing classifiers. Figure 8.8 below shows an interface (Seatable), which is realized by two different classes: Crate and Stool. By implementing this interface, these classes commit to provide mechanisms for *"maintaining the information corresponding to the type and multiplicity of the property and facilitate retrieval and modification of that information"*.

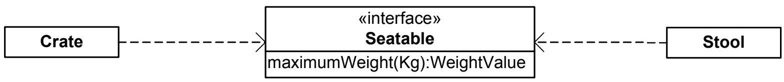


Figure 8-8 Example of an *Interface* and its implementing classifiers

8.2.2 Ontological Interpretation and Re-Design

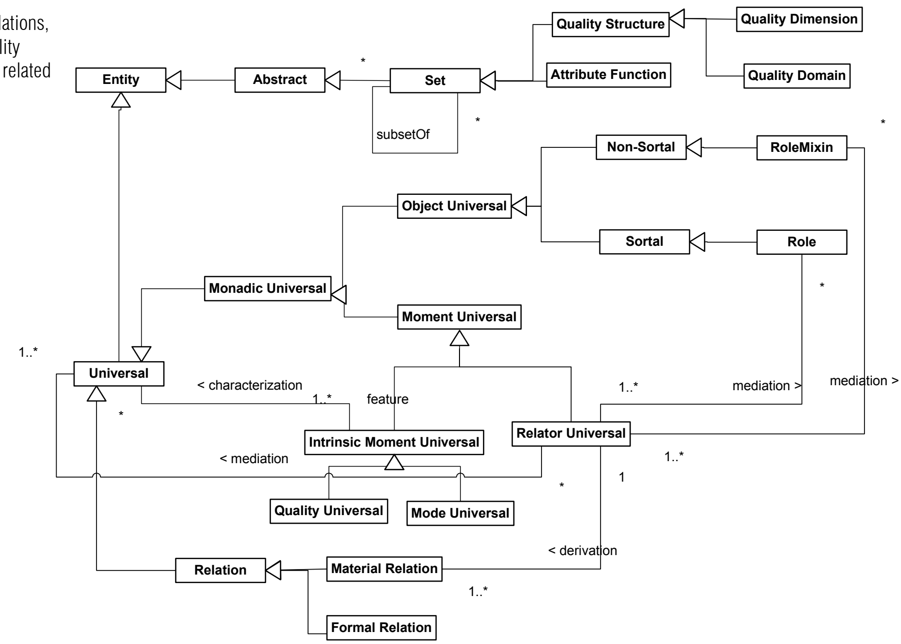
In section 8.1.2, in an interpretation mapping from the UML metamodel to our foundational ontology, we have mapped the UML class to the ontological category of *monadic universals*. However, we have focused our discussion on the subcategory of *substantial universals*. The other kind of monadic universal prescribed by our theory is *Moment Universal*.

The category of moment universal along with other ontological categories which are relevant for the discussion carried out on this section are depicted in figure 8.9 below.

Moment universals can be differentiated from their substantial counterparts as follows: a moment universal has the characteristic that all its instances inheres in and, thus, is externally existentially dependent on some other individual.

Within the category of moment universals we can use a different criterion to differentiate among its subcategories. An Intrinsic moment universal is a universal which instances are existentially dependent of a single entity (named its bearer). A Relator universal, conversely, is one which instances are existentially dependent of a plurality of entities. Both types of entities are fundamental from an ontological and conceptual point of view. Intrinsic moments constitute a foundation for *attributes* and *comparative formal relations*, but also *weak entities* and *qua individuals*. Qua individuals, in turn, constitute a foundation for *material roles*. Relators are the foundation of *material relations*.

Figure 8-9 Relations, Moments, Quality Structures and related categories



In a representation mapping from our reference ontology to the UML metamodel, we realize that there are no constructs that represent the ontological categories of *mode universals* and *relator universals* in the UML metamodel. Once more, there are ontological concepts prescribed by our theory that are not represented by any modeling construct in the language, i.e., another case of ***construct incompleteness*** at the modeling language level.

According to postulate 6.1, universals of the domain should be represented in a conceptual model as Classes. We therefore propose an extension of the original UML metamodel of figure 8.5 to incorporate these entities. Additionally, in a profile implementing this extended metamodel, we include two other stereotyped classes (base class UML class) to represent these entities. Mode universals are represented via a class decorated with the «mode» stereotype. Relator Universals are represented via the stereotype «relator».

Quality universals are typically not represented in a conceptual model explicitly but via *attribute functions* that map each of their instances to points in a quality structure (see discussion on chapter 6). For example, suppose we have the universal Apple (a substantial universal) whose instances exemplify the universal Weight. We say in this case that the quality universal Weight *characterizes* the substantial universal Apple. Thus, for an arbitrary instance *x* of Apple there is a quality *w* (instance of the quality universal Weight) that inheres in *x*. Associated with the universal Weight, and in the context of a given measurement system (e.g., the human perceptual system), there is a quality dimension *weightValue*, which is a set isomorphic to the half line of positive integers obeying the same ordering structure. Quality structures are taken here to be theoretical abstract entities modeled as sets. In this case, we can define an *attribute function* (another abstract theoretical entity) *weight(Kg)*, which maps each instance of apple (and in particular *x*) onto a point in a quality dimension, i.e., its *qualia*.

Following principle 6.2 (section 6.3.2), attribute functions are therefore the ontological interpretation of UML attributes, i.e., UML Properties which are owned by a given classifier. That is, an attribute of a class C representing a universal U is interpreted as an attribute function derived from the *elementary specification* of the universal U.

As any property, a UML attribute is a typed element and, thus, it is associated to Type. Type constrains the sort of entities that can be assigned to slots representing that attribute in instances of their owning classifier. Since Classifier is a specialization of Type, we have that both Classes and Datatypes can be the associated types of an UML attribute. In other words, an attribute represents both an attribute function and a sort of a *relational image function*[69] that, for example, in binary relation *ownership* between the classifiers Person and Car, maps a particular Person p to all instances of Car that are associated with p via this relation (i.e., all cars owned by p). From a software design and implementation point of view, an attribute represents a method implemented by the owning class, and the type of the attribute

[69]In LINGO (Guizzardi & Falbo & Gonçalves, 2002), for instance, a relational image function is formally defined as follows: Let R be a binary relation defined for the two sets X and Y. The function **Im** with signature $\mathbf{Im(\_,\_): X \times (X \Leftrightarrow Y) \rightarrow \wp(Y)}$ is defined as $\mathbf{Im(x,R) = \{y \mid (x,y) \in R\}}$.

represents the returning type of that method. However, from a conceptual point of view, in the UML metamodel an attribute stands both for a monadic (instrinsic) and for a relational property and, thus, it can be considered a case of ***construct overload***. To eliminate this problem, we prescribe that attributes should only be used to represent attribute functions. As consequence, their associated types should be restricted to DataTypes only. Attributes whose associated types are classes (known as *mappings* in the object-orientation literature) (see for instance Bock & Odell, 1997b) are more important from a design and implementation point of view than from a conceptual one. We propose here a different treatment of relational properties, which is discussed latter in this section.

The DataType associated with an attribute A of class C is the representation of the quality structure that is the co-domain of the attribute function represented by A (postulate 6.2). In other words, a quality structure is the ontological interpretation of the UML DataType construct. Moreover, we have that a multidimensional quality structure (quality domain) is the ontological interpretation of the so-called *Structured DataTypes* (postulate 6.3).

Quality domains are composed of multiple integral dimensions. This means that the value of one dimension cannot be represented without representing the values of others. The fields of a datatype representing a quality domain QD represent each of its integral quality dimensions. Alternatively we can say that each field of a datatype should always be interpreted as representing one of the integral dimensions of the QD represented by the datatype. The constructor method of the datatype representing a quality domain must reinforce that its tuples always have values for all the integral dimensions. Finally, an algebra can be defined for a DataType so that the relations constraining and informing the geometry of represented quality dimensions are also suitably characterized.

UML offers an alternative notation for the representation of attributes, namely, *navigable end names*. That is, the same ontological concept (*attribute function*) is represented in the language via more than one construct, which characterizes ***construct redundancy***. This situation could be justified from a pragmatic point of view if navigable ends were used to model only structured DataTypes (as in figure 8.7), and if the textual notation for attributes were only used to model the simple ones (e.g., figure 8.6.a). However, as exemplified in figure 8.6, there is no constraint on using both notations for both purposes. To eliminate the potential ambiguity of this situation, we propose to use navigable ends to represent only attribute functions whose co-domains are *multidimensional quality structures* (quality domains). Conversely, those functions whose co-domains are quality dimensions should only be represented by the attributes textual notation.

According to the UML specification, *"a dataType is a type whose values have no identity (i.e., they are pure values)…so two occurrences of the same value cannot be differentiated."* What exactly is meant by *"have no identity"* here? Take for instance, the DataType Integer. It is clear that the member values of this DataType have identity in the usual sense of the term, i.e., identity statements such as $(1=1)$ and $\neg(1=2)$ are not only meaningful but also determinate. However, it is also true that two occurrences of the integer 1 refer to exactly the same entity. So, a better statement would be that different occurrences of the same DataType value do not have a separate identity. A universal U (represented by a UML Class) is multiply exemplified in each of its numerically distinct instances. A DataType is an abstract entity that collects other abstract entities (*"pure values"*) that can be multiply referred. A DataType is therefore not a multiply instantiated universal but an individual (set) with other individuals as members.

The ontological category of *relations* can be mapped in a representation mapping onto the UML concept of *associations*. Relations can be *material* or *formal*. The latter in turn can be subdivided in basic formal relations (internal relations) and comparative formal relations. Alternatively we can classify relations in basic relations and derived relations. The latter includes both comparative formal relations (derived from the relations among certain qualities) and material relations (derived from relator universals).

Since class diagrams only represent universals, the only basic formal relations among the ones we have considered that should have a representation in these models are the relations of *characterization*, *mediation* and *derivation*. These concepts have no representation in the UML metamodel, which characterizes another case of ***construct incompleteness***. Once more we propose an extension of the UML metamodel of figure 8.5 to include associations (i.e., extensions of the UML association metaclass) to represent these ontological distinctions. Moreover, in a profile implementing this extended metamodel, we include stereotyped associations (i.e., the base class is the UML association) to represent these newly introduced formal relations. The relations of characterization and mediation are represented by the stereotyped associations «characterization» and «mediation», respectively. The relation of derivation has a special notation discussed later in this section.

Object universals are characterized by intrinsic moment universals. As previously discussed, in the modelling profile proposed in this section, quality universals are not explicit represented in conceptual specifications. In contrast, mode universals are represented via a class decorated with the «mode» stereotype. The formal relation of characterization that takes place between a mode universal and the universal it characterizes is explicitly represented by an association stereotyped as «characterization».

Associations stereotyped as «characterization» must have in one of its association ends a class stereotyped as «mode» representing the characterizing mode universal. Likewise, every «mode» must be an association end for one «characterization» relation.

The *characterization* relation between a mode universal and the universal it characterizes is mapped at the instance level onto an *inherence* relation between the corresponding individual modes and their bearer objects. That means that every instance *m* of a class *M* stereotyped as «mode» is existentially dependent of an individual *c*, which is an instance of the class *C* related to *M* via the «characterization» relation. Inherence has the following characteristics: (a) it is a sort of existential dependence relation; (b) it is a binary formal relation; (c) it is a functional relation. These three characteristics impose the following metamodel constraints on the «characterization» construct:

1. by (a) and (c), the association end connected to the characterized universal must have the cardinality constraints of one and exactly one (lower = upper = 1);
2. by (a), the association end connected to the characterized universal must have the meta-attribute (isreadOnly = true);
3. «characterization» associations are always binary associations (#memberEnd = #ownedEnd = 2).

As discussed in chapter 6, there are no optional properties. For this reason, the «characterization» relation must also have a minimum cardinality of *one* on the association end connected to the mode universal. For example, if the universal *Patient* is characterized by the mode universal *Symptom*, then every instance *x* of patient must bear an instance *y* of symptom. As a consequence we have the following additional constraint:

4. In a «characterization» relation, the association end connected to the characterizing quality universal must have the minimum cardinality constraints of one (lower = 1).

An analogous case can be made for attributes. For example, since every apple bears a color quality, and every color quality has a value on the color domain, ergo, the corresponding *attribute function* is a total function. Thus,

5. A property owned by a classifier (representing an attribute of that class) must have the minimum cardinality constraints of one (lower = 1).

Finally, this constraint also holds for DataTypes. As previously mentioned, the fields of a structured DataType represent the integral quality dimensions that compose the quality domain represented by this DataType. By its very definition, a quale in a quality domain must have values for all integral dimensions that compose that domain.

As discussed in section 6.3.3, the association class construct in UML exemplifies a case of ***construct overload***, since "an associaton class can have as instances either (a) a n-tuple of entities which classifiers are endpoints of the association; (b) a n+1-tuple containing the entities which classifiers are endpoints of the association plus an instance of the objectified association itself" (Breu et al., 1997). This is to say that an association class can be interpreted both as a *relational universal* and as a *factual universal*. In addition to that, since the *"instance of the objectified association itself"* is supposed to be an object identifier for the n-tuple, one cannot represent cases in which the same relator mediates multiple occurrences of the same n-tuple. Therefore, association classes on one hand represent a case of construct overload, on the other land, allow for a case of construct incompleteness at the instance level. We propose, therefore, to disallow the use of association classes in UML for the purpose of conceptual modeling.

In contrast, we propose to represent relational properties explicitly. We use the stereotype «relator» to represent the ontological category of relator universals. Relator universals can induce material relations. For instance, if there is a particular *Marriage m* connecting two persons *a* and *b*, we then can say that these persons stand up in a *married-to* relation, or that *derived-from([a,b],m)* holds. We say that a *derivation* relation holds between two universals F and G iff F is material relation (relational universal); G is a relator universal; For every instance *x* of F there is an instance y of G such that *derived-from(x,y)* holds.

A material relation induced by a relator universal R is represented by a UML association stereotyped as «material» (UML base class *association*). The basic formal relation *derivation* is represented by a dashed line with a black circle in one of the ends. A derivation relation is a specialized type of relationship between the stereotyped association representing the derived «material» association and the stereotyped class representing the founding «relator» universal. The black circle represents the role of foundation of the relator universal side. Every «material» association must be the association end of exactly one derivation relation.

The use of a different style of concrete syntax for the derivation relation than the one adopted for the other formal relations could in principle introduce some pragmatic inefficient, since the use of different syntax could allow for the misleading *implicature* (see chapter 2) that the represented entity belongs to a different ontological category. However, we believe that the drawback of this choice could be compensated by the benefit of using a syntax which is already familiar to UML users.

Derivation is a (total) functional relation and also a type of existential dependency. For this reason we have that every n-tuple *x* instance of a material relation F is derived from exactly one relator individual r and existentially dependent on the latter. As a consequence, the black circle end of the derivation relation must obey the following constraints: (a) have the cardinality constraints of one and exactly one (lower = upper = 1); (b) have the (isreadOnly = true).

The formal *mediation* relation between a relator universal and the universals it mediates is mapped onto the instance level to a *mediation* (*m*) relation between the corresponding individuals. That means that every instance r of a class R stereotyped as «relator» is existentially dependent of the individuals $c_1 \ldots c_n$ instances of the classes $C_1 \ldots C_2$ connected to R via the «mediation» relation. Mediation (as much as inherence) is a type of (binary) existential dependence relation, which imposes the following constraints on its representation:

1. the association ends owned by each of the mediated universals must have the minimum cardinality constraints of at least one (lower = 1);
2. the association ends owned by of the mediated universals must have the (isreadOnly = true);
3. «mediation» associations are always binary associations (#memberEnd = #ownedEnd = 2).

Moreover, since a relator is dependent (mediates) on at least two numerically distinct entities, we have the following additional constraint:

1. Let R be a relator universal and let $\{C_1 \ldots C_2\}$ be a set of universals mediated by R (related to R via a «mediation» relation). Finally, let $lower_{Ci}$ be the value of the minimum cardinality constraint of the association end connected to C_i in the «mediation» relation. Then, we have that

$$(\sum_{i=1}^{n} lower_{Ci}) \geq 2 \ .$$

Let us suppose that R is a relator universal and that a mediation relation holds between R and the universals F and G (see figure 8.10). This means that for every instance r::R there is at least one f::F and one g::F such that *m(r,f)* and *m(r,g)*. Now, let H be the material relation between F and G derived from R. Then we have that for every r::R there is at least one pair [f,g]::H such that *derived-from([f,g],r)* holds. As a direct consequence, the association end connected to H in the *derivation* relation must have the minimum cardinality constraint of one (lower = 1).

Still on figure 8.10, from the cardinality constraints of the two «mediation» relations we can derive the maximum cardinality of the derivation relation (on the material relation end) and the cardinality constrains on both association ends of the material relation itself. For instance, the upper constraint δ on the end connected to G in the H relation is the result of $(d \times h)$; the upper constraint β in the end connected to F is the result of $(f \times b)$. The upper constraint ϕ in the end H of the derivation relation is the result of $(b \times h)$. Likewise, we can calculate the derived minimum cardinality constraints in the following manner: $\gamma = c \times g$; $\alpha = e \times a$, and $\varepsilon = a \times g$.

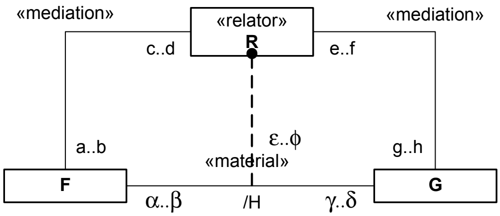


Figure 8-10 Material Relations and their founding relators (the cardinality constraints of the derived relation and the derivation relation itself can be calculated from the corresponding mediation relations involving the founding relators)

Two alternative versions of a concrete example of this situation are depicted in figures 8.11.a and 8.11.b below. However, due to the lack of expressivity of the traditional UML association notation, these two models seem to convey the same information (from the perspective of the material relation *supervised-by*), although they describe completely different conceptualizations. As discussed in section 6.3.3, the benefits of explicitly representing relator universals instead of merely representing material relations, becomes even more evident in n-ary relations with n > 2.

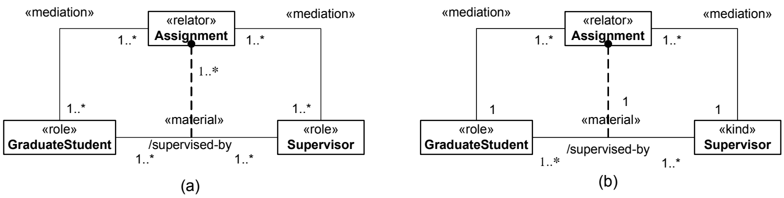


Figure 8-11 Examplification of how relators can disambiguate two conceptualizations that in the standard UML notation would have the same interpretation

Once more we should highlight that the relator individual is the actual instantiation of the corresponding relational property (the objectified relation). Material relations stand merely for the facts derived from the relator individual and its mediating entities. Therefore, we claim that the representation of the relators of material relations must have primacy over the representation of the material relations themselves. In other words, the representation of «material» relations can be omitted but whenever a «material» is represented it must be connected to an association end of a derivation relation.

We use the stereotype «formal» to represent domain formal relations. Comparative formal relations and material relations are derived relations. Whilst the former are derived from intrinsic properties of the related entities, the latter are derived from relational properties and their mediating entities. Therefore, we prescribe that «material» must have the meta-attribute (isDerived = true). Analogously, we use the meta-attribute (isDerived = true) to represent formal relations which are not *internal relations*, i.e., which are comparative.

Once more we should emphasize that there are no optional properties, e.g., there is nothing which has the optional property of *being married*. For this reason, «mediation» relations must have a minimum cardinality of one on the association end connected to the relator universal. For example, in figure 8.11, since the universal Supervisor is mediated by the relator universal Assignment, then every instance *x* of supervisor must have a *m(y,x)* relation with an instance *y* of assignment. A direct consequence of this rule is that the minimum cardinality constraints on both ends of the *supervised-by* material relation (derived from Assignment) must also have a value of at least one. The same applies to the association end connected to the material relation (*supervised-by*) in the derivation relation.

This is especially evident when the mediated entities are *Role* classifiers. Take, for instance, the examples of figure 8.11. As discussed in chapter 4, a role is a restriction of kind by means of a relational restriction condition. For instance, *a supervisor is a Person who supervises graduate students*. Thus, being related to an assignment is part of the very definition of the universal supervisor (in this conceptualization). As we have elaborated in chapter 7, for *x* to be a role individual, then there must a *role qua individual y* such that *i*(y,x). However, a role qua individual must be part of a relator. In fact, if *x* is an individual such that there is a qua individual *y* which inheres in *x* and which is part of a relator *z* then x is mediated by *z* (see chapter 6). As a consequence, we have that for every role individual *x* there must be a relator individual *z* that mediates *x*.

We can thus define the following constraints:

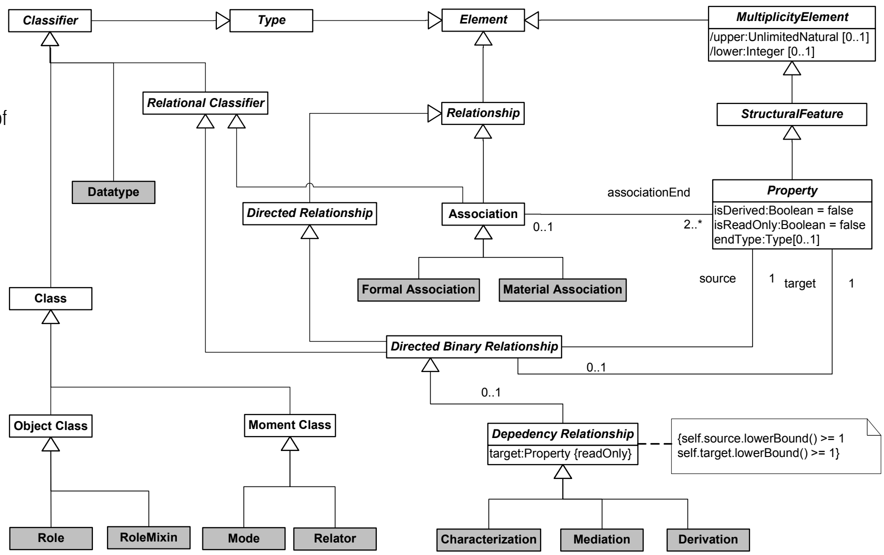
1. In a «mediation» relation, the association end connected to the relator universal must have the minimum cardinality constraints of one (lower = 1);
2. Every *relationally dependent* entity (role and role mixins) must be connected to an association end of a «mediation» relation.

Finally, in figure 8.5, we have a last subtype of classifier, namely, the meta-construct Interface. According to the UML specification, an interface is a declaration of a coherent set of features and obligations. It can be seen as a kind of contract that partitions and characterizes groups of properties which must be fulfilled by any instance of a classifier that implements that interface. In an interpretation mapping from the UML metamodel to the ontology of figure 8.9, an interface qualifies as a case of **construct excess**. This means that, being merely a design and implementation construct, there is no category in the conceptual modeling ontology proposed here that serve as the ontological interpretation for the UML interface. For this reason, we propose that the use of this construct should be disallowed in an ontologically well-founded version of UML that is suitable for conceptual modeling.

In this respect we therefore strongly disagree with the works of (Steimann, 2000a,b, 2001) that proposes that the ontological concept of roles can be the interpretation for the UML interface construct. As recognized by the UML specification itself, interfaces are merely declarations and, hence, *non-instantiable* model elements. Interfaces can be implemented by classes that obey different principles of identity. Thus, as a classifier, an interface can only be interpreted as a non-sortal. So, at most, it would be fair to propose the mapping from interface to the ontological category of *role mixin*. However, unlike role mixins, interfaces are neither necessarily relationally dependent nor anti-rigid. In summary, there is absolutely no ground to propose a (representation) mapping from an anti-rigid, relationally dependent sortal to a non-sortal which is not necessarily any of these things.

In figure 8.12, we present a revision of the UML metamodel of figure 8.5 that faithfully represents the ontological distinctions proposed in this section. All the different dependency relations discussed in this section, namely, *characterization*, *mediation* and *derivation* have in common that they are all directed binary relations. By distinguishing between domain associations and formal relations (represented by the *directed binary relationship* metaclass) in the extended metamodel we help to prevent the construction of *unintended models*, i.e., models that portrait state of affairs that are excluded by the underlying conceptualization (see chapter 3). In particular, we exclude the possibility of constructing models containing dependency relations with more than two association ends. In addition, all dependency relations on this meta-model stand for relations of *existential dependence* (i.e., *rigid specific dependence*). This is to say that, at the instance level, the dependent instance must be related to one and the same depended entity one in all possible worlds. As a consequence, the *target* meta-attribute of Dependency Relations in this meta-model is a *readOnly* meta-attribute, i.e., once assigned it cannot be modified. Finally, it is important to emphasize that this metamodel omits those elements which have no ontological interpretation (*interface*), or which are ill-defined from an ontological point of view (*association class*).

Figure 8-12 Revised Fragment of the UML 2.0 metamodel according to the ontological categories of figure 8.9



The UML metamodel elements proposed in this revised model together with their characterizing specific syntactical constraints are implemented in a modelling profile. This profile, which extends the one of table 8.1, is summarized in tables 8.2 and 8.3 below.

Table 8-2 Extension to the UML modelling profile of table 8.1 that implements the revised metamodel of figure 8.12

Stereotype	Base Class	Description
«role» **A**	Class	Refinement of the definition of the «role» stereotype presented in table 8.1.
Constraints		
1. <u>Refinement of condition 2 of «role» stereotype on table 8.1</u>: Every «role» class must be connected to an association end of a «mediation» relation;		
Stereotype	**Base Class**	**Description**
«roleMixin» ***A***	Class	Refinement of the definition of the «roleMixin» stereotype presented in table 8.1.
Constraints		
1. <u>Refinement of condition 1 «roleMixin» stereotype on table 8.1:</u> Every «roleMixin» class must be connected to an association end of a «mediation» relation;		

Stereotype	Base Class	Description
«mode» **A**	Class	A «mode» universal is an intrinsic moment universal. Every instance of mode universal is existentially dependent of exactly one entity. Examples include skills, thoughts, beliefs, intentions, symptoms, private goals.

