That's the Way It *Should* Work!

Recently, I was fortunate enough to see a demonstration of the new Taligent™ operating environment. Although it was only in beta form, it was quite impressive. Its intuitive drag-and-drop interface (called People, Places and Things™) allows you to create a "business card" to identify a person, and then use that business card icon to perform various actions related to that person. For example, to call that person, drop the business card on the telephone icon; to write a letter, drop the business card on the letter icon; and to sign the letter, drop your business card on the letter.

The demonstration consisted of a compound document that included text, a drawing, a full-motion video with audio, and a continuously updating ticker tape that could be instantly changed to display in any international character set. Also demonstrated were tools such as a highlighter, which could be used on any kind of data, whether text or graphics. The demonstration showed how to teleconference over a modem connection (simulated in the demo) using graphical displays that could be changed by either person, with the changes being immediately apparent to the other person. The resulting drawing could then be embedded in the compound document to become both—at the same time—an order for warehouse space (in the example) and a confirmation of that order. All these interactions were accomplished by program modules containing from 12 to 1,200 lines of code.

But the most amazing thing about the demo was that, in retrospect, it wasn't amazing at all. Chris Nelson, a panelist in the panel discussion in this issue, likes to say in his presentations on OpenDoc™ that, "...when the end user sees the results (of object technology), it will be perceived as a bug fix." And so it was with the demo. Every event was handled in a way that was natural and just the way that you'd expect to do it. You didn't have to worry about changing the format of data as you moved from one tool to the next; it just happened. Although there was a menu bar, you didn't need to access it for any of the actions shown in the demo—the most common tasks could be accomplished without it. Every time a particular function was shown, you could think, "That's the way it *should* work."

And so, I knew that we had to do an issue on Object-Oriented Programming (OOP). To ensure that our readers are kept abreast of this rapidly changing topic, we developed this special issue to introduce the topic. It includes a discussion with Cliff Reeves to provide the flavor of IBM's commitment to object technology. To address the current issues in object technology, we have included excerpts from a panel discussion of object technology that was held at the recent POWER Conference in Dallas, Texas. We also have articles on programming with SOM/6000, an update on Taligent's direction, a look at OpenDoc (reprinted from *BYTE* magazine), and a plan for preparing to develop object-oriented solutions as the new tools become available. We hope you find this issue helpful in preparing for the rapidly approaching world of object-oriented programming. Look to future issues of *AIXpert* for additional object-oriented articles as the technology unfolds.

George Noren

George Noren, *IBM Corporation, Internal Zip 2830, 11400 Burnet Road, Austin, TX 78758. Internet: geo@austin.ibm.com. Since joining IBM in September 1979, Mr. Noren has written manuals for System/34, System/36, and AIX on both the RT® and RISC System/6000® platforms, and was a member of the InfoExplorer™ design team. He has also worked as system administrator for several AIX server machines and their clients, and is currently responsible for the Prototype Evaluation Labs in Austin. Mr. Noren studied engineering at Illinois Institute of Technology, holds a BA in English from the University of Minnesota, and an MBA from St. Edwards University in Austin.*

Pick up from previous issue

George Noren