
Editorial

This issue of *EP-odd* is dedicated to active documents. The evolution of documents that has occurred over the last 20 years has resulted in a number of species in the electronic document family. The active document is one of them. Like many others it is based on an abstract representation of documents, but more than any other it strongly depends on a computing environment.

Word processors, desktop publishing systems and formatters make use of computers for preparing of documents, but the results they produce can be read on paper, in a traditional way, without any help from a computing system. When the result has been produced, computers are no longer *required* for using the final document, even though they can still be useful in helping the reader browse through a document. With active documents, a computing system is needed not only in the preparation phase, but also when using them. This is because each time the document is accessed by a user or a program it can be seen a different way and it can present a different content depending on the computing environment used for accessing it. In fact, an active document can be defined as a document that transforms itself and/or its computing environment according to the state of that environment and to the editing operations performed on the document. In this definition, the word 'environment' should be considered in a very broad sense, since it includes users as well as software and hardware.

This concept of an active document is mainly used in document-centered applications. The three papers selected in this issue show different applications implemented with that concept. Haake, Hüser and Reichenberger describe an example of an active publication: the Individualized Electronic Newspaper (IEN). It is a publication that is individualized and composed on demand for a reader according to a profile of the reader's interests. The other two papers present environments that can support various types of applications and they give some examples of applications: auto-localizing documents, multimedia employee directory, intelligent maintenance aid for field mechanics, proofreading tool, annotation tool, electronic index, cooperative editing, specialized editors, etc.

Most of the first systems developed in this field employed active documents as user interfaces. Human-computer interaction is still an important field of application for active documents. The three papers in this issue address the issue, in two different ways. They present examples of document-based user interfaces, but they also discuss the problem of building a user interface for manipulating active documents.

Another major issue in active documents is the one of system architecture, which is also addressed by the three papers. Developers of active document systems are faced with the problem of designing a system that handles electronic documents, with all the usual aspects, but they also have to care about document activity, i.e., the relationship between various document components and the corresponding actions that must be performed on certain events. Different approaches are proposed in the three papers. The IEN is based on a specific environment that has been built for the application. The other two papers present extensions that have been made to existing document preparation systems.

In all of the papers, the document model is considered as a key issue. It is interesting

to notice that the three systems presented are based on a structured model of documents and that two of them use the SGML approach of a logical structure constrained by a document type definition. It seems that active documents is a domain where document structure can bring a number of advantages.

Although active documents are considered as a major advance, they still pose some problems, which are highlighted in this issue. As well as multimedia documents (in fact active documents can also be multimedia documents, like the IEN), active documents are real electronic documents, which must be designed for interactive presentation as well as for a static paper presentation. Document designers have to decide how active documents should look and they must use metaphors that are well suited to the kind of manipulations users will perform on active documents.

Different means are proposed for the specification of activities in documents. This variety allow to develop various types of applications, but it also prevents active documents being exchanged between different types of system. It is probably too early for designing standards for active documents, but this concept will probably be of common use only when some well-accepted standard becomes available.

Development tools are also needed for helping users to make documents active, (without needing to learn a complex programming language), for making debugging and maintenance easier, etc. Like programs, active documents need a specific development environment.

Comparisons between programs and documents have often been made in the literature. With active documents, a new step is forged in combining the two. This convergence will provide more powerful applications—that will also be easier to use—by presenting users with very familiar objects: documents.

VINCENT QUINT, GUEST EDITOR