Constraints

1. Every «mode» must be (directly or indirectly) connected to an association end of at least one «characterization» relation;

Stereotype	Base Class	Description
«relator» **A**	Class	A «relator» universal is a relational moment universal. Every instance of relator universal is existentially dependent of at least two distinct entities. Relators are the instantiation of relational properties such as marriages, kisses, handshakes, commitments, and purchases.

Constraints

1. Every «relator» must be (directly or indirectly) connected to an association end on at least one « mediation » relation;

2. Let R be a relator universal and let $\{C_1 \ldots C_2\}$ be a set of universals mediated by R (related to R via a «mediation» relation). Finally, let $lower_{Ci}$ be the value of the minimum cardinality constraint of the association end connected to C_i in the «mediation» relation. Then, we have that

$$(\sum_{i=1}^{n} lower_{Ci}) \geq 2 .$$

Stereotype	Base Class	Description
«mediation»	Dependency Relationship	A «mediation» is a formal relation that takes place between a relator universal and the endurant universal(s) it mediates. For example, the universal *Marriage* mediates the role universals Husband and Wife, the universal *Enrollment* mediates Student and University, and the universal *Covalent Bond* mediates the universal Atom.

Constraints
1. An association stereotyped as «mediation» must have in its source association end a class stereotyped as «relator» representing the corresponding relator universal (self.source.oclIsTypeOf(Relator)=true); 2. The association end connected to the mediated universal must have the minimum cardinality constraints of at least one (self.target.lower ≥ 1); 3. The association end connected to the mediated universal must have the property (self.target.isreadOnly = true); 4. The association end connected to the relator universal must have the minimum cardinality constraints of at least one (self.source.lower ≥ 1). 5. «mediation» associations are always binary associations.

Stereotype	Base Class	Description
«characterization»	Dependency Relationship	A «characterization» is a formal relation that takes place between a mode universal and the endurant universal this mode universal characterizes. For example, the universals *Private Goal* and *Capability* characterize the universal *Agent.*

Constraints
1. An association stereotyped as «characterization» must have in its source association end a class stereotyped as «mode» representing the characterizing mode universal (self.source.oclIsTypeOf(Mode)=true); 2. The association end connected to the characterized universal must have the cardinality constraints of one and exactly one (self.target.lower = 1 and self.target.upper = 1); 3. The association end connected to the characterizing quality universal (source association end) must have the minimum cardinality constraints of one (self.source.lower ≥ 1); 4. The association end connected to the characterized universal must have the property (self.target.isreadOnly = true); 5. « characterization » associations are always binary associations.

Stereotype	Base Class	Description
Derivation Relation	Dependency Relationship	A derivation relation represents the formal relation of derivation that takes place between a material relation and the relator universal this material relation is derived from. Examples include the material relation *married-to*, which is derived from the relator universal *Marriage*, the material relation *kissed-by*, derived from the relator universal *Kiss*, and the material relation *purchases-from,* derived from the relator universal *Purchase*.

Constraints

1. A derivation relation must have one of its association ends connected to a relator universal (the black circle end) and the other one connected to a material relation (self.target.oclIsTypeOf(Relator)=true, self.source.oclIsTypeOf(Material Association)=true);

2. derivation associations are always binary associations;

3. The black circle end of the derivation relation must have the cardinality constraints of one and exactly one (self.target.lower = 1 and self.target.upper = 1);

4. The black circle end of the derivation relation must have the property (self.target.isreadOnly = true);

5. The cardinality constraints of the association end connected to the material relation in a derivation relation are a product of the cardinality constraints of the «mediation» relations of the relator universal that this material relation derives from. This is done in the manner previously shown in this section. However, since «mediation» relations require a minimum cardinality of one on both of its association ends, then the minimum cardinality on the material relation end of a derivation relation must also be ≥ 1 (self.source.lower ≥ 1).

Stereotype	Base Class	Description
«material»	Association	A «material» association represents a material relation, i.e., a relational universal which is induced by a relator universal. Examples include *student studies in university*, *patient is treated in medical unit*, *person is married to person*.

Constraints		
1. Every «material» association must be connected to the association end of exactly one derivation relation; 2. The cardinality constraints of the association ends of a material relation are derived from the cardinality constraints of the «mediation» relations of the relator universal that this material relation is derived from. This is done in the manner shown in this section. However, since «mediation» relations require a minimum cardinality of one on both of its association ends, then the minimum cardinality constraint on each end of the derived material relation must also be ≥ 1 ; 3. Every «material» association must have the property (isDerived = true).		
Stereotype	**Base Class**	**Description**
«formal»	Association	A «formal» association represents a formal relation, i.e., either a comparative relation (derived from intrinsic properties of the relating entities), or an internal relation. Examples include Person <u>*is older than*</u> Person, an Atom <u>*is heavier than*</u> another atom. We use UML symbolic representation for *derived* relations (/) to represent comparative relations.

Table 8-3 A constraint on properties representing attributes which is incorporated in this modelling profile

Metaclass	Description
Property	An attribute in the UML metamodel is a property owned by a classifier. Attributes are used in this profile to represent attribute functions derived for quality universals. Examples are the attributes color, age, and startingDate.
Constraints	
1. A property owned by a classifier (representing an attribute of that classifier) must have the minimum cardinality constraints of one (self.lower ≥ 1).	

Finally, in the UML 2.0, it is possible to apply a profile to a model, and indicate whether the model only have stereotypes of the profile or whether elements of the reference metamodel (e.g., the metamodel of figure 8.5) are still allowed. This option can be used in models adopting the profile proposed in this section to disallow the use of interfaces and association classes.

8.3 Aggregation and Composition

8.3.1 The UML Metamodel

We refer once again to the metamodel fragment depicted in figure 8.5, but now with the focus on the representation of part-whole relations in UML.

A part-whole relation is specified in UML in the following manner: *"An association may represent an aggregation; that is, a whole/part relationship. In this case, the association-end attached to the whole element is designated, and the other association-end of the association represents the parts of the aggregation. Only binary associations may be aggregations."*

Every member end of an association (being a property) has an attribute named *aggregation*. This attribute indicates whether that association represents a part-whole relation and, if so, which kind of part-whole relation. The possible values that this attribute can assume are specified in the enumeration *AggregationKind*. They are:

1. **none:** indicates that the property has no aggregation, i.e., that the association that has this property as a member end does not represent a part-whole relation. This is the default value for this property;

2. **shared:** indicates that the property has a shared aggregation and, consequently, that the corresponding association represents a shareable part-whole relation;

3. **composite:** indicates that the property has a non-shared aggregation and, consequently, that the corresponding association represents a non-shareable part-whole relation (composition). Composition is represented by the *isComposite* meta-attribute on the part end of the association being set to true.

The UML specification defines the following semantics for composition: "*Composite aggregation is a strong form of aggregation, which requires that a part* *instance be included in at most one composite at a time and that the composite object has sole responsibility for the disposition of its parts. This means that the composite object is responsible for the creation and destruction of the parts. In implementation terms, it is responsible for their memory allocation. If a composite object is destroyed, it must destroy all of its parts. It may remove a part and give it to another composite object, which then assumes responsibility for it. If the multiplicity from a part to composite is zero-to-one, the composite may remove the part, and the part may assume responsibility for itself, otherwise it may not live apart from a composite."*

Conversely, a shareable aggregation is defined as follows: "*A shareable aggregation denotes weak ownership; that is, the part may be included in several aggregates and its owner may also change over time. However, the semantics of a shareable aggregation does not imply deletion of the parts when an aggregate referencing it is deleted."*

In terms of primary meta-properties, *"both kinds of aggregations define a transitive, antisymmetry relationship; that is, the instances form a directed, non-cyclic graph. Composition instances form a strict tree (or rather a forest)."*

A part-whole relation is by default expressed as an aggregation in the UML. Otherwise, if the parts in the part-whole relation are non-shareable, then the relation is expressed as a composition (black-diamond). The use of composition implies the maximum cardinality of 1 w.r.t. the whole. An example of part-whole relation with shareable parts is the relation between researchers and research groups (researchers can be part of several research groups) depicted in figure 8.13.a. Conversely, the relation between a person and her heart (e.g., figure 8.13.b) is an instance of a non-shareable part-whole relation[70]. Finally, figure 8.13.c depicts a case of non-shareable parthood with optional wholes.

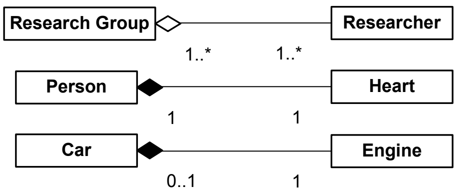


Figure 8-13 Part-Whole relations with: (a) shareable part; (b) non-shareable part and mandatory whole; (c) mandatory part and optional whole

[70] We assume here a conceptualization in which a human heart cannot be shared by more than one human body, thus, excluding the case of Siamese Twins. Once more, the example is used here for illustration purposes only and not to defend a particular ontological commitment.

8.3.2 Ontological Interpretation and Re-Design

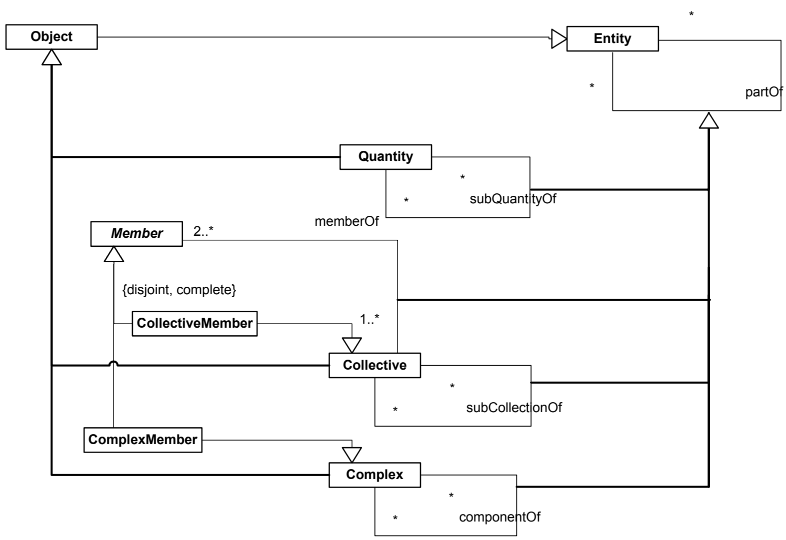
In spite of being a small fragment of the UML metamodel, this part relative to the modeling of part-whole relations poses a number of serious difficulties from a conceptual modelling point of view.

To start with, UML defines only one general sort of parthood relation, which can, in turn, be qualified in two types, namely aggregation and composition. This is because the UML metamodel considers only one equally general notion of objects, i.e., it is insensible to the ontological distinctions among types of substantial individuals.

As discussed extensively in chapter 5, in our ontology we consider four sorts of conceptual parthood relations, based on the type of entities they relate, namely, (a) **subQuantityOf**, which relates individuals that are *quantities*; (b) **subCollectionOf**, which relates individuals that are *collectives*; (c) **memberOf**, which relates individuals that are *functional complexes* or *collectives* as parts of individuals that are *collectives*; (d) **componentOf**, which relates individuals that are *functional complexes*.

These different sorts of conceptual parthood relations are depicted in Figure 8.14 below. These different relations between *objects* are specializations of a more abstract *partOf* relation defined to hold between *entities* in general.

Figure 8-14 Different types of meronymic relations according to the ontological category of their relata



In a first analysis, the general notion of aggregation in UML can be thought to represent the *partOf* relation in this ontology. In UML, the general aggregation is an *asymmetric* and *transitive* relation, but it is not said to be irreflexive. This is because, in contrast to the *partOf* relation, aggregation is a relation between types, not between tokens. Therefore, it is, in principle, conceivable that we have a relation which is type-reflexive but which is still irreflexive at the instance level (i.e., there is no instance that bears this relation to itself).

The *partOf* relation is an *anti-symmetric* and *non-transitive* relation. Non-transitivity means that transitivity holds for certain cases but not for others. The relation of partOf must be deemed non-transitive since: two of its subclasses are transitive (subQuantityOf and subCollectionOf), one is intransitive (memberOf), and one is itself non-transitive (componentOf). In addition, this relation obeys the *irreflexivity* axiom and *weak supplementation principle* (i.e., if x is a part of y then there must be a z disjoint of x, which also part of y - see chapter 5).

We conclude then we cannot identify elements of the UML metamodel that can be used to represent neither the *partOf* relation nor any of its subclasses. Once more we have a case of ***construct incompleteness***. We therefore propose an extension of the UML 2.0 metamodel depicted in figure 8.5 to include metaclasses that represent each of these concrete types of conceptual parthood relations.

By far, the most commonly used of these relations in conceptual modeling is the *componentOf* relation. This is because *functional complexes* are also the most commonly represented entities in conceptual specifications. For this reason, we use the standard UML symbolic notation for aggregation to represent this relation.

As any *partOf* relation, *componentOf* is an irreflexive and anti-symmetric relation, which obeys the weak supplementation principle. As a result of this last principle, we have the following constraint: Let U be a universal whose instances are functional complexes and let $\{C_1 \ldots C_2\}$ be a set of universals related to U via aggregation relations. Finally, let $lower_{Ci}$ be the value of the minimum cardinality constraint of the association end connected to C_i in the aggregation relation. Then, we have that

$$(\sum_{i=1}^{n} lower_{Ci}) \geq 2 .$$

Mutatis Mutandis, this constraint in fact holds for all types of *partOf* relations.

The *componentOf* is a non-transitive relation. Transitivity holds only in certain contexts, which can be isolated by following certain model patterns defined in sections 5.6 and 7.4.

In UML, only one of the secondary properties or part-whole relations is considered, namely *exclusiveness* (or *shareability*). A standard aggregation relation, represented as a hollow diamond in the association end representing the whole, stands for a non-exclusive (shareable) part-whole relation. The composition relation, represented as a black diamond in the association end representing the whole, stands for an exclusive (non-shareable) one instead. Despite making this distinction, the UML specification is not precise on what exactly the distinction is supposed to represent. Firstly, the specification states that "*composite aggregation is a strong form of aggregation, which requires that a part instance be included in at most one composite at a time"*. In the way it is stated, this sentence is simply wrong: since the composition relation is deemed transitive by the specification, an object x which is part of composite y is also part of any composite z of which y is part. Therefore, it cannot be a necessity for *"a part instance [to] be included in at most one composite at a time"*. Even if we ignore this problem as a formulation mistake, it is in principle still possible to interpret the non-shareability requirement as a demand for global exclusiveness. Global exclusiveness is defined as follows (definition 5.7): $(x <_x y) =_{df} (x < y) \wedge (\forall z\ (x < z) \rightarrow (z \leq y) \vee (y \leq z))$. This means that if x is an (globally) exclusive part of y, then if x is part of z (different from y) then either z is part of y, or y is part of z. As discussed in section 5.4.1, we adopt a weaker notion of exclusiveness, namely, one of exclusiveness w.r.t. to a given universal (or *local exclusiveness*). Therefore, if x of type X is an exclusive part of y of type Y, then there is no other z of type Y such that x is part of z. In summary, the composition relation in UML is taken here to represent an (locally) exclusive *componentOf* relation.

Another problem with the UML concept of composition is that it merges two different ontological meta-properties, namely, *exclusiveness* and *dependence*. This in itself is a case of ***construct overload***. However, the specification is even more ambiguous w.r.t. which kind of dependence relation is being considered. First, it states that "*the composite object is responsible for the creation and destruction of the parts… If a composite object is destroyed, it must destroy all of its parts."* This seems to imply that there is a specific dependence from the parts to the composite, i.e., the parts are *existentially dependent* on the composite. If this were the case, the composite relation in UML would also represent a case of *inseparable parthood* (definition 5.15). Nonetheless, in another part of the text, it is specified that *"[the composite] may remove a part and give it to another composite object, which then assumes responsibility for it. If the multiplicity from a part to composite is zero-to-one, the composite may remove the part, and the part may assume responsibility for itself, otherwise it may not live apart from a composite."* This implies that if the minimum multiplicity from a part to composite is *one* then the part is only *generically dependent* on the universal that the composite instantiates, i.e., the part object can be part of any composite of the same type (part with mandatory whole). Conversely, if the minimum multiplicity from a part to composite is *zero-to-one* then the part is not at all dependent on the composite (part with optional whole).

Despite the possible dependency relations that might hold from the part to the whole, there are also (specific or generic) dependency relations that might hold from the whole to the parts. That is, a whole object can be existentially dependent on a specific part (essential), or generically dependent on the universal that a part instantiates (mandatory part).

Shareability and inseparability/essentiality are completely orthogonal meta-properties, i.e., there are shared parts which are essential (and/or inseparable), and there are exclusive (non-shareable) parts which are only mandatory. An example of the former is depicted in figure 5.33. In an instance of that model, a particular Lecture can be shared by a Regular Course and by a Studium Generale Course. Nonetheless, the lecture is an inseparable part of both courses. An example of the latter is depicted in figure 8.13.b. A heart is not an essential part of a particular human being. However, it is a non-shareable part.

Finally, as discussed in depth in section 7.2, only rigid universals can participate as wholes in essential parthood relations. In the case of anti-rigid universals, the modal necessity of the parts can only be a case of *de dicto* necessity. These parts are named *immutable parts* instead.

In order to eliminate the construct overload aforementioned we propose that shared/composition notation in UML to be used to represent only shareability. To represent mandatory parts and mandatory wholes we use the minimum cardinalities of the represented parthood relation in the corresponding way: a minimum multiplicity of one in the side of the part represents a generic dependence from the whole to the part (mandatory part); a minimum multiplicity of one in the side of the whole represents a generic dependence from the part to the whole (mandatory whole). Additionally, we extend the original UML aggregation notation with the following *tagged values:* (a) {*inseparable* = *true*}: represents inseparable parts (i.e., every part is existentially dependent of a specific whole); (b) {*essential* = *true*}: represents essential parts (i.e., every whole is existentially dependent of a specific part); (c) {*immutable* = *true*}: represents immutable parts (i.e., every whole bears a parthood relation to a specific part in all circumstances that it instantiates that specific whole universal). Since inseparability of parts is a stronger constraint than generic dependence, then whenever {*inseparable* = *true*} the minimum cardinality constraint in the association end connected to the whole must be one. Likewise, whenever {*essential* = *true*} or {*immutable* = *true*} the minimum cardinality constraint in the association end connected to the part must be one.

Finally, in definition 5.14, we have introduced the notion of an extensional individual, i.e., an individual for which all parts are essential. We represent a universal whose instances are extensional entities by a tagged value {*extensional*} on top of the classifier representing that universal. A natural constraint is that all parthood relations in which an {*extensional*} classifier participates as a whole, must have the meta-property {*essential* = *true*}.

Although defined above for the *componentOf* relation, these meta-properties can also be used to qualify the other types of *partOf* relations. In the sequel, we analyze each of these relations separately.

subQuantityOf: As discussed in section 5.5.1, a *quantity* stands for a maximally-connected-amount-of-matter. Since a quantity is maximal, it cannot have as a part a quantity of the same kind. For the same reason, a *subQuantityOf* relation is always non-sharable. For example, take a case in which this relation holds between a quantity of alcohol *x* and a quantity of wine *y*. Since y is self-connected it occupies a self-connected portion of space. The same holds for x. In addition, the topoid[71] occupied by x must be a (improper) part of the topoid occupied by y. Now suppose that there is a portion of wine z (different from y) such that x is a subQuantityOf z. A consequence of this is that z and x overlap, and since they are both self-connected, we can define a portion of wine w which is itself self-connected. In this case, both z and x are part of w and therefore, they are not *maximally-self-connected-portions*. This contradicts the premises that x and z were quantities. Hence, we can conclude that the *subQuantityOf* (Q) relation is always non-sharable.

Since every part of a quantity is itself a quantity, Q-parthood must have a cardinality constraint of *one and exactly one* in the subquantity side. Take once more the alcohol-wine example above. Since alcohol is a quantity (and, hence, maximal), there is exactly one quantity of alcohol which is part of a specific quantity of wine.

As discussed in section 5.6, quantities are mereologically invariant, i.e., the change of any of its parts changes the identity of the whole. In other words, all parts of a quantity are essential. The consequences of this characteristic are: (i) *subQuantityOf* relations are essential parthood relations; (ii) since essential parthood relations are always transitive, *subQuantityOf* is always transitive (i.e., for all a,b,c, if Q(a,b) and Q(b,c) then Q(a,c)); (iii) since quantities are extensional entities, the *weak supplementation axiom* (3) defined to hold for all partOf relations can be replaced by the adoption of the *strong supplementation axiom* (chapter 5) for the case of the *subQuantityOf* relation.

The axiomatization of the *subQuantityOf* relation thus includes the basic axioms of any mereological theory, namely, irreflexivity, anti-symmetry and transitivity of the proper-part relation. But also the *strong supplementation axiom* and the *extensionality principle* (see section 5.1.2). Moreover, it includes

[71]A Topoid is a region of space with a certain mereotopological structure (Guizzardi & Herre & Wagner, 2002a).

the exclusive parthood (definition 5.9) and the essential parthood (definition 5.11) axioms.

In other words, the axiomatization of this relation is the one of *Extensional Mereology (EM)*, with non-shareable parts. That means that two quantities are the same iff they have the same parts, and no two instance of the same quantity kind overlap.

subCollectionOf: Like quantities, collectives are maximal entities. However, in contrast with quantities, the unifying relation of a collective is not necessarily one of physical connection. For this reason, a collective can be shared by two or more collectives. For instance, the crowd C_1 occupying street St_1 and the crowd C_2 occupying street St_2 can both have as a part (*subCollectionOf*) the crowd C_3 occupying the intersection between streets St_1 and St_2. Therefore, *subCollectionOf* (C) can be shareable in some case while non-shareable in others.

In this example, the crowds C_3 and C_1 (or C_2) are not unified by the same type of relation. In fact, the unifying relation of the former is a refinement of that of the latter. Since a collective is a maximal entity, it is not the case that a collective can have as part another collective of the same type (i.e., unified by the same relation). Moreover, for the same reason, any collective can have at maximum one subcollection of a given type. Finally, as discussed in section 5.5.2, since every subcollection of a collective is obtained by refining the unifying relation of the latter, the subCollectionOf relation is always transitive.

As discussed in section 5.6, unlike quantities, collectives do not necessarily have an extensional criterion of identity. That is, whereas for some collectives the addition or subtraction of a subcollection (or a member) renders a different individual, it is not the case that this holds for all of them.

In summary, if cardinality constraints are fully specified, then the C-parthood is as such that: (i) the cardinality constraints in the association end relative to the part is one and exactly one; (ii) only holds between collectives; (iii) it is transitive, i.e., for all a,b,c, if C(a,b) and C(b,c) then C(a,c). Every *subCollectionOf* relation is irreflexive, anti-symmetric, transitive and obeys the weak supplementation axiom. That is to say that the axiomatization of the *subCollectionOf* relation is the one of the *Minimum Mereology (MM)*.

memberOf: The *member-collection* relation is one that holds between a *singular entity* (either a complex or a collective considered as a unity) and a collective. As discussed in section 5.6, *memberOf* (M) relations are never transitive, i.e., they are *intransitive*. This is to say that for all a,b,c, if M(a,b) and M(b,c) then ¬M(a,c). Nonetheless, transitivity does hold across M and

C, i.e., if we have M(x,y) and C(y,z) then it is also the case that M(x,z). In other words, if an individual x is a *memberOf* a collective Y, which is itself a *subCollectionOf* collective Z, then x is also a *memberOf* Z.

Typically members can be shared by more than one collective. However, it is also conceivable to have non-shareable memberOf relations.

As previously mentioned, collectives are not necessarily extensional individuals. However, when this is the case, all *memberOf* relations that the extensional collective participates as a whole are relations of essential parthood. Since a collective (by definition) has a uniform structure, then all members of a collective are supposed to be indistinguishable. As a consequence, it cannot be the case that some members of a collection are essential and others not. In summary, *memberOf* relations are only relations of essential parthood if the collective in the association end connected to the whole entity is an extensional individual.

In figure 8.15 below we present a revision of the UML meta-model in order to faithfully represent the ontological distinctions discussed in this section. As previously discussed, the original UML metamodel only allows for the distinction among parthood relations w.r.t. the shareability meta-property (via the meta-attribute *aggregationKind*). In other words, parthood relations are treated as ordinary associations which can have the meta-property of being an aggregation or a composition. This is a poor modeling choice for several reasons. Firstly, because it does not make justice to the special meta-properties and semantics of parthood relations. Secondly, because it is imprecise on the meaning of non-shareability represented, which collapses both the representations for shareability and inseparability in one single construct. Finally, because it does not sufficiently constrain the metamodel, the current representation of part-whole relations in UML allows for the representation of state of affairs which are not truthful to the underlying conceptualization. For example, one can define a parthood association in which both association ends have incompatible values for the aggregationKind meta-attribute. Moreover, like dependency relations, parthood relations are directed binary relations and therefore cannot be allowed to have more than two association ends. However, in the original UML metamodel, one can define a parthood association with arity higher than two.

In our proposed revised metamodel, we separate the representation of standard domain association from meronymic associations, allowing for a faithful representation and special semantic treatment for the latter. Moreover, we explicitly represent the different kinds of meronymic relations (distinguished by the category of entities they relate) and provide mechanisms to fully qualify them in terms of the ontological meta-properties considered. Finally, we proscribe the use of standard UML associations to represent part-whole relations. In other words, we require that in any implementation of this metamodel the meta-attribute *aggregation* of the UML metaclass Property must have the value *none*.

