OBJECT-ORIENTED DESIGN OF MICROWAVE CIRCUIT SIMULATORS

P. CARVALHO - E. NGOYA - J. ROUSSET - J. OBREGON

IRCOM - URA CNRS n° 356

123, avenue Albert-Thomas 87060 LIMOGES cédex (FRANCE)

ABSTRACT :

To meet MMIC designers requirements, CAD softwares must continuously fit new specifications and offer new functions. To achieve this goal software architectures must be as flexible and extensible as possible, which is not the case at present. In this paper, an object-oriented design approach is proposed. An identification of the basic objects of circuit analysis is done and software architecture of typical circuit simulators are presented.

I - INTRODUCTION

As MMIC technologies are becoming more reliable and sophisticated electronic functions with increasing complexity can be integrated, in response to new market opening.

This requierement for more complex MMIC continuously requires new powerful CAD tools providing adapted functions. The CAD functions offered today by available microwave circuit simulators are far below the current demand. The reasons of this delay has been a rapid development of new microwave communication applications on one hand, and on the other hand CAD software architecture rigidity. In this lack of architecture flexibility lie the difficulties in evolving old function and building new ones.

Hence it takes a long time before the state of art knowledge in the field of circuit analysis is implemented. It took about a decade before efficient harmonic balance facilities were available in commercial simulators. It is therefore desirable to design simulator software in such a way to avoid recreating the wheel every time and speed up implementation of new knowledge. This is ended possible with bottom up software architecture as promoted by new object oriented languages like Eiffel [1] or C ++ [2].

Traditional software design methods are based on the actions, and system architectures are structured around functions. These architectures will need to be changed for every change in the system requierement, and are not appropriate for the long term view of continuous adaptation.

In the concept of object oriented software design, program architectures must be built around processed data as data remain ultimately the same from one function to another.

This paper proposes a reflection on basic data common to every circuit analysis and their implementation in form of "object classes", following object oriented design concept.

Once basic circuit analysis classes has been identified, implementation of any given analysis function will be greatly simplified as it results in as straightforward assembly of these classes. In section II, three fondamental families of circuit object classes will be introduced while section III presents object oriented architecture of various circuit analysis functions.

II - OBJECTS OF THE CIRCUIT ANALYSIS THEORY

Use of object-oriented design leads to software structures based on the objects (abstract data types) every system or subsystem manipulates. Object-oriented systems are built as collections of classes (object implementations) designed to be as general and reusable as possible. Classes offer services and may be connected by two types of relations: client and inheritance relations.

From object-oriented design point of view, an analysis kernel will be based on three main object families that are circuit theory objects, vectors space objects and circuit equation objects.

1 - Circuit theory objects

Network theory is based on properties of circuit elements and on how these elements are interconnected. Meaningful objects in circuit theory field may be obtained directly from common terminology of circuit theory, that is to say topology, electrical variables, Kirchhoff and element laws. These objects are completely described in the set of classes, sketched in figure 1 which shows their interrelations, structured around topmost class, that is the circuit class which models any type of electrical network.

2 - Vector space objects

As many of the problems found in circuit simulation can be ultimately reduced to solving a series of related systems of linear or nonlinear algebraic equations, the second important object family is that of objects related to vector space, i.e. vectors, matrices (full, sparse, ...), and associated operators.

CH3277-1/93/0000-1491$01.00 © 1993 IEEE

In this context, systems of equations will be assembly of these objects completed by solving technique implementations. Among these techniques, gaussian elimination and LU factorization, oriented or not towards sparse matrices, Newton-Raphson iterative technique and Fast-Fourier transform are the most common.

All these objects may be implemented in a set of classes layed out in figure 2. This set is structured around the topmost class, equation system, which offers a general purpose mathematical support.

3 - Circuit equation objets

Objects of this type are practical network formulations for computer applications throught with a circuit corresponds to a system of equations. They constitute a set of classes which acts as an interface between the two previous families of classes, as show in figure 3. Tableau, nodal and modified nodal formulations are the common circuit equation forms.

III - TYPICAL SIMULATOR ARCHITECTURES

In section II we were not at all concerned by any ultimate function to be realised. However while the three fondamental sets of circuit object classes are implemented, design of the topmost function is greatly simplified. This is readily made as an assembly of the above classes by means of client and inheritance relationships.