The metamodeling choices adopted (e.g., by making a *meronymic relationship* a special type of *directed binary relationship*) entails a cleaner metamodel and one that approximates the class of possible models to that of intended ones (see chapter 3).

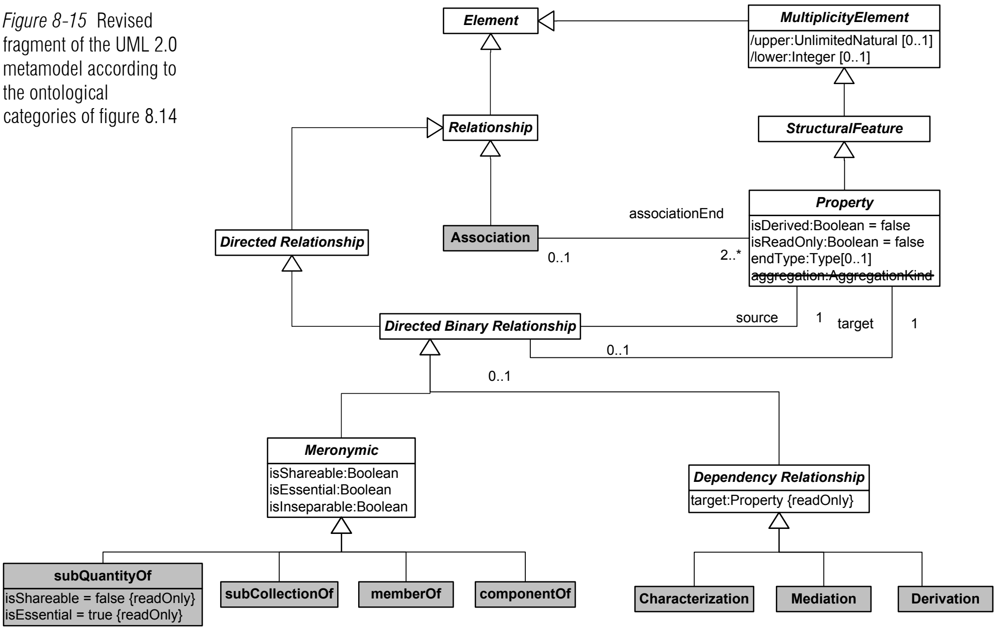


Figure 8-15 Revised fragment of the UML 2.0 metamodel according to the ontological categories of figure 8.14

Once more, we have defined a modelling profile implementing the corresponding revised fragment of the UML metamodel. In table 8.4, we summarize the results of this section by presenting this profile that implements the leaf ontological distinctions among *meronymic* relations depicted figure 8.14, together with the syntatical constraints that characterize these relations. This profile extends the ones presented in tables 8.1 to 8.3.

Table 8-4 Extensions to the UML profile of tables 8.1 to 8.3 which implement the revised metamodel of figure 8.15

Metaclass	Description
Meronymic	Abstract metaclass representing the general properties of all meronymic relations. Meronymic has no concrete syntax. Thus, symbolic representations are defined by each of its concrete subclasses.

Non-reflexivity, Anti-Symmetry, Non-Transitivity and Weak Supplementation.

Constraints

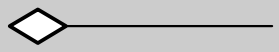
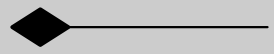
1. Weak Supplementation: Let U be a universal whose instances are wholes and let $\{C_1 \ldots C_2\}$ be a set of universals related to U via aggregation relations. Let $lower_{Ci}$ be the value of the minimum cardinality constraint of the association end connected to C_i in the aggregation relation. Then, we have that

$$(\sum_{i=1}^{n} lower_{Ci}) \geq 2;$$

2. Essential Parthood: The *isEssential* attribute represents whether the meronymic relation is one of essential parthood, i.e., whether the part is essential to the whole. In case the classifier connected to the association end representing the whole is an anti-rigid classifier, then the meta-attribute *isEssential* must be false, whereas the meta-attribute *isImmutable* may be true. However, if *isEssential* is true (in case of a rigid classifier with essential parts) then *isImmutable* must also be true. The concrete representation of this meta-property is via the *tagged value* {essential} decorating the association;

3. Inseparable Parthood: The *isInseparable* attribute represents whether the meronymic relation is one of inseparable parthood, i.e., whether the whole is essential to the part. The concrete representation of this meta-property is via the tagged value {inseparable} decorating the association;

4. Shareable Parthood: The *isShareable* attribute represents whether the meronymic relation is (locally) shareable, i.e., whether the part can be related to more than a whole of that kind. The concrete representation of this meta-property is via the color property of the symbol used to depict this relation (a diamond with or without a decorating letter): if (isShareable = true) then the symbol is shown in white color, otherwise, it is shown in black.

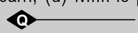
Metaclass	Description and Concrete Syntax
componentOf	componentOf is a parthood relation between two complexes. Examples include: (a) my hand is part of my arm; (b) a car engine is part of a car; (c) an Arithmetic and Logic Unit (ALU) is part of a Central Process Unit (CPU); (d) a heart is part of a circulatory system. Since this is by far the most used in conceptual modeling, we propose the use of the standard UML symbolic representation for aggregation/composition to represent this relation, i.e., we use the symbols  and  to represent the shareable and non-shareable componentOf relations, respectively.

Meta-Properties

Non-reflexivity, Anti-Symmetry, Non-Transitivity and Weak Supplementation.

Constraints

1. The classes connected to both association ends of this relation must represent universals whose instances are *functional complexes*. A universal X is a universal whose instances are functional complexes if it satisfies the following conditions: (i) If X is a sortal universal, then it must be either stereotyped as «kind» or be a *subtype* of a class stereotyped as «kind»; (ii) Otherwise, if X is a mixin universal, then for all classes Y such that Y is a *subtype* of X, we have that Y cannot be either stereotyped as «quantity» or «collective», and Y cannot be a *subtype* of class stereotyped as either «quantity» or «collective».

Metaclass	Description and Concrete Syntax
subQuantityOf	subQuantityOf is a parthood relation between two quantities. Examples include: (a) alcohol is part of Wine; (b) Plasma is part of Blood; (c) Sugar is part of Ice Cream; (d) Milk is part of Cappucino. We propose the icon  to represent this relation.

Meta-Properties

Non-reflexivity, Anti-Symmetry, Transitivity and Strong Supplementation (Extensional Mereology).

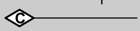
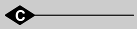
Constraints

1. This relation is always non-shareable (isShareable = false);

2. All entities stereotyped as «quantity» are extensional individuals and, thus, all parthood relations involving quantities are essential parthood relations;

3. The maximum cardinality constraint in the association end connected to the part must be one (self.target.upper = 1).

4. The classes connected to both association ends of this relation must represent universals whose instances are *quantities*. A universal X is a universal whose instances are quantities if it satisfies the following conditions: (i) If X is a sortal universal, then it must be either stereotyped as «quantity» or be a *subtype* of a class stereotyped as «quantity»; (ii) Otherwise, if X is a mixin universal, then for all classes Y such that Y is a *subtype* of X, we have that Y cannot be either stereotyped as «kind» or «collective», and Y cannot be a *subtype* of class stereotyped as either «kind» or «collective».

Metaclass	Description and Concrete Syntax
subCollectionOf	subCollectionOf is a parthood relation between two collectives. Examples include: (a) the north part of the Black Forest is part of the Black Forest; (b) The collection of Jokers in a deck of cards is part of that deck of cards; (c) the collection of forks in cutlery set is part of that cutlery set; (d) the collection of male individuals in a crowd is part of that crowd. We use the symbols  and  to represent the shareable and non-shareable subCollectionOf relations, respectively.

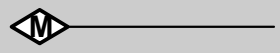
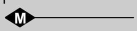
Meta-Properties

Non-reflexivity, Anti-Symmetry, Transitivity and Weak Supplementation (Minimum Mereology).

Constraints

1. The classes connected to both association ends of this relation must represent universals whose instances are *collectives*. A universal X is a universal whose instances are collectives if it satisfies the following conditions: (i) If X is a sortal universal, then it must be either stereotyped as «collective» or be a *subtype* of a class stereotyped as «collective»; (ii) Otherwise, if X is a mixin universal, then for all classes Y such that Y is a *subtype* of X, we have that Y cannot be either stereotyped as «kind» or «quantity», and Y cannot be a *subtype* of class stereotyped as either «kind» or «quantity».

2. The maximum cardinality constraint in the association end connected to the part must be one (self.target.upper = 1).

Metaclass	Description and Concrete Syntax
memberOf	memberOf is a parthood relation between a complex or a collective (as a part) and a collective (as a whole). Examples include: (a) a tree is part of forest; (b) a card is part of a deck of cards; (c) a fork is part of cutlery set; (d) a club member is part of a club. We use the symbols  and  to represent the shareable and non-shareable memberOf relations, respectively.

Meta-Properties

Non-reflexivity, Anti-Symmetry, Intransitivity and Weak Supplementation.
Although transitivity does not hold across two memberOf relations, an memberOf relation followed by subCollectionOf is transitive. That is, for all a,b,c, if memberOf(a,b) and memberOf(b,c) then ¬memberOf(a,c), but if memberOf(a,b) and subCollectionOf(b,c) then memberOf(a,c).

Constraints

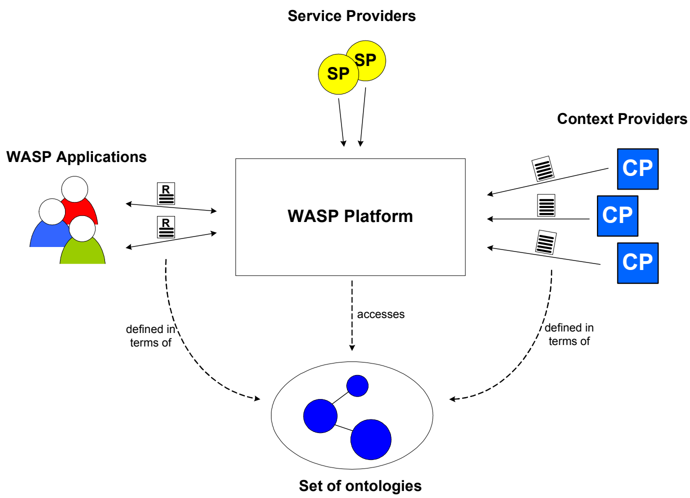
1. This relation can only represent essential parthood (*isEssential* = *true*) if the object representing the whole on this relation is an extensional (*isExtensional* = *true*) individual. In this case, all parthood relations in which this individual participates as a whole are essential parthood relations;

2. The classifier connected to association end relative to the whole individual must be a universal whose instances are *collectives*. The classifier connected to the association end relative to the part can be either a universal whose instances are collectives, or a universal whose instances are functional complexes.

8.4 An Application of the re-designed UML

In (Ríos, 2003), an architecture for an ontology-based context-aware service platform is proposed[72]. This platform, depicted in figure 8.16 below, employs distributed ontologies to define the semantics of syntactic items which are used to compose the messages exchanged by the platform and its environment. These messages include both context-aware applications service subscriptions and context-information supplied by external (context) providers.

Figure 8-16 An ontology-based version of the WASP platform



As demonstrated by Ríos, the use of ontologies in this version of the WASP platform brings a number of important benefits to the original proposal (Costa, 2003). These benefits include:

1. *More intelligent behaviour* and the ability to reason about context information;
2. *Reusability*: the platform can (re)use already existing ontologies for the modeling of context information;
3. *Flexibility*: in contrast to the original proposal the platform is not closed w.r.t. a pre-defined set of context modeling concepts.

Due to these benefits, *ontologies* are being considered in practically all the architectural evolutions of this platform (Costa, 2004b; Santos, 2004).

[72] This platform is discussed in more detail in chapter 3 of this thesis. For a complete description of the proposed architecture one should refer to (Ríos, 2003).

In spite of these benefits, in (Ríos, 2003), the author discusses the insufficiency of *semantic web languages* (and the *lightweight ontologies* produced) to prevent interoperability problems when different ontologies are integrated. Ríos, proposes the following illustrative example on the integration of five independent domain ontologies.

The first ontology (which has a fragment depicted in figure 8.17) is a *Spatial ontology* that defines the concepts of *Spatial Location* and *Physical Object* and their corresponding properties (e.g., Spatial Location includes attributes such as *latitude* and *longitude* coordinates). This ontology might be considered as a very simple upper level ontology, as it does not define knowledge related to any specific domain. Thus, it can be referred or imported by different domain ontologies. For example, this ontology could be used by a GPS sensor agent to provide a service to track the location of physical objects in a context-aware platform.

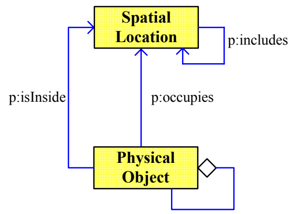


Figure 8-17 Fragment of a Spatial Ontology (from Ríos, 2003)

There are two axioms defined for the *Spatial* ontology:

1. For every two arbitrary physical objects X and Y, if there are two spatial locations A, B, such that X occupies A, Y occupies B, and A is equal to B, then X and Y are the same physical object.

This axiom helps the users of this ontology to identify an object in a given time instant (*synchronic identity*). However, it cannot distinguish if two physical objects X and Y at different spatial locations in different time instants are the same objects (*diachronic identity*). For this reason, the ontology prescribes the following axiom.

2. For every two arbitrary physical objects X and Y, X is equal to Y if and only if they have the same parts, i.e., the *identity criterion* for physical objects is determined by the sum of its parts (*extensional identity criterion*).

A second ontology presented is a fragment of a *Medical* ontology (figure 8.18) defining some medically related concepts such as *Human Organ* or *Human Being* and *Surgery Room*. Ríos presents a situation, in which the

Medical ontology imports the concepts of *Spatial Location* and *Physical Object* from the *Spatial* ontology (symbolized by the i: character in the name of the class representing these concepts). The idea is to allow for the possibility of defining applications for checking location of patients, locate organs for transplants, and so forth.

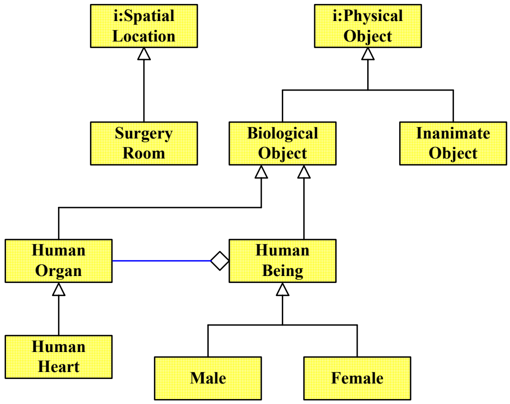


Figure 8-18 Fragment of a Medical Ontology (from Ríos, ibid.)

A third ontology presented is shown in figure 8.19. The idea is to represent a fragment of *Legal* ontology, which represents legal aspects of people and that can be used by bureaucratic applications. This ontology imports the concepts of *Human Being*, *Male* and *Female* from the *Medical* ontology. This import allows, for example, legal applications to refer to the medical histories of people; to have access to their personal data (e.g., blood type, skin colour, fingerprints, height, weight); to differentiate people by sex; or to maintain a record of living and deceased people in a community.

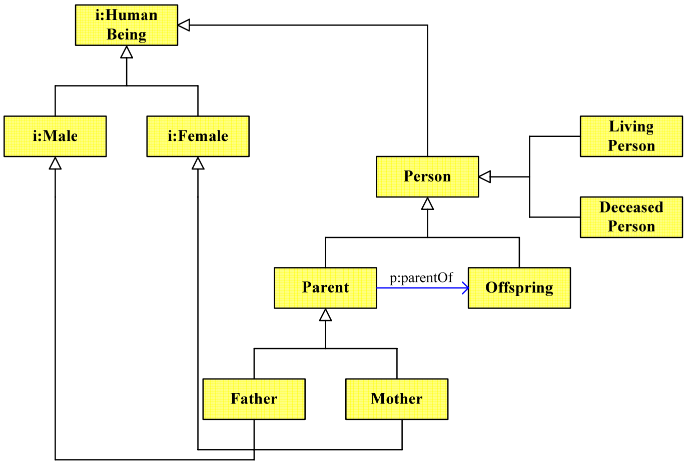


Figure 8-19 Fragment of a Legal Ontology (from Ríos, ibid.)

Figure 8.20 shows a fragment of *Museum* ontology, which imports the *Spatial* and the *Medical* ontology to define spatial locations like galleries within a museum, or inanimate objects like statues. These imported ontologies allow for applications to locate objects within the museum (e.g., statues, paintings) using the *Museum* ontology.

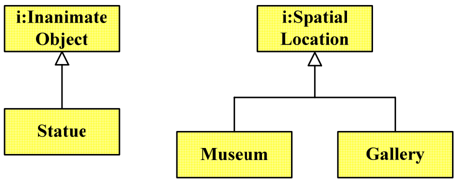


Figure 8-20 Fragment of a Museum Ontology (from Ríos, ibid.)

Finally figure 8.21 represents a fragment of a Musical Ontology containing some related concepts. Ríos defines as an application for the complete ontology (to which this fragment belongs) an *Event Advisor*, which notifies users about upcoming events that match their personal interests. The Music ontology imports from the Legal ontology concepts like person (and its possible attributes, like name, age, sex, etc.).

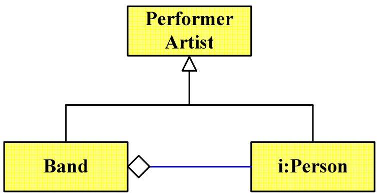


Figure 8-21 Fragment of a Music Ontology (from Ríos, ibid.)

Ríos describes the following problems that can occur with the integration of these ontologies:

1. "An application using the *Medical* ontology can derive the following wrong information: if a human being receives a heart transplant, he/she becomes a different human being. This is due to the extensional identity criteria, which is defined for physical objects in the *Spatial* ontology. If the identity of an object is defined by the sum of its parts, then changing one of the parts changes the identity of the object. Similarly, consider a tourist route planner application that plans a route including tourist points of interest or events never seen by the user of the application. Due to an accident, a human statue known by the user has lost a hand. The application will consider this statue different from the one the user visited; therefore it will be included in the route plan by error. This example uses a physical object (statue) for the purpose of illustration of the problem, but an analogous situation can be imagined with events such as a play or a concert";

2. "Suppose an application for the obituary section of a music newspaper, which sends information about artists who die. It uses the Musical ontology, which imports the Legal ontology (to reuse the concept of person). The application will malfunction and it will send information about every person who dies, since [according to the ontology of figure 8.21] every person is a performer artist. The intention in the ontology represented in figure [8.21] is to represent that either persons or bands are performer artists. However, as a side effect, the ontology also states that every person is a performer artist";

3. "Since Musical ontology imports the Legal ontology, which imports the Medical ontology, the heart (and all other parts) of a person can be inferred to be part of a band, due to transitivity of the "*partOf*" relation, which can cause undesirable inferences to be derived".

A fragment of a *Music Ontology* such as the one presented in figure 8.21 can be found in practice in the *MusicBrainz II Metadata proposal*[73]. A simplified version of the MusicBrainz database structure is presented in figure 8.22.

[73] MusicBrainz is a large database of music metadata (http://www.musicbrainz.org/).

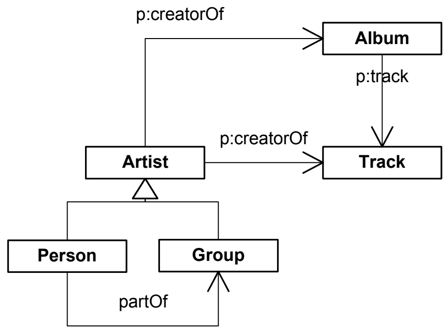


Figure 8-22 Fragment of the *MusicBrainz* metadata proposal

In fact, this model excerpt can be seen as an extension of a more general pattern found in the *FOAF (Friend-of-a-Friend)* ontology[74] (Brickley & Miller, 2003) shown in figure 8.23 below. The FOAF ontology is a proposal for capturing concepts related to the representation of personal information and social relationships. Its purpose is to serve as a basis for developing computational support for online communities. The FOAF ontology is also used by the SOUPA ontology (see discussion below) to support the expression and reasoning about a person's profile and social connections in pervasive computing applications.

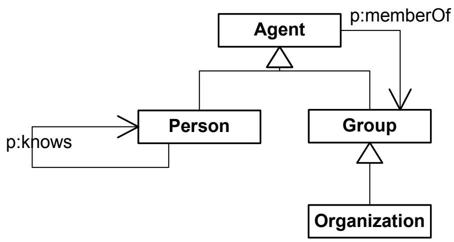


Figure 8-23 Fragment of the *FOAF (Friend-of-a-Friend)* ontology

The conceptualization modelled in the fragment of figure 8.23 has an analogous representation in the *SOUPA (Standard Ontology for the Ubiquitous and Pervasive Application)*[75] Ontology (Chen, 2004; Chen et al., 2004). The SOUPA ontology also includes a Spatial Ontology (such as the one of figure 8.17), whose fragment is presented in figure 8.24.

---

[74] See The FOAF Project (http://www.foaf-project.org/) and the FOAF Vocabulary Specification (http://xmlns.com/foaf/0.1/)
[75] http://pervasive.semanticweb.org.

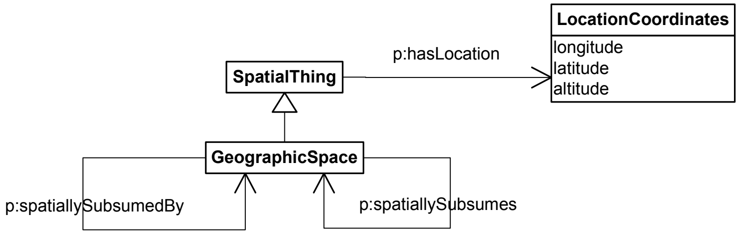


Figure 8-24 Fragment of the *SOUPA (Standard Ontology for the Ubiquitous and Pervasive Applications)* dealing with spatial concepts and relations

SOUPA is a proposal for a standard ontology for supporting pervasive and ubiquitous computing application. It integrates parts of several other ontologies such as FOAF, DAML-Time (Hobbs, 2002; Pan & Hobbs, 2004), OpenCyC[76] and OpenGIS (Cox et al., 2003) (Spatial Entities), Rei Policy ontology (Kagal & Finin & Joshi, 2003), and COBRA-ONT (Chen & Finin & Joshi, 2004), but also an ontology defining agent related concepts named *MoGATU BDI* ontology (see figure 8.25)[77].

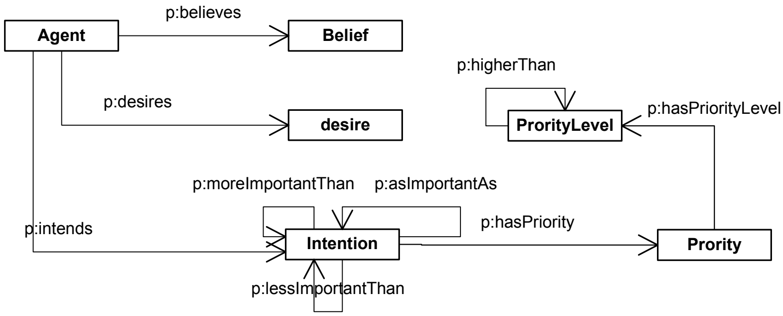


Figure 8-25 Fragment of the *MoGATU BDI* ontology

In the sequel, we apply the ontologically well-founded version of UML redesigned in this chapter, together with the modelling techniques proposed throughout this thesis, to illustrate the use of the language in making explicit the underlying ontological commitments and in producing an adequate conceptual model representation that integrates the ontologies used in the examples of (Ríos, 2003) as well as the fragments of MusicBrainz II, FOAF, SOUPA and MoGATU BDI presented above.

The result of redesigning the integrated model is shown in figure 8.26 below.

[76] http://www.opencyc.org/.
[77] http://mogatu.umbc.edu/bdi/.

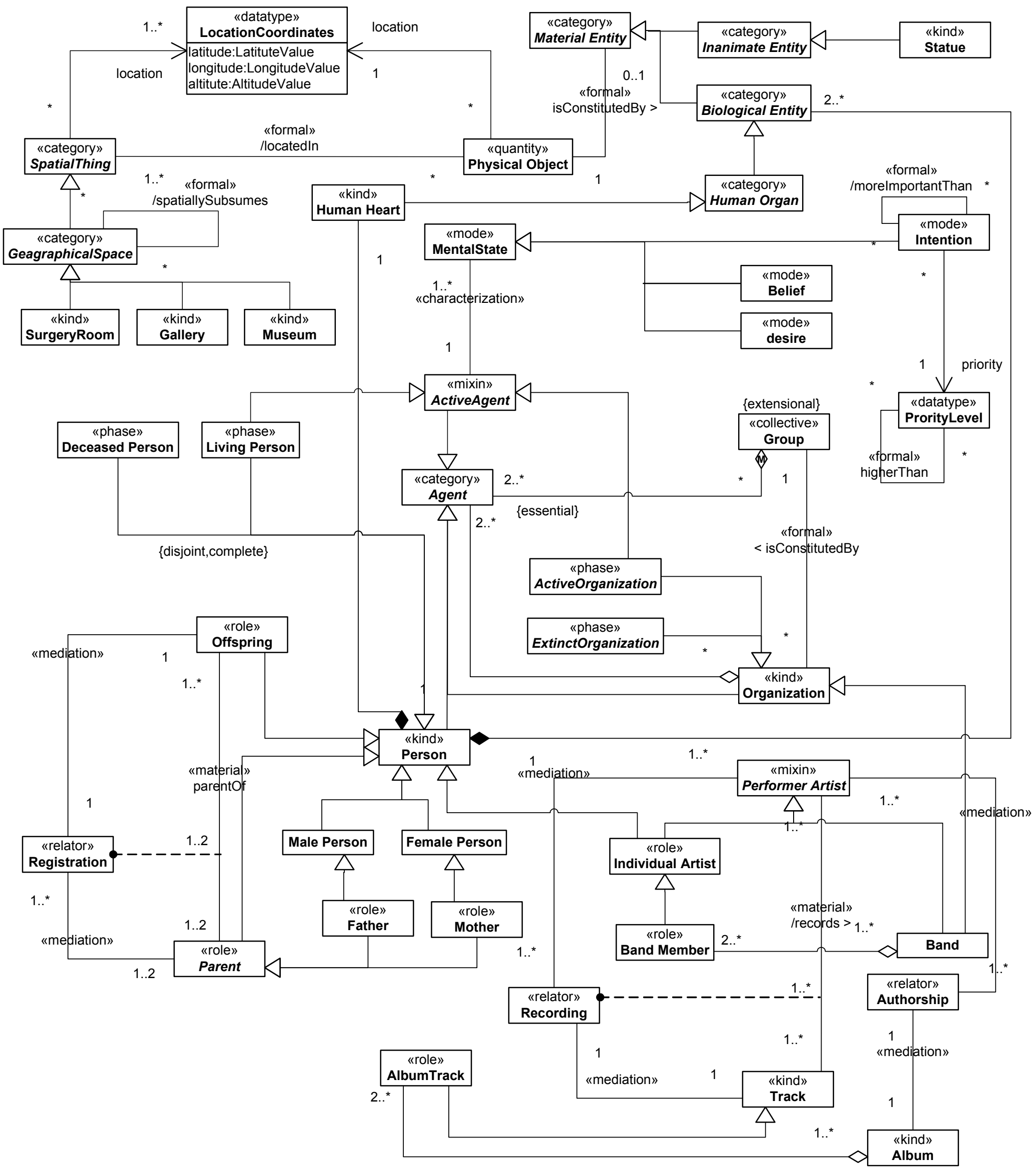


Figure 8-26 An integrated ontology using the ontologically well-founded version of UML proposed in this thesis

In producing this conceptual specification we have been forced to make a number of assumptions. This is due to the lack of information provided by the integrated ontologies w.r.t. the real-world semantics of the concepts represented. We emphasize, nonetheless, that the goal here is to demonstrate the suitability of the modelling language proposed. Thus, the underlying conceptualization which results from the set of assumptions made is of lesser relevance.

First of all, in the model of figure 8.26, we separate physical spaces such as Surgery Room, Gallery and Museum from their spatial location (as informed by a GPS system), as it is the case in the SOUPA ontology. Firstly, because some geographical spaces (e.g., university) can have several location coordinates, but also because different geographical spaces can be associated with a particular set of location coordinates in different circumstances. Thus, whereas the physical spaces are represented by the general category of spatial thing, the spatial location of these physical spaces is modelled here as a quality domain composed of the quality dimensions *latitude*, *longitude* and *altitude*. The relation *includes* between spatial locations in figure 8.17 is represented by the relation *spatiallySubsumes* between *geographical spaces* in figure 8.26.

A *physical object* is said to be *located in* (*isInside*) a geographical space if the location of the former is included in that of the latter. Additionally, we assume that, in contrast with physical objects, *Biological Entities*, *Persons* and *Inanimate Entities* do not carry extensional principles of identity. Therefore, we differentiate a *Biological Entity*, such as a heart, from the quantity of cellular tissue that constitutes this entity. Likewise, we differentiate an *inanimate object*, such as a statute, from the raw material that constitutes it (e.g., a lump of clay).

A person is composed of a number of biological entities that amount to the person's body and its constituent parts. A person has the spatial location of its body, which in turn is derived from the spatial location of its constituent physical object. The analogous holds for inanimate entities. By separating the universals that carry different principles of identity, we avoid the problems mentioned by Ríos in (1) above.

The problem mentioned in (2) is solved in figure 8.26 by applying the role modelling design pattern proposed in chapter 4. *Performer Artist* is a mixin universal, since it has as instances individuals that obey incompatible principles of identities, namely, bands (which are kinds of organizations) and individual artists (which are persons). However, in this case the mixin universal *Performer Artist* is a *semi-rigid mixin* (as opposed to a role mixin), since it is a rigid universal for some of its instances (bands) and anti-rigid for others (individual artists).

Figure 8.27 below presents an excerpt of the specification of figure 8.26 that focuses on the alleged derived meronymic relation between a human heart of a band member and its band. As this specification shows, the relation between a Human Heart and a Band Member is one of *indirect functional parthood* (i_1) and the relation between a Band Member and a Band is one of *direct functional parthood* (d_1). As discussed in chapter 7, transitivity does not hold across i_1 and d_1. Thus, in this specific specification, a Human Heart is not part of a Band.

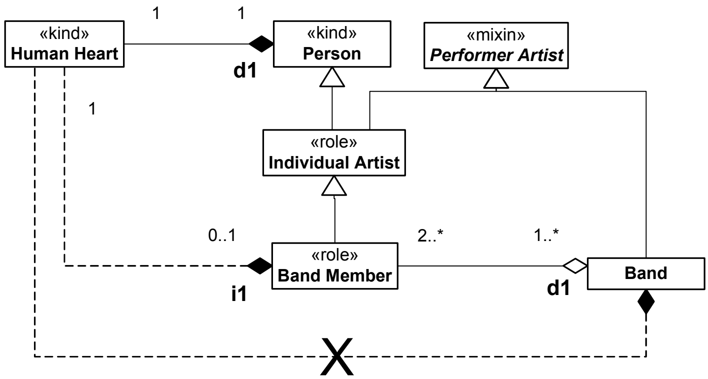


Figure 8-27 Instantiation of the pattern that exemplifies situations in which transitivity does not hold across functional parthood relations

As an agent, a Person can be a member of a group (figure 8.26). For example, Eric Clapton is a member of the British Guitar Players. However, Clapton's hands are not members of this group. That is, also in this case, transitivity does not hold across the two meronymic relations represented, since the combination of *componentOf* and *memberOf* relations is never transitive (figure 8.28).

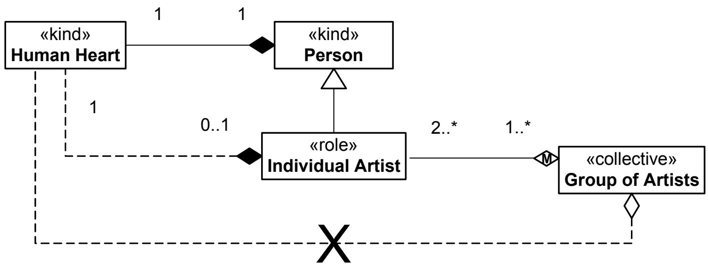


Figure 8-28 Although a human heart can be part of an individual artist, which in turn is part of a group of artists, a human heart is never a part of a group of artists. This is because the combination of *componentOf* and *memberOf* parthood relations is never transitive

There are a number of material relations in the model of figure 8.26. As it can be noticed, the explicit representation of the foundations of these relations contributes to make their meaning evident. Take for instance the relation *parentOf* between *Parent* and *Offspring*. In this case, we assume that parent is considered in this conceptualization in the legal not in the biological sense, and that in legal terms a person is a parent of another (offspring) iff the former is registered and legally recognized as such. Therefore, we explicit represent the *Registration* relators that connect parents to their offsprings.

The explicit representation of the relator universal allows for the unambiguous representation of the cardinality constraints associated to the relating universals. This is also the case for the *records* relation between the universals *Performer Artist* and *Track*. The multiplicity one-to-many from Track to Performer Artist leaves open several possible interpretations for the meaning of this relation. Does this multiplicity mean that a track like *Georgia on my mind* can have several recordings (e.g., one by Ray Charles, and another by Jerry Lee Lewis)? By explicitly representing the *Recording* relator, the model makes clear that what is meant by a track is the result of specific recording. However, several artists can participate in one single recording (e.g., both Clapton and B.B.King participate in the recording of *Riding with the King*). Thus, the track *Georgia on my mind* recorded by Ray Charles, and the one recorded by Jerry Lee Lewis are different tracks.

The model of figure 8.22 duplicates the relation creator between artist and album, and artist and track. The intention is to allow for the representation of tracks that are not parts of albums (i.e., that only exist as digital tracks). This situation is modeled in figure 8.26 by having *AlbumTrack* as a restriction of the type Track, in which the *restriction condition* is to be a part of an Album. However, it is unclear in the original model whether these two relations have the same real-world semantics. One interpretation is that, in case a track is part of an album, then the creator of the track must be same as the creator of that album. Still in this interpretation, in the case that different artists participate in the recording of different tracks of the same album (a song collection) they would all be considered creators of that album. The problem is that this interpretation does not allow for the situation in which an artist participates in the recording of one of more tracks of a given album, but is not considered an author of that album. Take, for example, U2's *Rattle & Hum*. Although B.B.King participates in the recording of *Angel of Harlem*, he is not considered an author of that album. We therefore assume that the relation between an artist and an album is one of *legal rights*. This is model in figure 8.26 by an *Authorship* relator universal.

In FOAF, the concept of an agent includes both living and deceased[78] persons. It has also an excessively informal concept of a *Group*, *"covering informal and ad-hoc groups, long-lived communities, organizational groups within a workplace, etc"*[79]. It also allows, for example, for the formation of groups

[78] The FOAF ontology also allows for both real and imaginary characters. We excluded imaginary characters in this specification for the sake of simplicity.

[79] FOAF Vocabulary Specification (http://xmlns.com/foaf/0.1/).

containing only deceased persons. The concept of group in FOAF, hence, seems to include entities that range from collectives with extensional identity principles to social complexes (organizations). We assume here that organizations should be considered as agents, but that mere extensional collections should not. Thus, Agents can be parts of Groups and, in particular, organizations are constituted by Groups. However, Groups are not considered here to be agents.

At first, it seems that the concepts of an agent in FOAF and in MoGATU BDI are equivalent. However, in MoGATU, an agent bears mental states such as beliefs, desires and intentions. We assume here that only living persons and actual organizations can bear these mental states. We use the universal *Active Agent* to model the MoGATU concept of an agent. Active Agent is *mixin* subsuming both *Living Person* and *Active Organization*. Consequently, under these assumptions, a MoGATU agent is an anti-rigid universal. So, for instance, although Aristotle is a (FOAF) Agent, it cannot in the present world have mental states and, consequently, it is not an instance of *Active Agent*. We have an analogous situation for Organizations. For example, *The Beatles* cannot be considered a MoGATU agent in the present world.

Mental states are existentially dependent entities. For example, a belief depends rigidly on a specific bearer active agent, i.e., a particular belief cannot exist without inhering one (and always the same) active agent. Mental states are therefore special types of intrinsic moments (modes) and are represented here as such.

By objectifying intrinsic moments we can also represent their attributes and the relations they participate. For instance, every *intention* has an inhering *priority* quality, which is modeled in figure 8.26 via an attribute function that maps intentions to priority qualia in a *priority level* quality dimension. This quality dimension is a finite set of values ranging from *level 0* to *level 10* and totally ordered under the *higherThan* and *lowerThan* relations.

The relations *moreImportantThan, lessImportantThan* and *asImportantAs* relations between intentions are formal relations derived from the individual priority levels of their relata. The first two relations are anti-symmetric and transitive. The last one is an equivalence relation. As discussed in chapter 6, these meta-properties are derived from the meta-properties of the relations between qualia in the corresponding underlying quality dimension.

8.5 Final Considerations

In chapter 2 of this thesis we have proposed a framework for language evaluation and design. This framework establishes a systematic way to evaluate the domain appropriateness of a modelling language by comparing a concrete representation of the universe of discourse referred by the language (represented in terms of a reference ontology) and a meta-model of this language. The goal is to reinforce the strongest possible homomorphism between these two entities.

In this chapter, we provide an exemplification of this framework. As a representation of the selected general (meta-level) domain, we have the foundational cognitive ontology produced throughout chapters 4 to 7 of this thesis. As a selected language, we take the Unified Modeling Language (UML) as a candidate language for conceptual modelling of structural aspects of application domains. The choice for UML is based on the fact that this language is currently a *de facto* standard, and because it represents an interesting case study due to the complexity of its metamodel. Furthermore, there is a growing interest in the adoption of UML as a language for conceptual modelling and ontology representation (Cranefield & Purvis, 1999; Baclawski et al., 2001; Kogut et al., 2002). A more explicit statement of interest on applying UML for this purpose is made by the OMG Ontology Metamodel Definition Request for Proposals (OMG, 2003a): *"The familiarity of users with UML, the availability of UML tools, the existence of many domain models in UML, and the similarity of those models to ontologies suggest that UML could be a means towards more rapid development of ontologies. This approach continues the Object Management Group's 'gradual move to more complete semantic models' as noted in the Model Driven Architecture paper. It would also create a link between the UML community and the emerging Semantic Web community, much as other metamodels and profiles have created links with the developer and middleware communities."*

As demonstrated in sections 8.1 to 8.3 of this chapter, the foundational ontology used suitably supports the redesign of the language's metamodel to obtain a conceptual cleaner, semantically unambiguous and ontologically well-founded version of UML.

Although not discussed here, an extension of this foundational ontology that includes *perdurants* and *intentional and social entities* has been employed to provide a foundation for agent modelling concepts (Guizzardi & Wagner, 2005b), and to analyze the ontological semantics of the some enterprise modeling languages and frameworks (Guizzardi & Wagner, 2005a), among them most notably the REA (Resource-Event-Agent) framework (Geert & McCarthy, 2002). Moreover, a preliminary version of the ontologically well-founded UML produced in this chapter has been used with interesting results for analyzing conceptual models of genome sequence applications in (Guizzardi & Wagner, 2002).

Finally, in order to show the usefulness of the redesigned version of UML produced, we employ it to tackle some semantically intetoperability problems highlighted by (Ríos, 2003), which can happen in the integration of lightweight ontologies. These problems happen exactly because of the inadequacy of the modelling language used (OWL) in making explicit the underlying ontological commitments of the conceptualizations involved.

The conceptual modelling language proposed here was proven useful in addressing these problems. First, by precisely representing the (modal) meta-properties of the underlying concepts, it allows for an explicit account of their ontological commitments. Second, by providing solutions to classical and recurrent problems in conceptual modelling (e.g., representation of roles with multiple allowed types, the problem of transitivity of parthood relations, the problem of collapsing single-tuple and multiple-tuple multiplicity constraints in the representation of associations, among others), it allows for the production of conceptually clean and semantically unambiguous integrated models.

This case study exemplifies the approach defended in this thesis for semantic interoperation of conceptual models (discussed in chapter 3). We defend that in a first phase of off-line meaning negotiation, an ontologically well-founded modelling language should be used. The main requirements of this language are *domain and comprehensibility appropriateness*. Once this meaning negotiation and semantic interoperation phase is complete, then a knowledge representation language can be used to express the results produced on this phase. The requirements of this second language instead are high computational efficiency and machine-understandability.

In the original scenario proposed by Ríos, the ontologies used are created for the purpose of the example, with the aim to represent stereotypical cases. However, in section 8.4, we demonstrate that real ontologies exist in current Semantic web efforts which are structurally similar to the ones proposed by Ríos, and which are used by practitioners in concrete semantic web applications. Moreover, we also show that there are concrete efforts to unify these separate lightweight ontologies in context-aware applications (e.g., SOUPA) in a manner analogous to the one described in the scenario proposed by Ríos.

Chapter 9

Conclusions

"Life can only be understood backwards
but it must be lived forwards"
Soren Kierkegaard

In this chapter we discuss some final conclusions about the work presented in this thesis. In each of the previous chapters, we have already presented a *final considerations* section including a detailed discussion of the benefits of the approach proposed in that chapter, as well as its main advantages when compared to related work in the literature. For this reason, in the remaining of this chapter we focus mainly on two topics, namely, the most important research contributions of this thesis (sections 9.1 to 9.12), and plans for future work (section 9.13).

9.1 Language Evaluation and (Re)Design Framework

In chapter 2 of this thesis, we elaborate on the relation between a modeling language and a set of real-world phenomena that this language is supposed to represent. We focus on two aspects of this relation, namely, the *domain appropriateness*, i.e., the suitability of a language to model phenomena in a given domain, and its *comprehensibility appropriateness*, i.e., how easy is for a user of the language to recognize what that language's constructs mean in terms of domain concepts and, how easy is to understand, communicate and reason with the specifications produced in that language. We defend that both these properties can be systematically evaluated for a modeling language w.r.t. a given domain in reality by comparing a concrete representation of the worldview underlying this language (captured in a *metamodel of the language*), with an explicit and formal representation of a conceptualization of that domain, or a *reference ontology*.

We therefore propose a framework for language evaluation and (re)design which aims, in a methodological way, to approximate or to increase the level of homomorphism, between a metamodel of a language and a reference ontology. This framework comprises a number of properties (*lucidity*, *soundness*, *laconicity*, *completeness*) that must be reinforced for an isomorphism to take place between these two entities.

The framework proposed combines two existing proposals in the literature:

1. the one of Corin Gurr (Gurr, 1998, 1999), which focuses on the evaluation of individual representations;
2. the one of Yair Wand and Ron Weber (Wand & Weber, 1989, 1990; Weber, 1997), which aims at the evaluation of representation systems.

In addition, our framework extends these two proposals in several ways. Compared to them, our framework possesses the following advantages:

– Gurr uses regular algebraic structures to model a domain conceptualization. We strongly defend the idea that the more we know about a domain the better we can evaluate and (re)design a language for domain and comprehensibility appropriateness. As we show in chapter 2, there are important meta-properties of domain entities (e.g., *rigidity*, *relational dependency*) that are not captured by ontologically-neutral mathematical languages (such as algebras or standard set-theories), and that the failure to consider these meta-properties hinders the possibility of accounting for other important aspects in the design of visual modeling languages.

– In Wand & Weber's work, no attention at all is paid to pragmatic aspects of modeling languages. Here we show how the ontological meta-properties of the domain concepts captured in a reference ontology can be exploited in the design of efficient visual pragmatics for visual modeling languages.

– Wand & Weber's work focuses solely on the design of general conceptual modeling languages. The framework and the principles proposed here instead can be applied to the design of conceptual modeling languages irrespective to which generalization level they belong, i.e., it can be applied both at the level of material domains and corresponding domain-specific modeling languages, and at the (meta) level of a domain-independent (meta) conceptualization that underpins a general conceptual (ontology) modeling language. In fact, this ontology-based framework amounts to an important contribution to the area of domain-specific languages design methodologies (as acknowledged, for instance, in Girardi & Serra, 2004).

– As in Wand & Weber's proposal, the focus of our framework is on the level of systems of representations, i.e., on the evaluation of modeling languages, as opposed to a focus on individual diagrams produced using a language. Nevertheless, as it is shown in chapter 2, by considering desirable properties of the mapping of individual diagrams onto what they represent, we are able to account for desirable properties of the modeling languages used to produce these diagrams, extending in this aspect Wand & Weber's work.

Wand & Weber's framework addresses solely the relation between ontological categories and the modeling primitives of a language. It pays no attention to the possible constraints governing the relation between these categories. Moreover, it does not consider the necessary mapping from these constraints to equivalent ones, to be established between the language constructs representing these ontological categories. For example, consider two ontological categories A and B, which are represented in a language metamodel by two constructs C and D. If there are constraints on the admissible relations between A and B, these constraints must be also represented in the language metamodel in terms of the possible relations between C and D. In chapter 3, we refine our framework, by establishing a formal relation between the set of valid models that can be produced in a domain modeling language, and the set of state of affairs which are deemed admissible by a conceptualization of that domain. To put it simply: let $\mathcal{O}$ be an ontology of a domain conceptualization $\mathcal{C}$ such that $\mathcal{M}$ is a valid model of $\mathcal{O}$ iff it represents a state of affairs deemed admissible by $\mathcal{C}$. Now, let $\mathcal{S}_{\mathcal{L}}$ be the set of valid models of language $\mathcal{L}$ delimited by $\mathcal{L}$'s metamodel $\mathcal{MT}$; let $\mathcal{S}_{\mathcal{O}}$ be the set of valid models of ontology $\mathcal{O}$. We can formally define how truthful $\mathcal{L}$ is to the domain conceptualization $\mathcal{C}$, by comparing how close the sets $\mathcal{S}_{\mathcal{L}}$ and $\mathcal{S}_{\mathcal{O}}$ are. In an ideal situation, these sets coincide and $\mathcal{MT}$ and $\mathcal{O}$ are isomorphic. In other words, in this case, $\mathcal{L}$ admits as valid models all and only those that represent admissible state of affairs according to $\mathcal{C}$.

9.2 Ontological Foundations for Conceptual Modeling

The language evaluation and design framework proposed here can be applied both at the level of material domains (e.g., genomics, archeology, multimedia, fishery, law, etc.) and corresponding domain-specific modeling languages, and at the (meta) level of a domain-independent (meta) conceptualization that underpins a general conceptual (ontology) modeling language. Nevertheless, due to the objectives of this thesis, we have focused on the level of meta-conceptualizations and of general conceptual modeling languages.

The domain appropriateness of a domain modeling language can be precisely defined in terms of the difference between the set of valid models of this language, and the set of *intended models* according to a given conceptualization of that domain. To put it baldly, an intended model according to a conceptualization C is a model that represents a state of affairs deemed admissible by C. Now, let C be a conceptualization of material domain in reality, such as genealogy. The possible state of affairs according to C would exclude, for instance, one in which a person is its own parent. As discussed in depth in chapter 3, the definition of which possible state of affairs comprises a material conceptualization C is far from arbitrary, but determined by the set of laws that constitute reality. So, we can define a meta-conceptualization C', which contains all material conceptualizations C_i that are truthful to reality, i.e., domain conceptualizations that admit only state of affairs that can actually exist in reality. By applying the same line of reasoning to this meta-level of abstraction we have that a suitable general conceptual (ontology) modeling language that can be used to create domain ontologies (or conceptual models of material domains) is one which valid models are exactly the intended models of the meta-conceptualization C'.

Defining a repertoire of real-world categories (meta-conceptualization C') that can be used to articulate domain conceptualizations of reality is the very business of the discipline of *formal ontology* in philosophy. Thus, this task can greatly benefit from the conceptual tools and theories that have been developed along the years in this area. A meta-ontology representing a meta-conceptualization C' constructed using the theories developed by formal ontology in philosophy is named a foundational ontology.

From an ontological point of view, most conceptual modeling languages (e.g., UML, ER, LINGO, CCT, OWL) commit to a simplistic set-theoretical meta-conceptualization, and to an inadequate theory of universals (either *class* or *predicate nominalism*). These ontological choices can be warranted for the so-called *lightweight ontology representation languages* (e.g., the Semantic web languages), by arguing in favor of mathematical simplicity and convenience, or to justify practical performance trade-offs. However, as discussed in depth in chapter 6, there are many serious philosophical problems with these commitments, which make them unsuitable as a foundation for a general conceptual modeling language.

One of the main contributions of this thesis is to construct a meta-ontology which can be used as a foundation for conceptual modeling concepts and languages. As discussed in chapter 3, due to the objectives of conceptual modeling languages, this foundational ontology should be the result of a *descriptive metaphysics* effort. Thus, the conceptual modeling ontology proposed here considers not only results from metaphysics, but also from areas such as cognitive science and linguistics, with the main objective of capturing the ontological distinctions underlying natural language and human cognition.

The foundational ontology proposed here is one of the few and one of the most comprehensives approaches for developing *cognitive ontological foundations* for conceptual modeling languages found in the literature. The benefits of such a choice have been demonstrated in the related work sections of chapters 4 to 7 of this thesis.

The ontological foundations for conceptual modeling proposed here have been developed in four complementary parts, namely, classes and class hierarchies (chapter 4), part-whole relations (chapter 5), properties (chapter 6) and roles (chapter 7). In the following four subsections, we discuss the contributions of this thesis in each of these topics, respectively.

9.3 Theory of Universals and Universals Taxonomic Structures

In chapter 4, we present a well-founded theory of universals to address the topic of universals in conceptual (ontology) modeling. Universals are fundamental for conceptual modeling, being represented in all major conceptual modeling languages (e.g., OO classes, EER Entity types, OWL concepts, etc.), and in the practice of conceptual modeling, a set of concepts is often used to represent distinctions in different sorts of universals (Type, Role, State, Mixin, among others).

However, there is still a deficiency of methodological support for helping the user of the language to decide how to model elements that denote universal properties in a given domain and, hence, modeling choices are often made in ad hoc manner. Likewise it is the judgment of what are the admissible relations between these concepts. In addition, as also discussed in chapter 4, there is still much debate in the conceptual modeling literature regarding the meaning of these categories.

The theory of universals we propose is founded in a number of results in the literature of philosophy of language and descriptive metaphysics and supported by substantial empirical evidence from research in cognitive psychology. By using a number of formally defined meta-properties (identity supply, identity carry, rigidity, relational dependency) we can generate a typology of universals, in which modeling concepts like types, roles, phases and mixins (among others) can be defined. Moreover, by considering the postulates proposed by this theory, we can formally define constraints on the possible relations to be established between these modeling concepts.

This theory represents an important contribution to the theory of conceptual modeling. First, because it can be used to formally characterize these concepts which are ubiquitous in conceptual modeling, and to provide methodological guidelines for how to apply these modeling concepts in practice, e.g., by helping the user to choose the most suitable concept to model a given general term in the universe of discourse. In addition, the theory can serve as a well-founded basis to harmonize different conceptions of these entities found in the literature.

Second, the theory can be used to evaluate the conceptual quality of type hierarchies (or concept taxonomies), and to solve recurrent modeling problems in the practice of conceptual modeling (as demonstrated in chapter 4). In particular, as discussed in section 9.7 below, by employing the categories and postulates of this theory we have been able to propose a *design pattern* capturing a solution to a recurrent and much discussed problem in *role modeling*.

9.4 Two Systems of Modal Logic with Sortal Quantification

In order to formally characterize the ontological distinctions and postulates proposed by the theory of universals aforementioned, we have presented two different (albeit complementary) extensions to traditional systems of quantified modal logics, namely, the languages L_1 and L_2 discussed in chapter 4. Although far from complete, these languages are successful in formally characterizing in a simpler way the important distinction between sortals and general property universals (non-sortals) w.r.t. to the former's exclusive ability to supply a principle of persistence and transworld identity to its instances. By doing this, we can address the limitations of classical (unrestricted extensional) modal logics, which reduces ontologically very different categories to the same logical footing.