As an example, hereafter we show software architectures of typical circuit analysis kernels [3, 4]:

- Frequency domain analysis - Figure 4a
- Time domain analysis - Figure 4b
- Harmonic balance analysis - Figure 4c
- Nonlinear noise analysis - Figure 4d
- Stability analysis of nonlinear circuit - Figure 4e
- Circuit Optimization - Figure 4f

These analysis modules has been implemented using the Eiffel language on SUN and HP workstations. The result is very hopeful. Based on these modules and fondamental classes we are constructing a library of reusable software components for circuit simulators.

IV-CONCLUSION

The paper has introduced the notion of object oriented design in the construction of circuit analysis software throught a reflection on the basic objects and object classes of circuit analysis . Simple architectures of the three fundamental families of circuit analysis objects are given . With these fundamental families of classes, it is shown that it is as easy to build a frequency analysis of a linear circuit as to build a harmonic balance analysis of a nonlinear circuit.

Definetely any new circuit analysis technique may be rapidly and efficiently experienced without the need of reconstructing the whole data structure as with old top down functional design. Indeed the object oriented concept gives the mean to the CAD tools to easily follow the state of art knowledge in the circuit theory and quickly respond to the increasing MMIC design demand.

V-REFERENCES

[1] **B. MEYER**
Object-oriented software construction.
Prentice Hall - 1988.

[2] **B. STROUSTUP**
The C++ programming language 2nd Edition.
Addison-Wesley - 1986.

[3] **V. RIZZOLI, A. NERI**
State of the art and present trends in nonlinear microwave CAD techniques.
IEEE Transactions on MTT, Vol. 36, n° 2, February 1988, pp 343-365.

[4] **R.J. GILMORE, M.B. STEER**
Nonlinear circuit analysis using the method of harmonic-balance - A review of the art. Part. I Introductory concepts.
International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering, Vol. 1, n° 1, 1991, pp 22-37.