In particular, in the case of L_1 we have employed the notion of *individual concept*, as a possible ontological interpretation to the notion of object identifiers in conceptual modeling. Moreover, as demonstrated by (Heller et al., 2004), a language such as L_1 can play an important role in relating *endurantistic* and *perdurantistic* views of entities (views of an entity as an object and as a process, respectively).

The language L_1 is a system of *intentional modal logic* based on the first of the four systems proposed in (Gupta, 1980). However, it contains some important differences. In one sense, it can be seen as an extension of Gupta's proposal by elaborating on different types of sortal universals. However, differently from Gupta, it assumes an *absolute qualified view of identity* (van Leeuwen, 1991), as opposed to a *contingent view*. Thus, in our approach, although all principle of identities are sortal supplied, if two objects are identical according to one principle, then they are necessarily identitical. For Gupta, conversely, two objects can be identical in a given world and diverse in other. The disadvantage of this contingent view of identity is that it denies the so-called *Leibniz's Law*, which is generally accepted as an axiom for the identity relation. As a result, it equates the identity relation with any equivalence relation.

9.5 Theory of Conceptual Part-Whole Relations

Part-whole relations are fundamental concepts both from a cognitive perspective, for the realization of many important cognitive tasks (Tversky, 1989), and from an ontological perspective, serving as a foundational for the formalization of other entities that compose a foundational ontology.

Parthood has been represented in practically all conceptual/object-oriented modeling languages (e.g., OML, UML, EER, LINGO), and although it has not yet been adopted as a modeling primitive in the Semantic web languages, some authors have already pointed out its relevance for reasoning in description logics (e.g., Lambrix, 2000).

Nonetheless, in conceptual modeling, the concepts of part and whole are often understood only superficially. Consequently, the representation of these concepts in conceptual modeling languages is based on the very minimal axiomatization that these notions require. Moreover, despite it being an active topic in the conceptual modeling literature, there is still much disagreement on what characterizes this relation and about the properties that part-whole relations should have from a conceptual point of view.

In chapter 5 we propose an ontological well-founded theory of conceptual part-whole relations, which aims at providing precise formal characterizations and real-world semantics for these concepts. This theory is organized in a typology of parthood relations. The categories in this typology are generated in the following manner. First, we consider a number of theories of parts from formal ontology in philosophy (Mereologies). These formal theories provide an important starting point for the understanding and axiomatization of the notion of part. However, despite their importance, these theories concentrate merely on a formal notion of part and, hence, they fall short in fully characterizing the relation of parthood from a conceptual point of view. Thus as an extension to the mereological core of our typology, on the basis of the literature on meronymic relations in linguistic and cognitive sciences, we recognize the existence of not one but four types of part-whole relations, based on the type on ontological entities they relate. As a second dimension to create categories in this typology, we consider a number of meta-properties that part-whole can possess. These meta-properties designate: (i) whether objects can share parts; (ii) whether an object *only exists* being part of a specific whole (or of a whole of certain kind); (iii) whether an object *only exists* having a specific object as part (or a part of a specific kind). All these meta-properties are formally characterized using a modal logics approach.

The theory also shows that these two dimensions are not completely orthogonal, i.e., by selecting the type of a parthood relation based on the first dimension (i.e., based on the ontological category of its relata), a number of meta-properties on the second dimension are implied. In addition, each type of parthood selected in the first dimension determines a particular type of mereology (e.g., minimal, extensional) that should constitute the basic axioms of that specific relation.

This typology contributes to the theory of conceptual modeling by providing ontological foundations and real-world semantics for this important construct of conceptual modeling languages. Moreover, it helps to precisely characterize the relation between some of the entities forming this typology and one of the most important modeling concepts in conceptual modeling, namely, the concept of *roles* (see chapter 7). In particular, we use the distinction between *de re* and *de dicto* modality in the literature of philosophical logic to characterize the different formal properties of the relations of *specific dependent parthood* from the whole to the part, depending whether the whole universal is rigid or anti-rigid.

The explicit consideration of the modal meta-properties of part-whole relations together with those of the universals instantiated by their relata, constitute an important contribution not only to conceptual modeling, but also to the areas of software design and implementation. For example, inseparability and essentiality of parts in a part-whole relation constitute a sort of existential dependence relation, which imposes constraints on the relation between the life cycles of the relata. Understanding and making explicit these constraints is of fundamental importance not only for understanding and describing reality, but also, for example, for correctly specifying the relation between the life cycles of objects in object-oriented programming.

9.6 Contributions to the Problem of Transitivity in Conceptual Part-Whole Relations

The problem of transitivity of part-whole relations is a much debated topic not only in conceptual modeling but also in the linguistic and cognitive science literatures.

In many conceptual modeling languages (e.g., UML) part-whole relations are always considered transitive. However, as discussed in chapter 5, examples of fallacious cases of transitivity among part-whole relations abound.

In chapter 5, we discuss the relation between transitivity and the *unifying relations* characterizing the parts of a given whole. We conclude that if we consider a unique general sense of parthood, transitivity cannot be said to hold unrestricted, but only with respect to certain *contexts*. The delimitation of contexts, however, typically requires extensive knowledge of the domain being modeled.

In order to provide methodological assistance to the conceptual modeler in this task, we build on the different types of part-whole relations comprising the typology discussed in section 9.5. We demonstrate in chapter 5 that, by selecting the type of a part-whole relation in the first dimension aforementioned (i.e., according to the ontology category of its relata), we can determine which combination of parthood relations is transitive. The four types of part-whole relations considered in this respect are: (a) *subQuantityOf*; (b) *subCollectionOf*; (c) *memberOf*; (d) *componentOf*. As we have demonstrated, the combinations (a) + (a), (b) + (b) and (c) + (b) are always transitive. Thus, chains such as (a) + … + (a), or (c) + (b) + … + (b) will always delineate contexts. The combinations (c) + (c), (d) + (c), in contrast, are never transitive. Since most individuals represented in conceptual models are functional complexes, most part-whole relations represented are of the sort (d). This combination (d) + (d) is however itself non-transitive, and holds only in certain contexts. Other combinations such as (a) + (d), (b) + (d) are not possible. In these cases, the involved relations require relata of incompatible types. For instance, the combination *x is a subQuantity of y*, and *y is a componentOf z* is not possible since *y* cannot be both a quantity and a functional complex.

The precise definition of these contexts for functional parthood relations is of great importance not only for problem-solving in conceptual modeling, but also for software design and implementation. As discussed in (Odell & Bock, 1998), the transitivity of parthood relations has a strong impact on the propagation of properties and method invocations in object-oriented languages. For this reason, by building on the pioneering theory of transitivity of linguistic functional parthood relations proposed by (Vieu &

Aurnague, 2005), we propose a number of visual patterns that can be used to identify these contexts in conceptual class diagrams (chapter 7).

9.7 The *Role Modeling with Disjoint Admissible Types* Design Pattern

A known and much discussed problem in role modeling in the literature is the problem of specifying *admissible types* for Roles that can be filled by instances of disjoint types. This problem led (Steimann, 2000b), for example, to propose a complete separation of role and type hierarchies in conceptual models. As discussed in chapter 4, this solution implies a radical transformation to the metamodels of most of the current conceptual modeling languages.

By employing the theory of universals discussed in section 9.3, we propose an ontological design pattern capturing a standard solution to this problem. The adequacy of this design pattern is demonstrated by several examples throughout the thesis.

We believe that *ontological design patterns* such as this one, which captures standard solutions for recurrent conceptual modeling problems, represent an important contribution to the task of defining sound *engineering tools and principles* for the practice of conceptual modeling.

9.8 A Foundation for Harmonizing Different Concepts of Roles in the Literature

In (Wieringa et al., 1995), the authors offer an interesting alternative to the traditional notion of roles in conceptual and object-oriented modeling. They also propose a separation of role and type hierarchies, but for radically different reasons. Their main motivation is based on a philosophical problem known as *The Counting Problem* (Gupta, 1980). As we have shown in chapter 7, this problem is actually fallacious and, thus, the separation of role and type hierarchies cannot be argued for on this basis. Nonetheless, there is an important truth highlighted by their argument which is generally neglected in most conceptual modelling approaches, namely, that in different situations one might want to count "role instances" in different senses.

The notion of role proposed by Wieringa et al. is one in which role individuals are entities that are *existentially dependent* of their players. Moreover, role universals are responsible for supplying principles of identity for their instances, which are different from the ones supplied by the universals of their players. This view is also shared by (Loebe, 2003).

By relying of the ontological categories of *moments* and *qua individuals* discussed in chapter 6, we manage to provide an ontological interpretation for the notion of roles proposed by Wieringa et al. and Loebe. Moreover, we manage to harmonize it with the more common view of roles taken in the literature (and, in particular, in the typology of section 9.3), and the one which more naturally represents the commonsense use of roles in ordinary language, namely, the conception of roles as *relationally dependent and anti-rigid substantial universals*.

Finally, by explicitly representing roles as both substantial universals and *qua individual* universals, we can account in an unambiguous way for the alternative senses of counting "role instances" previously mentioned.

9.9 An Aristotelian Ontology for Attributes, Weak Entities and Relationships

In chapter 6, we present the core of the foundational ontology proposed here, organized in terms of the so-called *Aristotelian ontological square* comprising the category pairs *Substantial-Substantial Universal*, *Moment-Moment Universal*. This ontology constitutes an important contribution to the theory of conceptual modeling, by enabling simple, unambiguous and ontologically interesting real-world semantics to be defined for the conceptual modeling categories of *attributes*, *weak entities* and *relationships*.

From a metaphysical point of view, these categories allow for the construction of a parsimonious ontology, based on the primitive and formally defined notion of *existential dependency*. This has been possible since, (i) by using this existential dependency meta-property we can differentiate which particulars in the domain are *substantials* and which are *moments*; (ii) based on the multiplicity of entities a moment depends on, we can distinguish between *qualities* and *relators*; (iii) with qualities we can explain *formal relations* and *weak entities*, and with relators, *material relations*; (iv) finally, with moments we can explain the relation of *exemplification*, and with exemplification *characterization*.

From a conceptual point of view, this ontology allows for the creation of a conceptual modeling language with great simplicity and preciseness of its constructs. For example, in the UML profile generated in chapter 8 all represented universals are either universals of independent entities (substantial universals), or universals which instances are existentially dependent of other instances (moment universals). In addition, all material associations can be reduced to primitive relations of existential dependence.

As a consequence, the user of this modeling language is guided in the choice of how to model the elements of the universe of discourse by a formally defined and, thus, unambiguous relation of existential dependence.

As discussed in chapter 6, despite being ubiquitous in conceptual modeling languages, relationships are particularly problematic and ambiguous constructs. However, by employing (explicitly represented) relators, we can provide not only an ontologically well-founded interpretation for them, but also one that can accommodate more subtle linguistic distinctions. Additionally, from a conceptual point of view, we can produce models that are free of cardinality constraint ambiguities, i.e., models in which *single-tuple* and *multiple-tuple* cardinality constraints are explicitly differentiated and represented. Furthermore, with the notions of qualities and relators we can define *qua individuals*, and with the latter we have managed to provide a foundation and terminological clarification, which harmonizes two competing notions of roles present in the conceptual modeling literature (see section 9.8). Furthermore, the ontological category of relators shall play an important role in extending our ontology with: (i) *an ontology of perdurants*: relators are typically founded on individual *processes* or *events* and, as discussed in chapter 6, they can play an important role as a mechanism for consistency preservation between static and dynamic conceptual models; (ii) *an ontology of social and intentional entities*: many social bonds (e.g., commitments, claims) are types of relators (Guizzardi & Wagner, 2005a). Moreover, the notion of relators presented here is akin to the one of affordances in organizational semiotics (Stamper et al., 2000).

9.10 Ontological Foundations for Attributes Values and Attribute Value Spaces

Traditionally, in conceptual modeling, value domains are taken for granted. In general, in conceptual modeling languages, they are only superficially considered by taking primitive datatypes as representing familiar mathematical sets (e.g., natural, integer, real, Boolean) and the focus has been almost uniquely on mathematical specification techniques.

In chapter 6, besides the entities that constitute the four-categorical ontology aforementioned, we explicitly take into account the *conceptual measurement structures* in which particular qualities are perceived (and conceived). By employing the theory of *conceptual spaces*, we can provide an ontological interpretation for the conceptual modeling notion of *attribute values* and *attribute value domains*.

As discussed in chapter 6, many benefits arise from the use of *conceptual spaces* and related notions (e.g., *quality domains* and *quality dimensions*), as theoretical tools for conceptualizing attribute values and value domains. Firstly, quality domains and the constraints relating different quality dimensions captured in its structure can provide a sound basis for the conceptual modeling representations of the associated structured datatypes, constraining the possible values that their fields can assume. In other words, whatever constraints should be specified for a datatype, they must reflect the geometrical structure of the quality domain underlying this datatype.

Moreover, as demonstrated in chapters 6 and 7, from the structure of a certain quality domain, one can derive meta-properties (e.g., reflexivity, asymmetry, transitivity) for the formal relations based on this domain. That is to say, explicitly represented characteristics of conceptual spaces can provide an ontological explanation for these meta-properties.

Other approaches to ontology-based conceptual modeling in the literature (e.g., Evermann & Wand, 2001b) propose that the conceptual modeling representations of *natural Kinds* (a type of *substance sortals*) should be accompanied by a constraint specification, which restricts the possible values that the attributes of a substantial thing can assume, i.e., which delineates the *lawful state space* of that kind. As discussed in chapter 6, a conceptual space associated with a substance sortal is the cognitive counterpart of the lawful state space of that substance sortal. However, differently from the *lawful state spaces*, the *conceptual spaces* approach acknowledges that *quality domains* can themselves be multidimensional, exhibiting a constrained substructure that can occur in the definition of conceptual spaces associated with possibly different substantial universals. For this reason, we believe that the explicit representation of quality domains (associated with quality universals) as *datatypes* not only provides a further degree of structuring on lawful state spaces, but it also allows for a potential reuse of specifications of a subset of its constraints.

Finally, the explicit account of conceptual spaces and their constituent domains and dimensions afford the conceptualization of alternative measurement structures for a given ontological entity. Furthermore, the possibility of defining transformations and projections between these explicitly represented structures facilitates the tasks of knowledge sharing and semantic interoperability.

9.11 An Ontologically Well-Founded UML version for Conceptual Modeling and Ontology Representation

The ontology constructed throughout chapters 4 to 7 of this thesis is aimed at providing ontological foundations for conceptual modeling concepts and real-world semantics for the related constructs. The choice of concepts on which we focus in this ontology (types and their instances, type taxonomies, associations, attributes, attribute values and attribute value spaces, part-whole relations) was strongly influenced by their importance in the practice of conceptual modeling, but also by their generality as being represented in constructs of existing conceptual modeling languages.

In chapter 8, we apply this foundational ontology as a reference for analyzing the ontological appropriateness of the Unified Modeling Language (UML) for the purpose of conceptual modelling and ontology representation. This effort is meant first as a case study of the language evaluation and (re)design framework proposed here. However, due to the following reasons, the ontologically well-founded re-designed version of UML proposed represents itself an important research contribution of this thesis: (i) the current status of UML as *de facto* standard modeling language; (ii) the growing interest in its adoption as a language for conceptual modelling and ontology representation; (iii) the current scarcity of *foundational* (as opposed to *lightweight*) conceptual modelling languages.

We emphasize, however, that this foundational ontology is *independent* of particular modeling languages and, hence, it can in principle be used as a foundation for analyzing any structural conceptual modeling language (e.g., EER, OML, LINGO, OWL). For instance, in (Guizzardi & Wagner, 2005a), an extension of the foundational ontology proposed here has been employed to analyze the ontological semantics of some enterprise modeling languages and frameworks (e.g., the REA framework) and, in (Wagner & Taveter, 2004), as the ontological foundation of the AORML agent-modeling language.

By using the evaluation and (re)design framework proposed here, we have managed to demonstrate cases in which each of the properties of a suitable representation mapping is lacking in the current UML meta-model when interpreted in terms of our reference ontology. In other words, as a conceptual modelling and ontology representation language, the UML meta-model can be shown to contain cases of construct *incompleteness*, *overload*, *redundancy* and *excess*.

In order to remedy this situation, we have proposed a number of modifications to the UML meta-model, producing a conceptually cleaner, semantically unambiguous and ontologically well-founded version of the language. Moreover, since the constructs in the re-designed meta-model can be unambiguously identified by formally defined ontological meta-properties, this approach provides the important additional benefit of methodologically supporting the user of the language in deciding how to model the elements of the universe of discourse.

We believe that one should attempt to shield as much as possible the user of a conceptual modelling language from the complexity of the underlying ontological theory. Therefore, it is our strategy to (whenever possible) represent the ontological principles underlying a language in terms of *syntactical constraints* of this language. For example, since the *postulate 4.2* of our typology of universals (see section 9.3) is represented as a constraint in the redesigned UML meta-model, a user's model in which a class regarded as anti-rigid (e.g., customer) is a subclass of one regarded as rigid (e.g., person) is a *syntactically (grammatically) incorrect model*. Therefore, even if a conceptual modeller is unlearned about the ontological foundations of a given language, he can still produce only ontologically correct models (*intended models*) just by being accurate in the use of that language's grammar.

Finally, we emphasize that this strategy follows an analogous process occurring in human natural languages, since, according to (Chomsky, 1986), the grammars of natural languages have the properties they do to reflect the properties of ontological categories underpinning human cognition. Thus, as exemplified in chapter 4, some natural language constructions are ungrammatical for ontological reasons (e.g., only general terms representing sortal universals can be combined with quantifiers and determiners).

9.12 A Case Study on the Integration and Analysis of Semantic Web Ontologies

In chapter 3, we discuss the two-level approach defended in this thesis for semantic interoperation of conceptual models. We defend that in a first phase of off-line meaning negotiation, an expressive and ontologically adequate modelling language should be used for comparing the candidate models and making explicit their underlying ontological commitments. Once this meaning negotiation and semantic interoperation phase is complete, then a computationally efficient representation language can be used to express the results of this phase.

In chapter 8, we exemplify the first phase of this process by using the UML version proposed here to analyze and integrate several Semantic web lightweight ontologies. The scenario analyzed is on the integration of Semantic web ontologies in the scope of a context-aware service platform. This scenario was originally proposed in (Ríos, 2003) to illustrate the inadequacy of Semantic web languages in making explicit the underlying ontological commitments of the conceptualizations involved. Here, this scenario has been extended by considering real Semantic web models, which are structurally similar to the ones proposed by Ríos, and which are used by practitioners in concrete Semantic web applications.

As shown in chapter 8, through this case study, we have managed to demonstrate the adequacy of the language proposed and, consequently, of its underlying foundational ontology. In particular, this case study exemplifies the importance of the results achieved here to Semantic web-related projects such as A-MUSE[80] (Architecture Modelling Utility for Service Enabling) and AWARENESS[81] (AWARE Mobile NEtwork and ServiceS).

Finally, in chapter 2 we have illustrated the design of a domain-specific modelling language based on a domain ontology of geneology. In that chapter, this example is presented as an illustration of the language evaluation and (re)design framework proposed. However, as discussed there, the derivation of suitable syntactical and pragmatic constraints for the designed language depends on suitability of the domain ontology represented. Therefore, that example can now be seen also as an additional exemplification of the usefulness of the modelling language and the ontological concepts proposed here.

9.13 Future Work

In this thesis, we aim at developing ontological foundations for structural conceptual modeling concepts. As a result, the foundational ontology proposed is centered in the ontological category of endurants and endurant universals. In (Guizzardi & Wagner, 2004; 2005a; 2005b), we present an extension of this ontology as a component of a larger one named UFO (Unified Foundational Ontology).

UFO is organized in three incrementally layered *compliance sets*:

1. UFO-A: essentially the foundational ontology proposed here;
2. UFO-B: defines, as an increment to UFO-A, terms related to perdurants;
3. UFO-C: defines, as an increment to UFO-B, terms explicitly related to the spheres of intentional and social things, including linguistic things.

This division reflects a certain stratification of our "world". It also reflects different degrees of scientific consensus: there is more consensus about the ontology of endurants than about the ontology of perdurants, and there is more consensus about the ontology of perdurants than about the ontology of intentional and social things.

[80] http://a-muse.freeband.nl/
[81] http://awareness.freeband.nl/

The usefulness of this extended ontology has been already demonstrated in analyzing agent related concepts (Guizzardi & Wagner, 2004; 2005b) and enterprise modeling languages (Guizzardi & Wagner, 2005a). Nevertheless, UFO-B and UFO-C are still in a preliminary stage of development and, thus, substantial work still needs to be done in understanding and formally characterizing the entities constituting these ontologies.

As discussed in chapter 3, in domain engineering, ontology representation languages can play an important role, providing useful abstractions for representing domain models. However, once a domain model (domain ontology) is represented, there is still the problem of systematically deriving from it reusable artifacts (e.g., frameworks), which preserve the characteristics of the conceptualization expressed in that model. The same situation occurs in traditional software engineering, in which design and implementation specifications should also preserve the real-world semantics of the domain conceptualization (captured in a conceptual model of the domain). This problem is known as the *impedance mismatch problem* (Guizzardi & Falbo & Pereira Filho, 2001a).

In (Guizzardi & Falbo & Pereira Filho, 2001a,b; 2002) and (Falbo & Guizzardi & Duarte, 2002), we propose a systematic and semantics-preserving approach for deriving object-oriented frameworks from domain ontologies expressed in LINGO. The derivation methodology proposed comprises a spectrum of techniques, namely, mapping directives, design patterns and formal translation rules. For example, in (Guizzardi & Falbo & Pereira Filho, 2002) a design pattern is proposed that guarantees the preservation of some ontological properties of part-whole relations (irreflexivity, asymmetry, transitivity and shareability) in object-oriented implementations.

When evaluated in terms of the foundational ontology proposed here, LINGO can be shown to be seriously incomplete. A natural extension of the work aforementioned is the investigation of similar methodological mechanisms for deriving object-oriented implementations and frameworks that preserve the ontological properties of domain models expressed in the conceptual modeling language proposed here.

There are also interesting possibilities for future work on the topic of domain-specific visual languages evaluation and (re)design. Firstly, a more systematic account is needed of the connection between ontological meta-categories (and their meta-properties) and, systems of visual signs. An interesting approach to pursue this topic would be in the lines of the *Algebraic Semiotics* program proposed in (Goguen, 1999), since such an approach should consider a more comprehensive exploration of semiotic theory, but doing it in a precise way. However, differently from the latter, domain models should be suitably represented in terms of expressive domain ontologies, in which ontological meta-properties of the elements of a conceptualization are explicitly represented, as opposed to focusing solely on abstract (algebraic and category-theoretical) domain specifications.

The establishment of systematic connections between real-world semantic domains, meta-models, and visual systems of concrete syntaxes shall provide a solid basis for the construction of semantic aware (meta) modeling case tools for domain-specific visual language evaluation and (re)design. Additionally, with the development of systematic approaches for deriving frameworks from expressive domain ontologies (along the lines previously discussed) these tools can also support the creation of libraries of reusable domain artifacts, thus, bridging the gap between domain-specific conceptual modeling and design/implementation. One should refer to (Falbo et al., 2002b; Falbo et al., 2003) for an example of an ontology-based semantic software engineering environment based on the frameworks derivation approach aforementioned.

In this thesis, we have shown many conceptual modeling benefits resulting from the application of the *theory of conceptual spaces* in the foundation ontology proposed here. However, there are many other aspects of this theory, which were not explored in this work, but which shall be pursued in future investigations.

For example, as briefly discussed in chapter 6, conceptual spaces can be used for the construction of *Portions* (Gerlst & Pribbenow, 1995), i.e., dynamically assembled collectives built by selecting certain parts of an integral object, according to certain internal properties of this object. Examples include: (a) the reddish parts of an object; (b) the tall animals in the group. The notion of *contrast classes* in the theory of conceptual spaces (Gardenfors 2000; 2004) can allow for interesting cases of non-monotomic reasoning in situations such as depicted in (b). For instance, although every squirrel is an animal, a tall squirrel is not a tall animal. As argued by (Gerlst & Pribbenow, 1995), *Portions* can play an important role in reasoning about physical objects. In particular, they can be useful in *context-dependent reasoning*, by supporting the construction of dynamically created aggregates depending on a given context (e.g., the crowded areas of a geographical location, the restored parts of a museum).

Another aspect of this theory that could be useful in context-aware scenarios is the notion of *context matrices*, which can be used to emphasize certain quality dimensions of conceptual spaces in detriment of others depending on the circumstance. Take, for instance, a *wayfinding service*. In a *day context*, the best wayfinding landmark to be used by this service could be a façade with the most contrasting color. In contrast, in a night context, it can be simply the highest or widest landmark (Raubal, 2004).

Finally, conceptual spaces provide a sound mechanism for accounting for relations such as *resemblance, complementation*, and *opposition* between universals. As a consequence, one can also account for relations such as these between individuals instantiating those universals. Besides the theoretical importance of this mechanism for a theory of conceptual modeling (as discussed in chapter 6), it allows for the construction of conceptual models which can support approximate (inexact) reasoning applications (e.g., *inexact service matching*). This topic should also be explored in future research.

Besides contributing to the development of a philosophically and cognitive principled theory of conceptual modeling, it is also a general objective of this thesis to work towards the development of sound practical tools for the disciplines of conceptual modeling and ontological engineering. We believe that the development of these tools constitutes an important step, contributing to the maturity of these disciplines as engineering disciplines.

In this thesis, we have pursue this latter goal in several manners, for instance, by proposing a systematic evaluation and (re)design method, by providing methodological guidelines for helping the user in deciding how to model elements of given conceptualization, by proposing a modeling language in which ontological principles are incorporated as the language's syntax constraints, or by proposing formal languages in which these ontological principles are embedded in the language's semantic theory. Moreover, an important contribution of this thesis in this direction is the creation of an *ontological design pattern* that captures a situation-independent solution for a classical and recurrent problem in conceptual modeling. With the extension of the ontological theory proposed here to incorporate other aspects of reality (UFO-B and UFO-C), we shall continue to investigate the development of conceptual modeling engineering tools in all these directions.

References

Abrial, J-R. 'Data semantics', Klimbie and Koffeman (eds.) Data Management Systems, North-Holland, 1974.

Aguiar, A., Baillargeon, R. '2.5-month-old infants_ reasoning about when objects should and should not be occluded'. Cognitive Psychology, 39, 116–157, 1999.

Albano, A.; Bergamini, R.; Ghelli, G.; Orsini, R. 'An object data model with roles', in: R. Agrawal, S. Baker, D. Bell (Eds.), Proceedings of the 19th International Conference on Very Large Databases, Morgan Kaufmann, Dublin, pp. 39-51, 1993.

Alberts L. K. 'YMIR: a sharable ontology for the formal representation of engineering design knowledge', Formal Design Methods for Computer-Aided Design, Elsevier/IFIP, 1994.

Almeida J.P.A; Guizzardi,G.; Pereira Filho, J.G. 'Agent-Mediators in Media on Demand Eletronic Commerce', VII International Congress of new technologies and Computer Applications, Cuba, 2000.

Angelelli, I. 'On identity and interchangeability in leibniz and frege'. Notre Dame Journal of Formal Logic, 8(1-2):94–100, 1967.

Anglin, J.M. 'Word, Object and Conceptual Development', New York, Norton, 1977.

Anscombe, G. E. M. 'Under a description'. Nous, 13(2), 1979.

Arango, G, Williams, G., and Iscoe, N. 'Domain Modeling for Software Engineering', Proceedings of ICSE 1991: The International Conference on Software Engineering. ACM Press, Austin, Texas, 1991.

Arango, G. 'Domain Analysis Methods', Workshop on Software Architecture, Los Angeles. Los Angeles: USC Center for Software Engineering, 1994.

Arango, G., Prieto-Diaz, R. 'Domain Analysis Concepts and Research Directions'. Workshop on Software Architecture, USC Center for Software Engineering, Los Angeles, EUA. Los Angeles,1994.

Aristotle. 'Complete Works', edited by Jonathan Barnes, 2 vols., Princeton: Princeton University Press, 1984.

Armstrong, D.M. 'A World of State of Affairs', Cambridge Studies in Philosophy, Cambridge University Press, 1997.

Armstrong, D.M. 'Universals: An Opinionated Introduction', Westview Press, 1989.

Artale, A., Franconi, E., Guarino, N., and Pazzi, L., 'Part-Whole Relations in Object-Centered Systems: an Overview', Data and Knowledge Engineering, 20(3), 1996, pp. 347-383.

Baader F., Horrocks, I., Sattler, U. 'Description logics as ontology languages for the semantic web'. Lecture Notes in Artificial Intelligence. Springer, 2003.

Baclawski K. et al. 'Extending UML to Support Ontology Engineering for the Semantic Web', Fourth International Conference on the Unified Modeling Language: UML 2001, Toronto, Canada, 2001.

Bacon, J. 'Universals and Property Instances. The Alphabet of Being', Oxford: Blackwell, 1995.

Baek, A. 'Syllogism with reduplication in Aristotle'. Notre Dame Journal of Formal Logic, 23(4):453–458, 1982.

Balaguer, M. 'Platonism in Metaphysics', Stanford Encyclopedia of Philosophy, [online: http://plato.stanford.edu/entries/platonism/], capture in 2004.

Baldinger, K. 'Semantic Theory: Towards a Modern Semantics', Palgrave Macmillan, 1980.

Batra, D.; Hoffler, J. A.; Bostrom, R. P. 'Comparing representations with relational and EER models', Communications of the ACM, v.33 n.2, p.126-139, Feb. 1990

Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuiness, D.; Patel-Shneider, P.; Stein, L.A., 'OWL Web Ontology Language Reference', W3C Recommendation, Feb. 2004, [online: http://www.w3.org/TR/owl-ref/], 2004.

Berners-Lee T. 'Information Management: A Proposal'. CERN Report, [online: http://www.w3.org/History/1989/proposal.html], 1989-1990.

Berners-Lee T., Hendler, J., Lassila, O. 'The Semantic Web', Scientific American, May, 2001.

Bézivin, J., Gerbé, O., 'Towards a Precise Definition of the OMG/MDA Framework', ASE'01, Automated Software Engineering, San Diego, USA, November 26-29, 2001.

Bock, C.; Odell, J. 'A More Complete Model of Relations and Their Implementation: Relations as Object Types', Journal of Object-Oriented Programming Vol 10, No 3. June, 1997a.

Bock, C.; Odell, J. A 'More Complete Model of Relations and Their Implementation: Mappings', Journal of Object-Oriented Programming, October, pp.28-30, 1997b.

Bock, C.; Odell, J.: 'A More Complete Model of Relations and their Implementation: Roles'. Journal of OO Programming, May, 1998, 51–54.

Bodart, F., Patel, A., Sim, M., Weber, R. 'Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests', Information Systems Research, Vol.12, No. 4, December, pp.384-405, 2001.

Bonatti, L., Frot, E., Zangl, R., & Mehler, J. 'The human first hypothesis: identification of conspecifics and individuation of objects in the young infant'. Cognitive Psychology, 44, 388–426, 2002.

Booch, G., 'Object-Oriented Analysis and Design', Benjamin-Cummings, 1994.

Booth, Amy E. & Waxman, Sandra R. 'Mapping words to the world in infancy: Infants' expectations for count nouns and adjectives'. Journal of Cognition and Development, 4(3), 357-381, 2003.

Borgida, A. 'Knowledge Representation, Semantic Data Modeling: What's the Difference?', Proceedings of the 9th International Conference on Conceptual Modeling, Lausanne, Switzerland, 1990.

Borst, W. N. 'Construction of Engineering Ontologies', Centre for Telematica and Information Technology, University of Twente, Enschede, The Netherlands, 1997.

Botazzi, E., Catenacci, C., Gangemi, A., Lehmann, J. 'From Collective Intentionality to Intentional Collectives: An Ontological Perspective', revised version of the paper presented at Collective Intentionality IV, Siena, 2004. Submitted to Cognitive Systems Research, Special Issue on Cognition and Collective Intentionality, Elsevier, 2006.

Bottoni, P., Minas, M. editors 'Graph Transformation and Visual Modeling Techniques', volume 72/3 of Electronic Notes in Theoretical Computer Science. Elsevier Science Publishers, 2003.

Boyd, R. 'Realism, Anti-Foundationalism and the Enthusiasm for Natural Kinds'. Philosophical Studies 61: 127–148, 1991.

Brachman, R. J. 'On the Epistemological Status of Semantic Networks', In N. V. Findler (Ed.), Associative Networks: Representation and Use of Knowledge by Computers.Academic Press, 1979.

Brachman, R. J. and J. G. Schmolze 'An Overview of the KL-ONE Knowledge Representation System', Cognitive Science 9, 171-216. Carrara, M. 1992. Identità e persona nella riflession, 1985.

Breu, R.; Hinkel, U.; Hofmann, C.; Klein, C.; Paech, B.; Rumpe, B.; Thurner, V. 'Towards a Formalization of the Unified Modeling Language', Proceedings fo the 11th European Conference on Object-Oriented Programming (ECOOP), Jyväskylä, Finland, 1997.

Brickley, D.; Miller, L. 'FOAF Vocabulary Specification', revision 1.47 edition, 2003.

Bunge M. 'Ontology I: The Furniture of the World', Treatise on Basic Philosophy, vol. 3, D. Reidel Publishing, New York, 1977.

Bunge M. 'Ontology II: A World of Systems', Treatise on Basic Philosophy, vol. 4, Reidel Publishing Company, Dordrecht, Holland, 1979.

Campbell, K. 'Abstract Particulars', Blackwell, Cambridge, 1990.

Caplinskas, A. 'An Ontology-based Approach to Enterprise Engineering', Emerging Database Research in East Europe, Proceedings of the Pre-Conference Workshop of VLDB 2003.

Cartwright, H. M. 'Heraclitus and the Bath Water'. Philosophical Review 74, 1965.

Casati, R.; Varzi, A.C. 'Holes and Other Superficialities', Cambridge, Mass.:MIT Press, 1994.

Chen S.L. 'Information technology in business processes', Business Process Management Journal, Vol. 6 No. 3, 2000.

Chen, H. L. 'An Intelligent Broker Architecture for Pervasive Context-Aware Systems', PhD Thesis, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County (UMBC), USA, 2004.

Chen, H. L.; Finin, T.; Joshi, A. 'An Ontology for Context-Aware Pervasive Computing Environments', Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, 18(3):197–207, 2004.

Chen, H. L.; Perich, F.; Finin, T.; Joshi, A. 'SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications', In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004), Boston, MA, August 22-26, 2004.

Chen, H., Finin, T. and Joshi, A. 'The Role of the Semantic Web in Pervasive Context-Aware Systems', Draft submitted to ISWC03, Sanibel Island, Florida, USA, October 2003.

Chen. P. 'The entity-relationship model: Towards a unified view of data', ACM Transactions on Database Systems 1(1), 1976.

Chisholm, R.M. 'A Realistic Theory of Categories: An Essay on Ontology', Cambridge: Cambridge University Press, 1996.

Chomsky, N. 'Aspects of a Theory of Syntax'. Cambridge, Massachusetts, MIT Press, 1965.

Chomsky, N. 'Knowledge of Language: Its Nature, Origin and Use'. New York, Praeger, 1986.

Chomsky, N. 'Rules and Representations'. New York, Columbia University Press, 1980.

Chu, W.W.; Zhang, G. 'Associations and roles in object-oriented modeling', in: D.W. Embley, R.C. Goldstein (Eds.), Proceedings of the 16th International Conference on Conceptual Modeling: ER'97, Springer, Berlin, pp. 257-270, 1997.

Ciocoiu, M., Nau D. 'Ontology-Based Semantics', In: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000), Breckenbridge, Colorado, April 12-17, 2000.

Ciocoiu, M.; Gruninger, M. 'Ontologies for Integrating Engineering Applications', Journal of Computing and Information Science in Engineering 1(1): 12-22, 2000.

Clancy, W. J. 'The Knowledge Level Reinterpreted: Modelling Socio-Technical Systems', International Journal of Intelligent Systems, 8: 33-49, 1993.

Cocchiarella, Nino 'Logical Investigations of Predication Theory and the Problem of Universals'. Napoli: Bibliopolis, 1986.

Codd, E.F. 'A Relational Model of Data for Large Shared Data Banks', Communications of the ACM, June, p.377-387, 1970.

Cohen, L.B.; Younger, B.A. 'Perceptual Categorization in the Infant, in New Trends in Conceptual Representation', E.K.Scholnick (ed.), Hillsdale, New Jersey, Erlbaum, 197-200, 1983.

Corazzon, R.; 'Notes on the History of Ontology, Descriptive and Formal Ontology: A resource guide to contemporary research', [online: www.formalontology.it], captured in 2004.

Costa, P.D. 'Towards a Service Platform for Context-Aware Applications', Master Dissertation in Telematics, University of Twente, The Netherlands, 2003.

Costa, P.D.; Ferreira Pires, L.; van Sinderen, M.; Filho, J.G.P. 'Towards a Services Platform for Mobile Context-Aware Applications', First International Workshop on Ubiquitous Computing (IWUC 2004), Portugal, April 2004a.

Costa, P.D.; Ferreira Pires, L.; van Sinderen, M.; Rios, D. 'A services platform for context-aware applications', 2nd European Symp. on Ambient Intelligence (EUSAI 2004), LNCS 3295, 8-10 November 2004, Eindhoven, The Netherlands, 2004b.

Cota, R.I., Menezes, C.S., Falbo, R.A. 'Organizational Modeling using Ontologies and Analysis Patterns' (in Portuguese: Modelagem Organizacional Usando Ontologias e Padroes de Analise), VII Iberoamerican Workshop on Requirements Engineering and Sofware Development Environments (IDEAS'2004), Arequipa, Perú, 2004.

Cox, S.; Daisey, P.; Lake, R; Portele, C.; Whiteside, A. 'Geography markup language (gml 3.0)'. In OpenGIS Documents. OpenGIS Consortium, 2003.

Cranefield, S., Purvis M. 'UML as an ontology modelling language', Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99), Germany, University of Karlsruhe, 46-53, 1999.

Cruz, I. 'NSF Information and Data Management Program Report on the International Semantic Web Working Symposium (SWWS)', Stanford University, California, USA, July 30 - August 1, 2001.

Davidson, D. 'The Logical Form of Action Sentences'. In [28], pp. 105-148, 1980.

Davies, J., Fensel, D., & van Harmelen, F. 'Towards the Semantic Web: Ontology-driven Knowledge Management', West Sussex: Wiley, 2003.

Davis, R., Shrobe, H. & Szolovits, P. 'What Is a Knowledge Representation?', AI Magazine, Spring, 1993.

de Saussure, F. 'Course in General Linguistics', Roy Harris (trans.), Open Court Publishing Company, 1986 (original from 1916).

Degen, W., Heller B., Herre H., Smith, B. GOL: 'Towards an axiomatized upper level ontology'. In B. Smith and N. Guarino (eds.), Proc. of 2nd Intl. Conf. of Formal Ontologies and Information Systems (FOIS'01), Ogunquit, Maine, USA, ACM Press, 2001.

Devedžić, V. 'Understanding Ontological Engineering', Communications of the ACM, April 2002, Vol.45, no. 4ve.

Dignum, V. 'A model for organizational interaction: based on agents, founded in logic', PhD Thesis, University of Utrecht, The Netherlands, 2003.

Dijkman, R.M., Ferreira Pires, L., Joosten, S.M.M. 'Calculating with Concepts: A Technique for the Development of Business Process Support', In A. Evans, R. France, A. Moreira, and B. Rumpe (eds.): Proceedings of the UML 2001 Workshop on Practical UML-Based Rigorous Development Methods, Toronto, Canada, pp. 87-98, 2001.

Eimas, P.; Quinn, P. 'Studies on the formation of perceptually-based basic-level categories in young infants'. Child Development, 65, 903-917, 1994.

Elmasri, R; Weeldreyer, J.A.; Hevner, A.R. 'The Category Concept: An Extension to the Entity-Relationship Model', International Journal on Data and Knowledge Engineering, 1(1):75-116, 1985.

Ereshefsky, M. 'Species', Stanford Encyclopedia of Philosophy, [online: http://plato.stanford.edu/entries/species/], 2002.

Erwig, M. 'Visual Graphs', 15th IEEE Symposium on Visual Languages, 122-129, 1999.

Essink, L.J.B.; Erhart, W.J. 'Object modelling and system dynamics in the conceptualization stages of information systems development', in: F. van Assche, B. Moulin, C. Rolland (Eds.), Proceedings of the IFIP TC8/WG8.1. Working Conference on the Object Oriented Approach in Information Systems, North-Holland, Amsterdam, pp. 89-116, 1991.

Euzenat, J. 'Towards a principled approach to semantic interoperability'. In: Proceedings of the IJCAI-01, Workshop on Ontologies and Information Sharing, Seattle, USA, 2001.

Evans, A. and Kent, S. 'Core meta-modelling semantics of UML: the pUML approach'. In UML'99 The Uni_ed Modeling Language – Beyond the Standard: Second International Workshop, Fort Collins, CO, October 28-30, 1999.

Evermann, J. and Wand, Y. 'An ontological examination of object interaction in conceptual modeling', In Proceedings of the Workshop on Information Technologies and Systems WITS'01, New Orleans, December 15-16, 2001a.

Evermann, J. and Wand, Y. 'Towards ontologically based semantics for UML constructs', In H. Kunii, S. Jajodia, and A. Solvberg, editors, Proceedings of the 20th International Conference on Conceptual Modeling, Yokohama, Japan, Nov. 27-30, 2001b.

Falbo, R. A., Guizzardi, G., Duarte, K. C. 'An Ontological Approach to Domain Engineering', In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE), Ischia, Italy, 2002.

Falbo, R. A., Menezes, C. S., Rocha, A. R. C. 'A Systematic Approach for Building Ontologies', In Proceedings of the Sixth Iberoamerican Conference on Artificial Intelligence (IBERAMIA), Lisboa, Portugal, 1998.

Falbo, R.A., Natali, A. C. C., Mian, P. G., Bertollo, G., Ruy, F. B. 'ODE: Ontology-based software Development Environment', IX Argentine Conference on Computer Science, La Plata, Argentina, Outubro 2003.

Falbo, R.A., Ruy, F.B., Bertollo, G., Togneri, D.F. 'Learning How to Manage Risks Using Organizational Knowledge', Advances in Learning Software Organizations, Proceedings of the 6th International Workshop on Learning Software Organizations (LSO'2004), Melnik G. and Holz, H. (Eds.): LNCS 3096, pp. 7-18, Springer-Verlag Berlin Heidelberg, Banff, Canada, June 2004.

Falbo, R.A.; Guizzardi, G.; Duarte, K.C.; Natali, A.C.C. 'Developing Software for and with Reuse: An Ontological Approach', ACIS International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA-02), Foz do Iguacu, Brazil, 2002a.

Falbo, R.A.; Guizzardi, G.; Natali, A.C.; Bertollo, G.; Ruy F.; Mian P. 'Towards Semantic Software Engineering Environments', XIV International Conference on Software Engineering and Knowledge Engineering (SEKE-2002), Ischia, Italy, 2002b.

Fensel, D.; Sycara, K.;Mylopoulos, J.(Eds.): 'The SemanticWeb - ISWC 2003', Second International SemanticWeb Conference, Sanibel Island, FL, USA, October 20-23, 2003. published as Lecture Notes in Computer Science, Vol.2870, Springer-Verlag, Berlin.

Fensel, D; Horrocks, I; van Harmelen, F.; McGuinness, D.L.; Patel-Schneider, P. 'OIL: An Ontology Infrastructure for the Semantic Web'. In IEEE Intelligent Systems, Vol. 16, No. 2, March/April 2001.

Fernández-López, M., Gómez-Pérez, A., Pazos-Sierra, A., Pazos-Sierra, J. 'Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment'. IEEE Intelligent Systems & their applications. January/February Pages 37-46, 1999.

Ferrario, R.; Oltramari, A. 'Towards a Computational Ontology of the Mind', Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS), Torino, Italy, 2004.

Ferreira Pires, L. 'Architectural Notes: A Framework for Distributed Systems Development', PhD Thesis, Centre for Telematics and Information Technology, University of Twente, The Netherlands, 1994.

Ferreira Pires, L., van Sinderen, M., de Farias, C. R. G., Almeida, J. P. A. 'Use of Models and Modeling Techniques for Service Development', in Manuel J. Mendes, Reima Suomi and Carlos Passos (Eds.), Digital Communities in a Networked Society: eCommerce, eGovernment and eBusiness, Kluwer Academic Publishers, pp. 441-456, 2004.

Fettke, P. and Loos, P. 'Ontological evaluation of reference models using the Bunge-Wand-Weber model', In Proceedings of the 2003 Americas Conference on Information Systems, August 4-6, Tampa, Florida, 2003.

Fielding, J.M.; Simon, J.; Ceusters, W.; Smith, B. 'Ontological Theory for Ontology Engineering'. in: Proceedings of KR 2004, Ninth International Conference on the Principles of Knowledge Representation and Reasoning, Whistler, Canada, 2004.

Fine, K.. 'Acts, events and things'. In Sixth International Wittgenstein Symposium, Kirchberg-Wechsel (Austria), pages 97–105, 1982.

Fischer Nilsson, Jørgen (1999): 'A Conceptual Space Logic'. In Proceedings of the 9th European-Japanese Conference on Information Modelling and Knowledge Bases, pp. 39-53. Hachimantai, Iwate, Japan.

Fitting, M., Mendelsohn, R.L. 'First-Order Modal Logic', Synthese Library Studies in Epistemology Logic, Methodology, and Philosophy of Science, Volume 277, Kluwer Academic Publishers, 1998.

Fodor, J.A. 'The Language of Thought'. New York, Crowell, 1975.

Frege, G. 'The Foundations of Arithmentic', translated from original of 1884 by J.L.Austin, Oxford, 1934

Freund, M.A. 'A Complete and Consistent Formal System for Sortals', Studia Logica, 65, pp. 367-381, Kluwer Academic Publishers, The Netherlands, 2000.

Funkhouser, E. 'The Determinable-Determinate Relation', Nous, 2004.

Gangemi A., Fisseha F., Pettman, I., Pisanelli, D.; Taconet, M.; Keizer, J. 'A Formal Ontological Framework for Semantic Interoperability in the Fishery Domain', ECAI-02 Workshop on Ontologies and Semantic, Lyon, July 22, 2002.

Gangemi A., Guarino N., Masolo, C.; Oltramari, A. 'Sweetening Ontologies with DOLCE'. AI Magazine, 24(3), Fall 2003.

Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A 'Understanding top-level ontological distinctions', Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) , Workshop on Ontologies and Information Sharing, 2001.

Gangemi, A., Pisanelli, D.M., Steve, G. 'An Overview of the ONIONS project: Applying Ontologies to the Integration of Medical Terminologies', Data and Knowledge Engineering, vol. 31, 1999.

Gangemi, A., Prisco, A., Sagri, M.T., Steve, G., Tiscornia, D. 'Some ontological tools to support legal regulatory compliance, with a case study', Workshop on Regulatory Ontologies and the Modeling of Complaint Regulations (WORM CoRe 2003). Part of the International Federated Conferences (OTM'03). Proceedings published by Springer LNCS , November 4, 2003, Catania, Sicily, Italy.

Gangemi, A.; Catenacci, C.; Battaglia, M. 'Inflammation Ontology Design Pattern: an Exercise in Building a Core Biomedical Ontology with Descriptions and Situations', In D.M. Pisanelli (ed.) Ontolgies in Medicine, IOS Press, Amsterdam, Berlin, pp. 64-80, 2004.

Gärdenfors, P. 'Conceptual Spaces: the Geometry of Thought', MIT Press, Cambridge, USA, 2000.

Gärdenfors, P. 'How to make the Semantic Web more Semantic'. Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS), Torino, Italy, 2004.

Geert, G. & McCarthy, W. E. 'An Ontological Analysis of the Primitives of the Extended-REA Enterprise Information Architecture', The International Journal of Accounting Information Systems, Vol. 3, 2002, pp. 1-16.

Gelman, S.; Taylor, M. 'How 2-year-old children interpret proper and common names for unfamiliar objects'. Child Development 55:1535-1540, 1984.

Gemino, A. 'Empirical Comparisons of Systems Analysis Modeling Techniques', Ph.D. thesis, University of British Columbia, Canada, 1999.

Genesereth, M. R.; Nilsson, N. J. 'Logical Foundation of Artificial Intelligence', Morgan Kaufmann, Los Altos, California, 1987.

Gentner, D. 'Why Nouns are Learned Before Verbs'. In Language Development, S. Kuczav (ed.), Hillsdale, New Jersey, Erlbaum, Vol.2, 1982.

Gerstl, P. and Pribbenow, S., 1995, 'Midwinters, End Games, and Bodyparts. A Classification of Part-Whole Relations', *International Journal of Human-Computer Studies* 43: 865-889.

Girardi, R.; Serra, I. 'Using Domain Ontologies for the Specification of Domain-Specific Languages in Multi-Agent Domain Engineering', Sixth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2004), 16th International Conference on Advances in Information Systems Engineering (CAiSE), Latvia, 2004.

Gleitman, H. 'Psychology', W.W.Norton & Company, New York-London, 1991.

Goguen, J. 'An Introduction to Algebraic Semiotics, with Applications to User Interface Design, in Computation for Metaphor, Analogy and Agents', Chrystopher Nehaniv (ed.), Springer Lecture Notes in Artificial Intelligence, volume 1562, 1999.

Goguen, J.; Malcom, G. 'Software Engineering with OBJ: Algebraic Specification in Action', Kluwer, 2000.

Gómez-Pérez, A., Fernández-López, M., Corcho, O. 'Ontological Engineering', Springer Verlag, Berlin, 2004.

Gómez-Pérez, A.; Corcho, O.; Fernández-López, M. 'Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and Semantic Web', Advanced Information and Knowledge Processing, Springer-Verlag, Berlin, 2002.

Goodman, N. 'A World of Individuals', reprinted in N.Goodman, Problems and Projects, Bobbs-Merril Co., New York, 1972, 1956.

Goodman, N. 'The Structure of Appearance', Harvard University Press, Cambridge, USA, 1951.

Gottlob, G.; Schrefl, M.; Rock, B. 'Extending object-oriented systems with roles', ACM Transactions on Information Systems 14 (3), 268-296, 1996.

Green, P. and Rosemann, M. 'Ontological analysis of integrated process modelling', Information Systems, 25(2), 2000.

Grice, H.P. 'Further notes on logic and conversation', In: Syntax and semantics 9: Pragmatics (P. Cole, ed.). Academic Press, New York, pp. 113-128, 1978.

Grice, H.P. 'Logic and conversation', In: Syntax and Semantics: Vol 3, Speech Acts (P. Cole & J. Morgan, eds). Academic Press, New York, pp. 43-58, 1975.

Griffin, N. 'Relative Identity', Oxford, 1977.

Grosso, W. 'Knowledge Modeling at the Milennium (The Design and Evolution of Protégé-2000)'. Knowledge Aquisition Workshop, Banff, Canada, 1999.

Gruber, T. R. 'Toward Principles for the Design of Ontologies Used for Knowledge Sharing', International Journal of Human and Computer Studies, 43(5/6): 907-928, 1995.