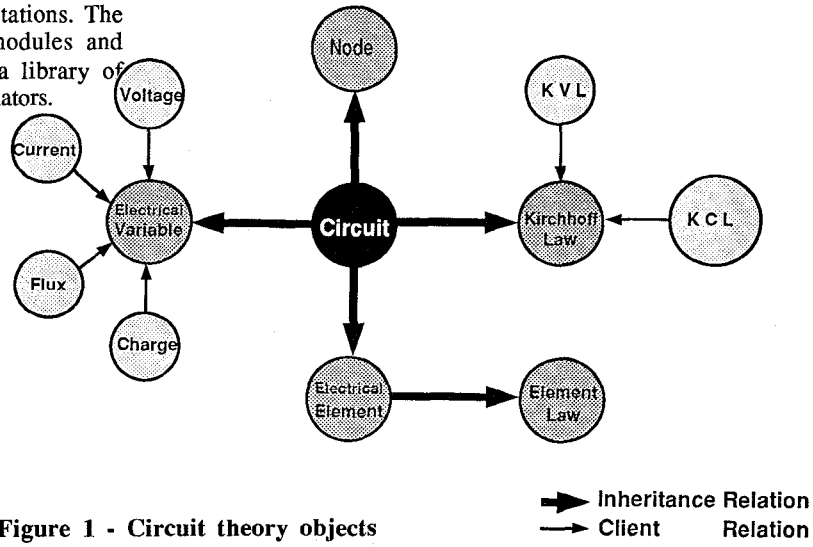

Figure 1 - Circuit theory objects

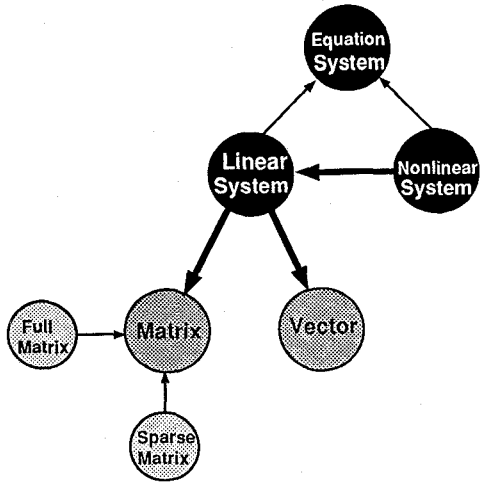

Figure 2 - Vector space objects

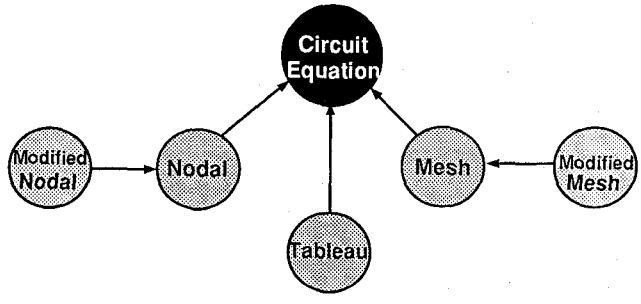

Figure 3 - Circuit equation formulations

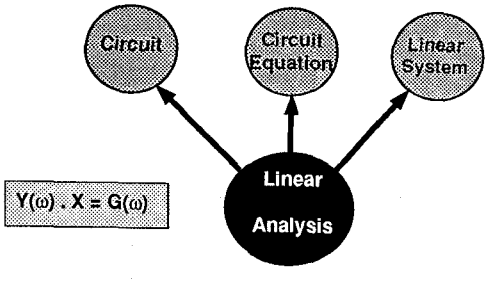

Figure 4a - Frequency domain analysis

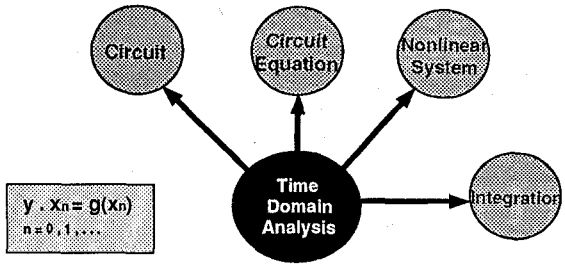

Figure 4b - Time domain analysis

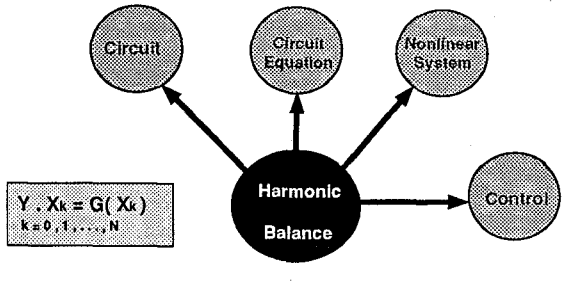

Figure 4c - Harmonic balance analysis

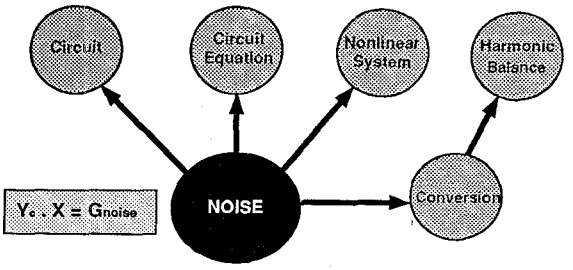

Figure 4d - Nonlinear noise analysis

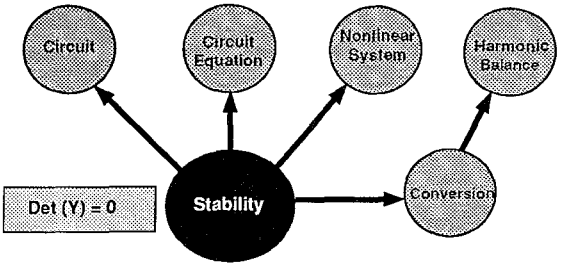

Figure 4e - Stability analysis of nonlinear circuit

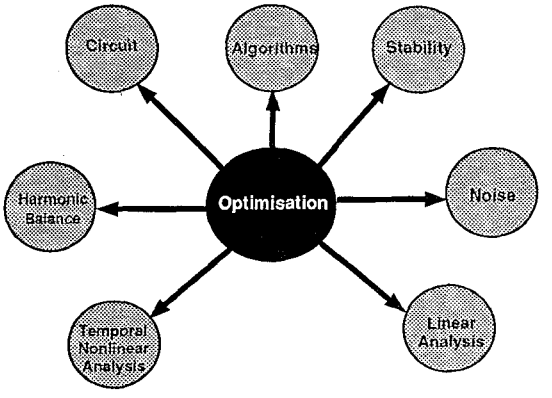

Figure 4f - Circuit Optimization