Gruber, T. R., & Olsen, G. R. 'An ontology for engineering mathematics', In J. Doyle, P. Torasso, & E. Sandewall (Eds.), Fourth International Conference on Principles of Knowledge Representation and Reasoning, Gustav Stresemann Institut, Bonn, Germany, Morgan Kaufmann, 1994.

Gruninger, M., and Fox, M.S. 'The Role of Competency Questions in Enterprise Engineering', Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway, 1994a.

Gruninger, M., and Fox, M.S., 'The Design and Evaluation of Ontologies for Enterprise Engineering', Workshop on Implemented Ontologies, European Workshop on Artificial Intelligence, Amsterdam, The Netherlands, 1994b.

Gruninger, M., Atefi, K., and Fox, M.S. 'Ontologies to Support Process Integration in Enterprise Engineering', Computational and Mathematical Organization Theory, Vol. 6, No. 4, pp. 381-394, 2000.

Guarino, N. 'Formal Ontology and Information Systems', In Guarino, N. (ed.) Formal Ontology in Information Systems, Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS), Trento, Italy, June 6-8. IOS Press, Amsterdam: pp. 3-15, 1998.

Guarino, N. 'Formal Ontology, Conceptual Analysis and Knowledge Representation', International Journal of Human and Computer Studies, 43(5/6): pp. 625-640, 1995.

Guarino, N. 'The Ontological Level', In R. Casati, B. Smith and G. White (eds.), Philosophy and the Cognitive Science. Holder-Pivhler-Tempsky, Vienna: pp. 443-456, 1994.

Guarino, N. 'Understanding, Building, and Using Ontologies: A Commentary to "Using Explicit Ontologies in KBS Development"', by van Heijst, Schreiber, and Wielinga. International Journal of Human and Computer Studies, 46: pp. 293-310, 1997.

Guarino, N. and Giaretta, P. 'Ontologies and Knowledge Bases: Towards a Terminological Clarification', In N. Mars (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. IOS Press, Amsterdam: pp. 25-32., 1995.

Guarino, N.; Welty, C., 'A Formal Ontology of Properties', Proceedings of the 12th Int.Conf. on Knowledge Engineering and Knowledge Management, 2000.

Guarino, N.; Welty, C., 'An Overview of OntoClean', in S. Staab, R. Studer (eds.), Handbook on Ontologies, Springer Verlag, pp. 151-159, 2004.

Guarino, N.; Welty, C., 'Evaluating Ontological Decisions with OntoClean', Communications of the ACM, 45(2), pp.61-65, 2002a.

Guarino, N.; Welty, C., 'Identity and Subsumption', in R. Green, C. A. Bean, S. Hyon Myaeng (eds.), The Semantics of Relationships: An Interdisciplinary Perspective, Kluwer, pp.111-126, 2002b.

Guizzardi, G. 'Role Modeling in UML' (Invited presentation), Technologies for Interactive Cognitive and Communicative Agents, Final Workshop, Trento, Italy, [online: http://www.loa-cnr.it/Social_Roles.html], 2004.

Guizzardi, G. "Agent Roles, Qua Individuals and The Counting Problem", Invited Chapter in Software Engineering of Multi-Agent Systems, P. Giorgini, A.Garcia, C. Lucena, R. Chores (eds.), Springer-Verlag, 2005.

Guizzardi, G., Herre, H., Wagner G. 'On the General Ontological Foundations of Conceptual Modeling'. In Proc. of 21st Intl. Conf. on Conceptual Modeling (ER 2002), Springer-Verlag, Berlin, Lecture Notes in Computer Science, 2002a.

Guizzardi, G., Herre, H., Wagner G. 'Towards Ontological Foundations for UML Conceptual Models'. In Proc. of 1st Intl. Conf. on Ontologies, Databases and Applications of Semantics (ODBASE 2002), Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2002b.

Guizzardi, G., Wagner, G. 'Some Applications of a Unified Foundational Ontology in Business Modeling', In Ontologies and Business Systems Analysis, Michael Rosemann and Peter Green (Eds.), IDEA Publisher, 2005a.

Guizzardi, G.; Falbo, R.A.; Pereira Filho, J.G. 'From Domain Ontologies to Object Oriented Frameworks', Workshop on Ontologies (ONTO-2001), Joint German-Austrian Conference on Artificial Intelligence, Vienna, Austria, 2001a.

Guizzardi, G.; Falbo, R.A.; Pereira Filho, J.G. 'Using objects and Patterns to implement domain ontologies', Journal of Brazilian Computer Society (JBCS), Special Issue on Software Engineering (ISSN 0104-6500), Volume 8, Number 1, July 2002.

Guizzardi, G.; Falbo, R.A.; Pereira Filho, J.G. 'Using objects and Patterns to implement domain ontologies', XV Brazilian Symposium on Software Engineering (SBES-2001), Rio de Janeiro, 2001b.

Guizzardi, G.; Ferreira Pires, L.; van Sinderen, M. 'An Ontology-Based Approach for Evaluating the Domain Appropriateness and Comprehensibility Appropriateness of Modeling Languages', ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, Montego Bay, Jamaica, 2005a, Lecture Notes in Computer Science LNCS (forthcoming), Springer-Verlag.

Guizzardi, G.; Ferreira Pires, L.; van Sinderen, M. 'On the role of Domain Ontologies in the Design of Domain-Specific Visual Languages', 2nd Workshop on Domain-Specific Visual Languages, 17th ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2002), Seattle, Washington, USA, 2002. Helsinki School of Economics Printing (ISSN 1235-5674).

Guizzardi, G.; Ferreira Pires, L.; van Sinderen, M. 'Ontology-Based Evaluation and Desing of Domain-Specific Visual Modeling Languages', Proceedings of the 14th International Conference on Information Systems Development, Karlstad, Sweden, 2005b.

Guizzardi, G.; Wagner G. 'Using Formal Ontologies to Define Real-World Semantics for UML Conceptual Models', 1st Workshop on Ontologies for Biology, Heidelberg, Germany, 2002.

Guizzardi, G.; Wagner, G. 'On the Ontological Foundations of Agent Concepts', Sixth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2004), 16th International Conference on Advances in Information Systems Engineering (CAiSE), Latvia, 2004.

Guizzardi, G.; Wagner, G. 'Towards Ontological Foundations for Agent Modeling Concepts using UFO', Agent-Oriented Information Systems (AOIS), selected revised papers of the Sixth International Bi-Conference Workshop on Agent-Oriented Information Systems. Lecture Notes on Artificial Intelligence (LNAI) 3508, Springer-Verlag, 2005b, ISBN 3-540-25911-2.

Guizzardi, G.; Wagner, G.; Guarino, N.; van Sinderen, M. 'An Ontologically Well-Founded Profile for UML Conceptual Models', 16th International Conference on Advances in Information Systems Engineering (CAiSE), Latvia, 2004. Springer-Verlag, Berlin, Lecture Notes in Computer Science.

Guizzardi, G.; Wagner, G.; Herre, H. 'On the Foundations of UML as an Ontology Representation Language', 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW), Northamptonshire, UK, 2004. Springer-Verlag, Berlin, Lecture Notes in Computer Science 3257 Springer 2004, ISBN 3-540-23340-7.

Guizzardi, G.; Wagner, G.; van Sinderen, M. 'A Formal Theory of Conceptual Modeling Universals', Procedings of the Workshop on Philosophy and Informatics (WSPI), Cologne, Germany, 2004.

Guizzardi, R.S.S., Aroyo, L., Wagner, G. 'Help&Learn: A Peer-to-Peer Architecture to Support Knowledge Management in Collaborative Learning Communities', Proceedings of XIV Brazilian Symposium on Computers in Education, Rio de Janeiro, Nov. 2003.

Gupta, A. 'The Logic of Common Nouns: an investigation in quantified modal logic', Yale University Press, New Haven, 1980.

Gurr, C.A. 'Effective Diagrammatic Communication: Syntatic, Semantic and Pragmatic Issues', Journal of Visual Languages and Computing, 10, 317-342, 1999.

Gurr, C.A. 'On the isomorphism, or lack of it, of representations', In: Theory of Visual Languages, Chap. 10 (K. Marriot & B. Meyer, eds), Springer, Berlin, 1998.

Hahn, J.; Kim, J. 'Why are some Diagrams Easier to Work With?: Effects of Diagrammatic Representation on the Cognitive Integration Process of Systems Analysis and Design', In: ACM Transactions of Computer-Human Interaction, 6/3, 1999.

Hall, G. D. 'Continuity and the Persistence of Objects: When the Whole Is Greater Than the Sum of the Parts', Cognitive Psychology 37, 28–59, 1998.

Halpin, T. ' UML Data Models from an ORM Perspective: Part One', Journal of Conceptual Modeling, Issue 1, April, 1998.

Harel, D.; Rumpe, B. 'Modeling Languages: Syntax, Semantics and All That Stuff', technical paper number MCS00-16, The Weizmann Institute of Science, Rehovot, Israel, 2000.

Hayes P. 'The Naïve Physics Manifesto', In D. Ritchie (Ed.) Expert Systems in Microeletronics age. Edinburgh University Press, pp 242-270, 1978.

Hayes, P. 'Naïve Physics I: Ontology for Liguids', in Hobbs & Moore, pp.71-107, 1985.

Heller, B., Herre, H. 'Ontological Categories in GOL', Axiomathes 14: 71-90 Kluwer Academic Publishers, 2004.

Heller, B.; Herre, H.; Burek, P.; Loebe, F.; Michalek, H. 'General Ontological Language', Technical Report no. 7/2004, Institute for Medical Informatics, Statistics and Epidemiology (IMISE), University of Leipzig, Germany, 2004.

Henderson-Sellers, B., Barbier F. 'What Is This Thing Called Aggregation?', proceedings of Technology of Object-Oriented Languages and Systems Europe'99, Nancy, France, IEEE Computer Society Press, pp. 236-250, June 7-10, 1999 (ISBN: 0-7695-0275-X).

Hirsch, E. 'The Concept of Identity', Oxford University Press, New York, Oxford, 1982.

Hirschheim, R., Klein, H., and Lyytinen, K. 'Information Systems Development and Data Modeling – Conceptual and Philosophical Foundations', Cambridge University Press, Cambridge, UK, 289 pages, 1995.

Hitchman, S. 'An interpretive study of how practitioners use entity-relationship modeling in a ternary relationship situation', Communications of the Association for Information Systems , 2003, 11, 451-485

Hobbs, J. R. 'A DAML ontology of time', [online: http://www.cs.rochester.edu/_ferguson/daml/daml-time-20020830.txt], 2002.

Hornsby, K. Egenhofer, M. J. 'Identity-Based Change Operations for Composite Objects', Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, Canada, 1998.

Horrocks, I., Patel-Schneider, P., van Harmelen, F. 'From SHIQ and RDF to OWL: The making of a web ontology language', Journal of Web Semantics, 1(1):7-26, 2003.

Howells, M. 'Standards for Technology in Genealogy', Ancestry Magazine, Vol.16/2, 1998.

Husserl, E. 'Logical Investigations', Eng. trans. by J.N. Findlay, London: Routledge and Kegan Paul, 1970 (from original *Logische Untersuchungen. Zweiter Band. Untersuchungen zur Phänomenologie und Theorie der Erkenntnis* 1900/1901) .

Jackendoff, R 'The Architecture of the language faculty', Cambridge, Mass. MIT Press, 1997.

Jackendoff, R. 'Semantic Structures', Cambridge, Mass., MIT Press, 1990.

Jackendoff, R. 'Semantics and Cognition', Cambridge, Mass., MIT Press, 1983.

Jacobson, I., Booch, G., Rumbaugh, J., 'The Unified Software Development Process', Addison-Wesley, 1998.

Jardine, D.A. 'The ANSI/SPARC DBMS model', proceedings of the second SHARE Working Conference on Data Base Management Systems, Montreal, Canada, April 26-30, 1976.

Johansson, I. 'On the Transitivity of the Parthood Relations', in H. Hochberg and K. Mulligan (eds.) Relations and Predicates, ontos verlag, Frankfurt 2004, pp.161-181, 2004.

Johnson, W.E. 'Logic, Vol. 1', Cambridge University Press, 1921.

Jungclaus, R.; Saake, G.; Hartmann, T.; Sernadas, C. 'Object-Oriented Specification of Information Systems: The TROLL Language', Informatik Berichte 91-04 TU Braunschweig, Braunschweig, 1991.

Kagal, L; Finin, T.; Joshi, A. 'A Policy Based Approach to Security for the Semantic Web'. In 2nd International Semantic Web Conference (ISWC2003), September 2003.

Kaneiwa, K.; Mizoguchi, R. 'Ontological Knowledge Base Reasoning with Sort-Hierarchy and Rigidity', Proc. of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004), Canada, 2004.

Katz, N.; Baker, E.; Mcnamara, J. 'What's in a name? A study of how children learn common nouns and proper names'. Child Development 45:469-473, 1974.

Keil, F. 'Semantic and Conceptual Development: An Ontological Perspective', Cambridge, MA: Harvard University Press, 1979.

Keil, F.C. 'Concept, Kinds and Cognitive Development', MIT Press, 1992.

Kelly, S., Tolvanen, J.-P. 'Visual domain-specific modeling: Benefits and experiences of using metaCASE tools', International workshop on Model Engineering, European Conference on Object-Oriented Programming (ECOOP) ed. J. Bezivin, J. Ernst., 2000.

Klien, E., Einspanier, U., Lutz, M. & Hübner, S. 'An Architecture for Ontology-Based Discovery and Retrieval of Geographic Information', 7th Conference on Geographic Information Science (AGILE 2004).

Kogut, P, Cranefield, S., Hart, L., Dutra, M., Baclawski, K., Kokar, M, Smith, J. 'UML for Ontology Development', Knowledge Engineering Review Journal Special Issue on Ontologies in Agent Systems, Vol. 17, 2002.

Koivunen, M.R.; Miller, E., 'W3C Semantic Web Activity', proceedings of the Semantic Web Kick-off Seminar in Finland Nov 2, 2001.

Kreuse, D. 'On a Quasi-Set Theory', Notre Dame Journal on Formal Logic, 33, 402-11, 1992.

Kripke, S. 'Naming and Necessity', Harvard University Press, 1982.

Kristensen, B.B. 'Object-oriented modeling with roles', in: J. Murphy, B. Stone (Eds.), OOIS '95: Proceedings of the International Conference on Object-Oriented Information Systems, Dublin, pp. 57-71, 1995.

Krogstie, J. 'Evaluating UML: A Practical Application of a Framework for the Understanding of Quality in Requirements Specifications and Conceptual Modeling', Norwegian Informatics Conference (NIK), 2000.

Lambrix, P., 'Part-Whole Reasoning in an Object-Centered Framework', Lecture Notes in Artificial Intelligence 1771, Springer Verlag, 2000.

Landman, F. 'Plurality'. In S. Lappin (Ed.), Handbook of Contemporary Semantics (425-457). Oxford: Blackwell, 1996.

Leibniz, G.W. von 'New Essays on Human Understanding', translated by P.Remnant & J.Bennet, Cambridge University Press, Cambridge, 1981 (original from 1705).

Lenat, D.B.; Guha, R.V. 'Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project', Addison-Wesley, 1990.

Leonard, H. S. and Goodman, N., 1940, 'The Calculus of Individuals and Its Uses', *Journal of Symbolic Logic* 5: 45-55.

Lesniewski, S., 'Foundations of the General Theory of Sets. I', Eng. trans. by D. I. Barnett:, in S. Lesniewski, *Collected Works*, ed. S. J. Surma, J. Srzednicki, D. I. Barnett, and F. V. Rickey, Dordrecht: Kluwer, 1992, Vol. 1, pp. 129-173., original from 1916.

Lewis, D. K. 'On the Plurality of Worlds', Oxford, Blackwell, 1986.

Lidz, J.; Waxman, S.; Freedman, J. 'What Infants know about syntax but couldn't have learned: experimental evidence for syntactic structure at 18 months', Cognition, 89, 2003.

Link, G. 'Algebraic Semantics in Language and Philosophy'. Stanford, CA: CSLI Lecture Notes, 1997.

Little, E. 'A Proposed Methodology for The Development of Application-Based Formal Ontologies', in Pierre Grenon Ed., Reference Ontologies and Application Ontologies, Institute for Formal Ontology and Medical Information Science (IFOMIS) reports (ISSN 1611-4019), 2003.

Loebe, F. 'An Analysis of Roles', Master thesis in Computer Science, University of Leipzig, 2003.

Loux, M. J. 'The Problem of Universals', in Metaphysics: contemporary readings, M. J. Loux, (ed.), Routledge, 2001.

Lowe, E. J. 'Kinds of Being: A Study of Individuation, Identity and the Logic of Sortal Terms', Oxford: Blackwell, 1989.

Lowe, E.J. 'The possibility of Metaphysics: Substance, Identity and Time', Oxford University Press, 2001.

Majithia, S.; Walker, D.W; Gray, W.A. 'Automated Web Service Composition using Semantic Web Technologies', Proceedings of the International Conference on Autonomic Computing (ICAC'04), New York, USA, 2004.

Markman, E. 'Categorization and naming in children'. MIT Press, 1989.

Markman, E. M.; Hutchinson, J. E. 'Children's sensitivity to constraints on word meaning: taxonomic vs. thematic relations'. Cognitive Psychology, 16, p.1-27, 1984.

Marks, J.; Reiter, E. 'Avoiding unwanted conversational implicature in text and graphics', In: Proceedings of AAAI-90, pp. 450-456. 1990.

Martin, C. B. 'Substance Substantiated', Australasian Journal of Philosophy, 58, 1980.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A. 'Ontology Library', WonderWeb Deliverable D18, 2003a.

Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., Guarino, N. 'Social Roles and their Descriptions', in D. Dubois, C. Welty, M.A. Williams (eds.), Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004), Whistler, Canada, June 2-5, 2004, pp. 267-277, 2004.

Masolo, C.; Guizzardi, G.; Vieu, L.; Botazzi, E.; Ferrario, R. 'Relational Roles and Qua Individuals', AAAI Fall Symposium on Roles, an Interdisciplinary Perspective, Virginia, USA, 2005.

Masolo, C; Oltramari, A; Gangemi, A.; Guarino, N; Vieu, L. 'The Perspective of Applied Ontology' (in Italian: La Prospettiva dell'Ontologia Applicata). Rivista di Estetica (22), 2003b.

Maturana, H.R.; Varela, F.J. 'The Tree of Knowledge – The Biological Roots of Human Understanding'. Shambhala, Boston, 1987.

Mayfield, J; Finin, T. 'Information retrieval on the Semantic Web: Integrating inference and retrieval', SIGIR Workshop on the Semantic Web, Toronto, 2003.

McGuinness, D.L; Fikes, R.; Hendler, J.; Stein, L.A. 'DAML+OIL: An Ontology Language for the Semantic Web', In IEEE Intelligent Systems, Vol. 17, No. 5, pages 72-80, 2002a.

McGuinness, D.L;Fikes, R.; Stein, L.A.; Hendler, J. 'DAML-ONT: An Ontology Language for the Semantic Web', In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, 2002b.

McIlraith, S.; Son, T.C.; Zeng, H. 'Semantic Web Services', IEEE Intelligent Systems. Special Issue on the Semantic Web. 16(2):46-53, March/April, 2001.

Mcnamara, J, 'The Cognitive Basis of Language Learning in Children', Psychological Review 79:1-13, 1972.

Mcnamara, J, 'Names for Things: A Study of Human Learning'. Cambridge, MIT Press, 1982.

McNamara, J. 'A Border Dispute, the Place of Logic in Psychology'. Cambridge, M.I.T. Press, 1986.

McNamara, J. 'Logic and Cognition'. In McNamara, J.; Reyes, G. (eds.), The Logical Foundations of Cognition, Vancouver Studies in Cognitive Science, Vol. 4 (1994)

Mealy, G. H. 'Another Look at Data', Proceedings of the Fall Joint Computer Conference, November 14–16, Anaheim, California (AFIPS Conference Proceedings, Volume 31), Washington, DC: Thompson Books, London: Academic Press, 525–534, 1967.

Merriam-Webster Dictionary, [online:www.m-w.com], captured in 2004.

Mervis, C. B.; Crisafi, M. A. 'Order of Acquisition of subordinate-, basic-, and superordinate-level categories'. Child Development, 53, p. 258-266, 1982.

Mian, P. G., Falbo, R.A., 'Building Ontologies in a Domain Oriented Software Engineering Environment', IX Argentine Conference on Computer Science, La Plata, Argentina, Outubro 2003.

Mian, P. G., Falbo, R.A., 'Supporting Ontology Development with ODEd', Proceedings of the 2nd Ibero-American Symposium on Software Engineering and Knowledge Engineering, Salvador, Brazil, 2002.

Mian, P. G., Natali, A. C. C., Carvalho, A. L., Falbo, R.A. 'Domain Orientation in ODE' (in Portuguese: Orientação a domínio em ODE). Proceedings of the XXVIII Latin-American Conference on Informatics. CLEI'2002, Montevideo, Uruguay, 2002.

Milikan, R.G. 'A common structure for concepts of individuals, stuffs, and real kinds: More Mama, more milk, and more mouse', Behavioral and Brain Sciences, 21, 55–100, 1998.

Milton, S.K., Kamierczak, E. 'An Ontology of Data Modeling Languages: A Study Using a Common-Sense Realistic Ontology', Journal of Database Management, Special Issue on Ontological Analysis, Evaluation, and Engineering Of Business Systems Analysis Methods, 18 pages, 2004.

Montague, R. 'Formal philosophy : selected papers of Richard Montague', Yale University Press, 1974.

Moore, G. E. 1919-20. 'External and internal relations'. Proceedings of the Aristotelian Society 20, pp. 40-62.

Moore, M.K.; Borton, R.; Darby, B.L. 'Visual tracking in young infants: evidence for object identity or object permanence?', Journal of Experimental Child Psychology, 25, 183-198, 1978.

Morris, C.W. 'Foundations of a theory of signs', In: International Encyclopedia of Unified Science (O. Neurath, R. Carnap & C. Morris, eds), Chicago University Press, Chicago, pp. 77-138, 1938.

Mulligan, K., Simons, P., Smith, B., 'Truth-Makers', Philosophy and Phenomenological Research, 44, 287–321, 1984.

Mulligan, K.; Smith, B. 'A Relational Theory of the Act', Topoi (5/2), 115-30,1986.

Mylopoulos, J. 'Conceptual modeling and Telos', In P. Loucopoulos and R. Zicari, editors, Conceptual modeling, databases, and CASE, chapter 2, pages 49--68. Wiley, 1992.

Mylopoulos, J. 'Information Modeling in the Time of the Revolution', Information Systems 23(3-4), June, 1998.

Natali, A. C. C., Falbo, R.A., 'Knowledge Management in Software Engineering Environments', Proceedings of the XVI Brazilian Symposium on Software Engineering (SBES'2002), pp.238-253, Gramado, Brazil, October 2002.

Neighbors J. 'Software Construction Using Components', Phd Thesis, University of Califórnia, Irvine, EUA, 1981.

Neuhaus, F.; Grenon, P.; Smith, B. 'A Formal Theory of Substances, Qualities, and Universals'. Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS), Torino, Italy, 2004.

Object Management Group 'Ontology Definition Metamodel Request for Proposals', OMG Document ad/2003-03-40, 2003a.

Object Management Group, 'UML 2.0 Infrastructure Specification', Doc.# ptc/03-09-15, Sep. 2003b.

Object Management Group, 'UML 2.0 Superstructure Specification', Doc.# ptc/03-08-02, 2003c.

Object Management Group, 'Unified Modeling Language Specification (version 1.4)', 2001.

Odell, J.; Bock, C. 'A Foundation for Composition', in Advanced Object-Oriented Analysis and Design Using UML by James J. Odell, Cambridge University Press, 1998.

Odell, J.; Nodine, M.; Levy, R. ' A Metamodel for Agents, Roles, and Groups', Agent-Oriented Software Engineering (AOSE) IV, James Odell, P. Giorgini, Jörg Müller, eds., Lecture Notes on Computer Science volume (forthcoming), Springer, Berlin, 2005.

Odell, J.; Parunak, H.V.D; Fleischer, M. 'The Role of Roles in Designing Effective Agent Organizations,', Software Engineering for Large-Scale Multi-Agent Systems, Garcia, A.; Lucena, C.; Zambonelli, F.; Omicini, A.; Castro, J. (eds.), Lecture Notes on Computer Science volume 2603, Springer, Berlin, 2003, pp 27-28.

Odell, J.J. 'Six Different Kinds of Composition', in Advanced Object-Oriented Analysis and Design Using UML, Cambridge University Press, 1998.

Ogden, C. K., Richards, I. A. 'The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism', New York: Harcourt Brace Jovanovich, 1923.

Opdahl A., Henderson-Sellers, B., Barbier F. 'Ontological Analysis of whole-part relationships in OO-models', Information and Software Technology 43, pp. 387-399, 2001.

Opdahl, A. L., Henderson-Sellers, B. 'Grounding the OML metamodel in ontology', The Journal of Systems and Software, 57(2), 119-143, 2001.

Opdahl, A., Henderson-Sellers, B., Barbier, F. 'An ontological evaluation of the OML metamodel', In E. Falkenberg and K. Lyytinen, editors, Information System Concepts: An Integrated Discipline Emerging. IFIP/Kluwer, 1999.

Pan, F.; Hobbs, J. R. 'Time in OWL-S'. In Proceedings of AAAI-04 Spring Symposium on Semantic Web Services, Stanford University, California, 2004.

Pan, J.; Horrocks, I. 'RDFS(FA) and RDF MT: Two semantics for RDFS', In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, Proc. of the 2003 International Semantic Web Conference (ISWC 2003), number 2870 in Lecture Notes in Computer Science, pages 30-46. Springer, 2003.

Paolucci, M. ; Sycara, K.; Kawamura, T. 'Delivering Semantic Web Services', Techinical Report CMU-RI-TR-02-32, Robotics Institute, Carnegie Mellon University, May, 2003.

Parsons, J. and Wand, Y. 'The object paradigm - two for the price of one?', In Proceedings of the Workshop on Information Technology and Systems WITS 1991, pages 308-319, 1991.

Parsons, J.; Wand, Y. 'Emancipating instances from the tyranny of classes in information modeling', ACM Transactions on Database Systems (TODS), Volume 25 Issue 2, 2000.

Parsons, T. 'Events in the Semantics of English: A Study in Subatomic Semantics'. Cambridge/MA: MIT Press, 1990.

Partridge, C. 'The Role of Ontology in Integrating Semantically Heterogeneous Databases', LADSEB-CNR Technical Report 05/2002.

Peirce, C. S. 'Collected Papers of Charles Sanders Peirce', Volume II. C. Hartshorne and P. Weiss (eds.), Cambridge, Mass., Harvard University Press, 1960.

Peirce, C. S. 'Scientific Metaphysics', Vol. VI of the Collected Papers, Cambridge, Mass.: Harvard University Press, 1935 (original from 1892-1893).

Peitgen, H-O; Jurgens, H.; Saupe, D. 'Chaos and Fractals: New Frontiers of Science', Springer-Verlag, New York, 1992.

Penrose, R. 'The Large, The Small and The Human Mind', Cambridge University Press, 2000.

Perry, J. 'The same F'. Philosophical Review (1970)

Petre, M.; Green, T.R.G. 'Requirements of graphical notations for professional users: electronics CAD systems as a case study', Le Travail Humain 55, 47-70, 1992.

Pinker, S. & Bloom, P. 'Natural language and natural selection'. Behavioral and Brain Sciences, 1990, 13:4, 707-784.

Pisanelli D., Gangemi A., Steve G. 'An Ontological Analysis of the UMLS Methatesaurus', Proceedings of AMIA 98 Conference, 1998.

Pisanelli, D.M., Zaccagnini, D., Capurso, L., Koch, M. 'An ontological approach to evidence-based medicine and meta-analysis', Proceedings of Medical Informatics Europe (MIE), St. Malo, France, May 2003.

Poli, R. 'Qua-theories'. In Shapes of Forms. Kluwer, Dordrecht, 1998.

Pribbenow, S. 'Meronymic Relationships: From Classical Mereology to Complex Part-Whole Relations', The Semantics of Relationships, R.Green and C. A. Bean (ed.), Kluwer Academic Publishers, 2002.

Price, H.H. 'Thinking and Experience', Hutchinson, 1953.

Putnam H.: 'Logic and Psychology'. In McNamara, J.; Reyes, G. (eds.), The Logical Foundations of Cognition,Vancouver Studies in Cognitive Science, Vol 4 (1994).

Quine, W. V. O. 'On What There Is', as reprinted in From a Logical Point of View, New York: Harper & Row, 1953.

Quine, W. V. O. 'Ontological relativity', and Other Essays, New York: Columbia University Press, 1969.

Quine, W.V.O. 'Word and Object', MIT Press, Cambridge, 1960.

Quinton, A. 'Properties and Classes', Proceedings of the Aristotelian Society, vol. 58, 1957.

Raubal, M. 'Formalizing Conceptual Spaces', Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS), Torino, Italy, 2004.

Rector A., Solomon W., Nowlan W., Rush T. 'A Terminology Server for Medical Language and Medical Information Systems', Methods of Information in Medicine, Vol. 34, 147-157, 1995.

Richardson, J.; Schwartz, P. 'Aspects: Extending objects to support multiple, independent roles', in: J. Cli.ord, R. King (Eds.), Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data, SIGMOD Record ACM Press, vol. 20, no. 2, pp. 298-307, 1991.

Rios, D. 'Using Ontologies in Context-Aware Service Platforms', Master Thesis in Telematics, University of Twente, The Netherlands, 2003.

Rios, D.; Costa, P.D.; Guizzardi, G.; Ferreira Pires, L.; Pereira Filho, J.G., van Sinderen, M.; 'Using Ontologies for Modeling Context-Aware Service Platforms', OOPSLA 2003 Workshop on Ontologies to Complement Software Architectures, Anaheim, CA, USA, October 26-30, 2003.

Rosch, E.; Mervis, C. B.; Gray, W. D.; Johnson, D.M.; Boyes-Braem, P. 'Basic objects in natural categories', Cognitive Psychology, Volume 8, Issue 3, July 1976, p. 382-439, 1976.

Rowe, D., Leaney, J.R., 'Evaluating evolvability of computer based systems architectures - an ontological approach', Proceedings of the 10th International Conference on the Engineering of Computer Based Systems (ECBS'97), Monterey, California, 1997.

Rowe, D., Leaney, J.R., Lowe, D.B, 'Defining systems evolvability - a taxonomy of change', Proceedings of the 11th IEEE Conference and Workshop n Engineering of Computer-Based Systems (ECBS'98), Jerusalem, Israel, 1998.

Russel, B. 'An Enquire into Meaning and Truth', Allen and Unwin, 1940.

Russel, B. 'Human Knowledge, its Scopes and Limits', Allen and Unwin, 1948.

Russel, B. 'On Denoting', Mind, 14:479-93, 1905.

Russel, B., 'The Philosophy of Logical Atomism', 1918.

Sagri, M.T 'Project for the Development of an Online Lexical Juridical Network through the specification of ItalWordnet' (in italian: Progetto per lo svulippo di una rete lessicale giuridica on line attraverso la specializzazione di ItalWordnet). Informatica e Diritto, ESI, Napoli, 2003.

Sagri, M.T; Tiscornia, D; Gangemi, A. 'An Ontology-Based Model for Representing "Bundle-of-Rights', Submitted to the Workshop on Regulatory Ontologies and the Modeling of Complaint Regulations (WORM CoRe 2004). Part of the International Federated Conferences (OTM'04), Larnaca, Cyprus, 2004.

Saksena, M.; France, R. B.; Larrondo-Petrie, M. 'A Characterization of Aggregation', in Proceedings of the 5th International Conference on Object-Oriented Information Systems (OOIS'98), September 1998.

Saksena, M.; Larrondo-Petrie, M.; France, R. B.; Evett, M. 'Extending Aggregation Constructs in UML', UML'98: Beyond the Notation - International Workshop, 1998.

Santos, L.O.B.S. 'Semantic Services Support for Context-Aware Platforms', Master Dissertation, Department of Computer Science, Federal University of Espirito Santo, Brazil, 2004.

Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., Lee, J. 'The Process Specification Language (PSL): Overview and Version 1.0 Specification', NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD, 2000.

Schneider, L. 'Formalised Elementary Formal Ontology', ISIB-CNR Technical Report 03/2002, [online: http://www.loa-cnr.it/Publications.html], 2002.

Schneider, L. 'Foundational Ontologies and the Realist Bias', in Pierre Grenon Ed., Reference Ontologies and Application Ontologies, Institute for Formal Ontology and Medical Information Science (IFOMIS) reports (ISSN 1611-4019), 2003a.

Schneider, L., 'Designing Foundational Ontologies: The Object-Centered High-Level Reference Ontology OCHRE as a Case Study', 22th International Conference on Conceptual Modeling (ER), USA, 2003b.

Schunn, C. D., & Vera, A. H. 'Causality and the categorization of objects and events'. Thinking & Reasoning, 1(3), 237-284, 1995.

Schwartz, S. 'Natural Kinds and Nominal Kinds', Mind 89, pp. 182-195, 1980.

Shanks, G.; Tansley, E.; Weber, R. 'Using ontology to validate conceptual models', Special Section on Conceptual modeling and system architecting, Communications of the ACM, Volume 46 Issue 10, Oct. 2003.

Shepard, R. N. 'The Analysis of Proximities: multidimensional scaling with an unknown distance function'. I, Psychometrika 27: 125-140, 1962a.

Shepard, R. N. 'The Analysis of Proximities: multidimensional scaling with an unknown distance function'. II, Psychometrika 27: 219-246, 1962b.

Sheth, A.P., Kashyap, V. 'So Far (Schematically) yet So Near (Semantically)', Proceedings of the IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5), Victoria, Australia, 1992.

Shimojima, A. 'Operational constraints in diagrammatic reasoning', In: Logical Reasoning with Diagrams. Oxford University Press, New York, pp. 27-48, 1996.

Simons, P. ' Particular in Particular Clothing: Three Trope theories of Substance', Philosophy and Phenomenological Research, 54, 553-576. Reprinted in: S. Laurence and C. Macdonald (eds.), Contemporary Readings in Foundations of Metaphysics. Oxford: Blackwell, 1998, 364-384, 1994.

Simons, P. 'Relational Tropes', Haefliger, G., Simons, P. (eds.). Analytic Phenomenology: Essays in Honor of Guido Küng. Dordrecht: Kluwer, 1995.

Simons, P.M. 'Parts. An Essay in Ontology', Oxford: Clarendon Press, 1987.

Smith, B, 'On Substances, Accidents and Universals'. In Defence of a Constituent Ontology". Philosophical Papers 27, pp. 105-127, 1997.

Smith, B. 'Formal Ontology, Commonsense and Cognitive Science', International Journal of Human and Computer Studies, 43(5/6): pp. 625-640, 1995.

Smith, B. 'Logic and Formal Ontology', in J. N. Mohanty and W. McKenna, eds., Husserl's Phenomenology: A Textbook, Lanham: University Press of America, 29-67, 1989.

Smith, B. 'Ontology: Philosophical and Computational', [online: http://ontology.buffalo.edu/ontology(PIC).pdf], capture in Ago/13, 2004.

Smith, B. 'The Basic Tools of Formal Ontology', from Nicola Guarino (ed.), Formal Ontology in Information Systems Amsterdam, Oxford, Tokyo, Washington, DC: IOS Press (Frontiers in Artificial Intelligence and Applications), 19–28, 1998.

Smith, B., 'Fiat Objects', 11th European Conference on Artificial Intelligence, The Netherlands, 1994.

Smith, B.; Brogaard, B. 'Quantum Mereotopology', Annals of Mathematics and Artificial Intelligience , 35/1-2 (2002), 153-175, 2002.

Smith, B.; Mulligan, K. 'Framework for Formal Ontology', Topoi 2: 73-85, 1983.

Smith, B.; Welty, C. 'Ontology: Towards a new synthesis', In Chris Welty and Barry Smith, eds., Formal Ontology in Information Systems. Pp. iii-x. Ongunquit, Maine: ACM Press, 2001.

Smith, E. E.; Medin, D. L. 'Categories and concepts'. Harvard University Press, 1981.

Snoeck, M.; Dedede, G. 'Existential Dependency: The Key to semantic integrity between structural and behavioral aspects of object types', IEEE Transactions on Software Engineering, Vol.24, No.4, Apr.1998.

Snoeck, M.; Dedene, G. 'Core Modelling Concepts to define Aggregation', L'Objet , vol. 7, no. 3, pp.281-306. Extended version published as CTEO Research Report OR 0034 Characterising Aggregations with Existence Dependency 38 p. by Monique Snoeck and Guido Dedene, 2001.

Soffer P., Golany, B.; Dori, D.; Wand, Y. 'Modelling Off-the-Shelf Information Systems Requirements: An Ontological Approach', Requirements Engineering 6(3): 183-199, 2001.

Sowa, J. 'Knowledge Representation: Logical, Philosophical and Computational Foundations', Brooks/Cole, 2000.

Sowa, J.F. 'Conceptual Structures: Information Processing in Mind and Machine', Addison-Wesley, New York, 1984.

Sowa, J.F. 'Using a lexicon of canonical graphs in a semantic interpreter', in: M.W. Evens (Ed.), Relational Models of the Lexicon:Representing Knowledge in Semantic Networks, Cambridge University Press, Cambridge; 1988, pp. 113-137, 1988.

Spelke, E. S., Kestenbaum, R., Simons, D. J., & Wein, D. 'Spatio-temporal continuity, smoothness of motion and object identity in infancy'. British Journal of Developmental Psychology, 13, 113–142, 1995

Spelke, E.S. 'Principles of object perception'. Cognitive Science, 14, 29-56, 1990.

Spivey, J. M. 'Understanding Z', Cambridge University Press, 1988.

Stamper, R.; Liu, K; Hafkamp, M.; Ades, Y. 'Understanding the roles of signs and norms in organizations - a semiotic approach to information systems design', Behaviour & Information Technology, 2000, Vol. 19, No. 1, 15-27.

Steimann, F. 'A radical revision of UML's role concept'. In: A Evans, S Kent, B Selic (eds.), UML 2000 Proc. of the 3rd Intl. Conf., Springer (2000a) 194–209

Steimann, F. 'Role = Interface: a merger of concepts'. Journal of Object-Oriented Programming 14:4 (2001) 23–32

Steimann, F.: 'On the representation of roles in object-oriented and conceptual modeling'. Data & Knowledge Engineering 35:1 (2000b) 83–106

Stout, G.F. 'The Nature of Universals and Propositions', Oxford University Press (British Academy Lecture, reprinted in G. F. Stout, Studies in Philosophy and Psychology, Mcmillan, 1930), 1921.

Strang, T.; Linnhoff-Popien, C.; Korbinian. F. 'CoOL: A Context Ontology Language to enable Contextual Interoperability', Lecture Notes in Computer Science 2893, pp. 236-247, 2003.

Strawson, P. F. 'Individuals. An Essay in Descriptive Metaphysics', London and New York: Routledge , 1959.

Szabó, Z.G. 'On qualification'. Philosophical Perspectives, 17(1):385, February 2003.

Thomasson, A. L. 'Methods of Categorization'. Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS), Torino, Italy, 2004.

Thompson, E. 'Color Vision, Evolution and Perceptual Content', Synthese 104: 1-32, 1995.

Tolvanen, J.-P., Gray, J., Rossi, M., (editors) 'Domain-Specific Modeling with Visual Languages, Special Issue of the Journal of Visual Languages and Computing, Elsevier Science, 2004.

Tversky, B. 'Parts, Partonomies and Taxonomies', Developmental Psychology, Vol. 25, No 6, pp. 983-995, 1989.

Tversky, B., & Hemenway, K. 'Objects, parts, and categories'. Journal of Experimental Psychology: General, 113, 169-193, 1984.

Ullmann, S. 'Semantics: An Introduction to the Science of Meaning', Basil Blackwell, Oxford, 1972.

Uschold, M., King, M., Moralee, S., Zorgios, Y. 'The Enterprise Ontology', The Knowledge Engineering Review, Vol. 13, Special Issue on Putting Ontologies to Use (eds. Mike Uschold and Austin Tate). Also available from AIAI as AIAI-TR-195, 1998.

van Belle, J.P. 'Moving Towards Generic Enterprise Information Models: From Pacioli to CyC'. Proceedings of the 10th Australasian Conference on Information Systems, Wellington, 1999.

van der Vet, P. E., Mars, N. J. I. 'Bottom-Up Construction of Ontologies: The Case of an Ontology of Pure Substances', IEEE Transactions IEEE Transactions on Knowledge and Data Engineering 10(4) (July/August 1998), 513-526., 1995.

van der Vet, P. E., Mars, N. J. I. 'The Plinius Ontology of Ceramic Materials', In: Workshop on the Comparison of Implemented Ontologies, Amsterdam: N.J.I. Mars, 1994.

van Leeuwen, J. 'Individuals and sortal concepts : an essay in logical descriptive metaphysics', PhD Thesis, Univ. of Amsterdam, 1991.

van Sinderen, M. 'On the Design of Application Protocols', PhD Thesis, University of Twente, The Netherlands, 1995.

Varejao, F. M., Menezes, C. S., Garcia, A. C. B., Souza, C. S., Fromherz, M. 'Towards an Ontological Framework for Knowledge-Based Design Systems', In: VI International Conference on Artificial Intelligence in Design - AID 2000, Worcester, Massachussets, EUA, 2000.

Varela, F. J. 'Principles of Biological Autonomy', Appleton & Language, 1979.

Varzi, A. 'Mereology', Stanford Encyclopedia of Philosophy, [online: http://plato.stanford.edu/entries/mereology/], 2003.

Varzi. A.C.: 'Parts, wholes, and part-whole relations: The prospects of mereotopology'. Journal of Data and Knowledge Engineering, 20:259–286, 1996.

Veres, C.; Hitchman, S, 'Using Psychology to Understand Conceptual Modeling', 10th European Conference on Information Systems (ECIS 2002), Poland.

Veres, C.; Mansson, G. 'Cognition and Modeling: Foundations for Research and Practice', JITTA: Journal of Information Technology Theory & Application. vol. 7, no. 1, 93-10, 2005.

Vermeer, M.W.W. 'Semantic interoperability for legacy databases', PhD Thesis, University of Twente, The Netherlands, 1997.

Vermeer, M.W.W., Apers, P.M.G. 'On the Applicability of Schema Integration Techniques to Database Interoperation', ER 1996, 1996: p. 179-194.

Vieu, L.; Aurnague, M. 'Part-of Relations, Functionality and Dependence', In: M. Aurnague, M. Hickmann & L. Vieu (eds.), Categorization of Spatial Entities in Language and Cognition. Amsterdam: John Benjamins, 2005.

Vigna, S. 'ERW: The Manual', [online: http://erw.dsi.unimi.it/ERW/index.html], 2004.

Vissers C. A.; Ferreira Pires L.; van Sinderen, M., Quartel D. 'The Design of Telematics Systems', Lecture Notes, University of Twente, The Netherlands, 2000.

Vissers, C. A.; van Sinderen, M.; Ferreira Pires, L. 'What Makes Industry Believe in Formal Methods', In André Danthine, Guy Leduc, and Pierre Wolper, editors, Protocol Specification, Testing, and Verification, XIII, pages 326, the Netherlands, Elsevier Science Publishers B.V. (North-Holland), 1993.

Vögele, T; Hübner, S.; Schuster, G. 'BUSTER - An Information Broker for the Semantic Web', KI - Künstliche Intelligenz 3, pp. 31-34, July, 2003.

Wagner, G. 'The Agent-Object-Relationship Metamodel: Towards a Unified View of State and Behavior'. Information Systems 28:5, 2003.

Wagner, G.; Taveter; K. 'Towards Radical Agent-Oriented Software Engineering Processes Based on AOR Modeling', Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), 2004.

Wand, Y. & Weber, R. 'Mario Bunge's Ontology as a formal foundation for information systems concepts', In Weingartner, P. & Dorn, G.J.W. (eds.), Studies on Mario Bunge's Treatise, Rodopi, Atlanta, 1990.

Wand, Y. and Weber, R. 'An ontological evaluation of systems analysis and design methods', In E. Falkenberg and P. Lingreen, editors, Information System Concepts: An In-Depth Analysis. Elsevier Science Publishers B.V., North-Holland, 1989.

Wand, Y. and Weber, R. 'On the ontological expressiveness of information systems analysis and design grammars'. Journal of Information Systems, (3), 217-237, 1993.

Wand, Y. and Weber, R. "On the Deep Structure of Information Systems." Information Systems Journal, 1995, 3, 203-223.

Wand, Y., Storey, V. C., and Weber, R. 'An ontological analysis of the relationship construct in conceptual modeling', ACM Transactions on Database Systems, 24(4), 494-528, 1999.

Wand. Y., Wang, R. 'Anchoring Data Quality Dimensions in Ontological Foundations', Communications of the ACM, Vol.39, no. 11, 1996.

Wandinger, N. 'Masses of Stuff and Identity', Erkenntnis 48: 303–307, Kluwer Academic Publishers, 1998.

Waxman, S. R., & Markow, D. R. 'Words as invitations to form categories: Evidence from 12- to 13-month-old infants'. Cognitive Psychology, 29, 257-302, 1995.

Weber, R. 'Ontological Foundations of Information Systems', Coopers & Lybrand, Melbourne, Australia, 1997.

Weisstein, E. W. MathWorld, Wolfram Research, online:[http://mathworld.wolfram.com/], (captured in 2004).

Welty, C., Guarino, N. 'Supporting Ontological Analysis of Taxonomic Relationships'. Data and Knowledge Engineering, 39(1), 51-74, 2001.

Whitehead, A.N. 'Process and Reality: An Essay in Cosmology', corrected edition, D.R.Griffin & D.W.Sherburne (eds.), Free Press, New York, 1978 (original from 1929).

Wieringa, R.J. de Jonge, W., Spruit, P.A.: 'Using dynamic classes and role classes to model object migration'. Theory and Practice of Object Systems, 1(1), 61-83, 1995.

Wiggins, D. 'Sameness and Substance Renewed'. Cambridge University Press, 2001.

Williams, D.C. 'The Elements of Being', in Principles of Empirical Realism, by D.C. Williams, Charles Thomas, 1966.

Wilson, R. A.; Keil, F. C. (ed.) 'The MIT Encyclopedia of the Cognitive Sciences', MIT Press, 1999.

Winston M.E.; Chaffin R.; Herrman D. 'A taxonomy of part-whole relations', Cognitive Science 11 (1987), pp.417-444.

Wittgenstein, L. 'Tractactus Logico-Philosophicus', 1922.

WordNet 'WordNet 2.0: A Lexical Database for The English Language', [online: http://wordnet.princeton.edu/], captured in May, 3rd, 2005.

Xu, F. 'Categories, kinds, and object individuation in infancy'. in L. Gershkoff-Stowe and D. Rakison (eds.), Building object categories in developmental time. Papers from the 32nd Carnegie Symposium on Cognition. New Jersey: Lawrence, 2004.

Xu, F., Carey, S. 'Infants' metaphysics: The case of numerical identity', Cognitive Psychology, 30(2), 111-153, 1996.

Xu, F.; Baker, A. 'Object individuation in 10-month-old infants using a simplified manual search method'. Journal of Cognition and Development, 2003.

Xu, F.; Carey, S.; Quint, N. 'The Emergence of Kind-Based Object Individuation in Infancy', Cognitive Psychology, 40, 155-190, 2004.

Zimmerman, D. W. 'Theories of Masses and Problems of Constitution', The Philosophical Review 104, 53–110.

Index

Ontological Foundations For Structural Conceptual Models

Giancarlo Guizzardi

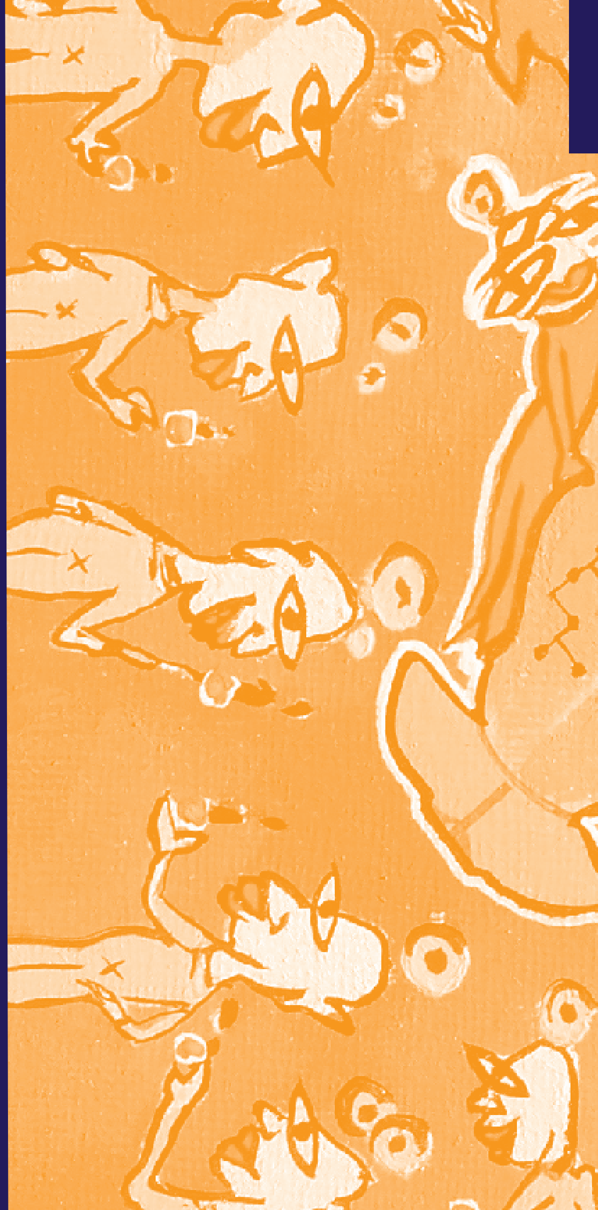
The main objective of this thesis is to contribute to the theory of Conceptual Modeling by proposing ontological foundations for structural conceptual models.

Conceptual Modeling is a discipline of great importance to several areas in Computer Science. Its main objective is concerned with identifying, analyzing and describing the essential concepts and constraints of a universe of discourse, with the help of a (diagrammatic) modeling language that is based on a set of basic modeling concepts (forming a metamodel).

In this thesis, we show how conceptual modeling languages can be evaluated and (re)designed with the purpose of improving their ontological adequacy. In simple terms, ontological adequacy is a measure of how close the models produced using a modeling language are to the situations in the reality they are supposed to represent. The thesis starts by proposing a systematic evaluation method for comparing a metamodel of the concepts underlying a language to a reference ontology of the corresponding domain in reality. The focus of this thesis is on general conceptual modeling languages (as opposed to domain specific ones). Hence, the proposed reference ontology is a foundational (or upper-level) ontology. Moreover, since, it focuses on structural modeling aspects (as opposed to dynamic ones), this foundational ontology is an ontology of objects, their properties and relations, their parts, the roles they play, and the types they instantiate.

The proposed ontology was developed by adapting and extending a number of theories coming, primarily, from formal ontology in philosophy, but also from cognitive science and linguistics. Once developed, every sub-theory of the ontology is used in the creation of methodological tools (e.g., modeling profiles, guidelines and design patterns). The expressiveness and relevance of these tools are shown throughout the thesis to solve some classical and recurrent conceptual modeling problems.

Finally, the thesis demonstrates the applicability and usefulness of both the method and the proposed ontology by analyzing and extending a fragment of the Unified Modeling Language (UML) which deals with the construction of structural conceptual models.



ISBN 90-75176- 81-3

CTIT
Centre for Telematics and Information Technology

Telematica Instituut