Silicon Graphics' IRIS InSight: An SGML Success Story

Robert J. Glushko
Passage Systems, Inc.

Ken Kershner
Silicon Graphics, Inc.

SUMMARY

This article is a case history of the development of the Silicon Graphics *IRIS InSight™* system, the first system for viewing online documentation from a computer vendor that uses SGML, the Standard Generalized Markup Language. We describe the SGML publishing process from the perspectives of authors, production staff, and management. We review the key decisions and turning points in four phases of the project:

1. **Project initiation and requirements**
2. **Design and development**
3. **Process characterization and institutionalization**
4. **Deployment and enhancement**

This article tells the story of Silicon Graphics' *IRIS InSight,*™ the first system for viewing online documentation from a computer vendor that uses SGML, the Standard Generalized Markup Language (ISO 8879) (Goldfarb 1990). SGML's explicit encoding of structure and its separation of structure and presentation make possible structure-based search, alternative structural views of the same information, dynamic reformatting, and alternative presentation styles. Silicon Graphics chose this technology because it enables them to produce more consistent and complete documents while automating many otherwise tedious production tasks. Over time, SGML will allow the company to reuse of much more of the information it creates and to synchronize the development of documentation with the products it supports. Since late 1992 the *IRIS InSight* viewer and the core *IRIS InSight* Document Library, which contains several technical manuals, have been bundled with every Silicon Graphics computer. *IRIS InSight* is a family of tools that provide online access to product, support, and technical information and to various electronic services.

We begin this article with the user's perspective on *IRIS InSight*. Although SGML is the basis of many of the viewer's functions, SGML is not directly visible to end users. It is visible and important to varying degrees to authors, production staff,

This article has been peer reviewed.

management, and others involved in the overall *IRIS InSight* effort. We explain these different perspectives so that others can appreciate how the costs and benefits of SGML are allocated among different individuals and organizations in an overall SGML-based publishing process.

Since the *IRIS InSight* effort began in late 1990, we've seen an enormous increase in the visibility and viability of SGML. The growing success of SGML makes our decision to adopt it seem far more obvious and easy than it was at the time. We review the key decisions and turning points in the project's history to identify important lessons for others considering a similar project.

INTRODUCTION TO *IRIS INSIGHT*

IRIS InSight is an information viewer that takes advantage of the rich visual processing capabilities of Silicon Graphics computers while providing all the familiar entry points into a book. Figure 1 is a "screen shot" showing a typical user interaction with the system. Many of the low-level features of *IRIS InSight* reflect its conformance to the user interface standards embodied in the *Motif* toolkit and the *X Window System*.

The top pane in Figure 1 shows the *IRIS InSight* document library, in which books for different types of users or applications are arranged on named bookshelves. A user can directly view an online book by double-clicking on its icon or title. Alternatively, the user can search for information in one or more selected books by single-clicking on them and then typing the search terms or phrases into the search field (in the middle of the screen shot in Figure 1). Boolean and proximity operators can be used to refine a search. The sections of books in which the search terms can be found are displayed in a "hit list" (the lower pane of Figure 1).

The display of a selected book or book section appears as in Figure 2. The display is organized as a "structure view" in the top pane and a "content view" below it. In Figure 2, the structure view in the

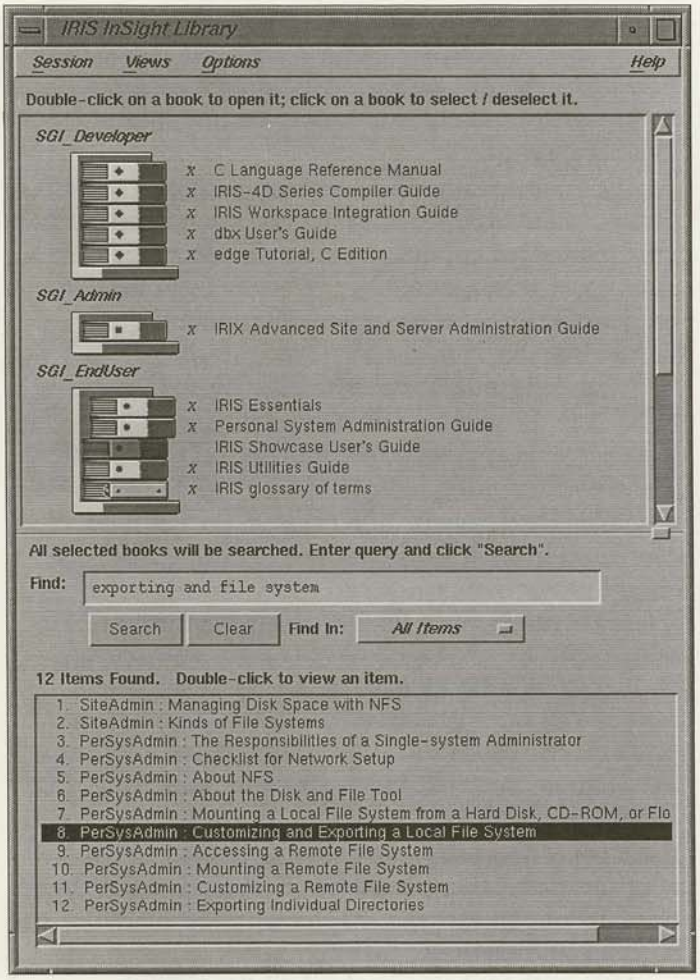

Figure 1. Selecting a book or topic in **IRIS InSight**

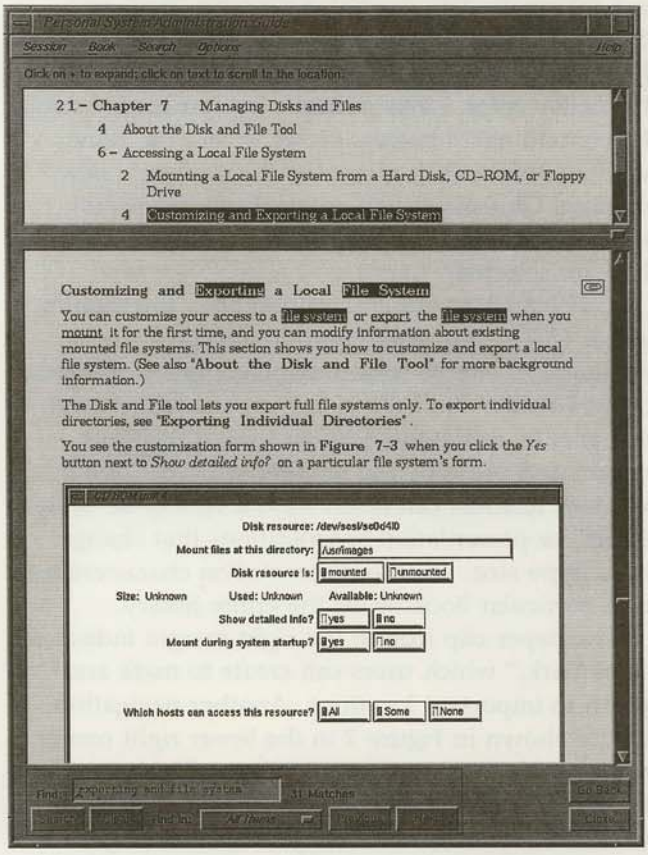

Figure 2. Displaying a topic in **IRIS InSight**

top pane is the book's table of contents, presented as a dynamic outline for progressive display of details (Furnace 1986; Raymond 1992), but other structure views like lists of tables, figures, or code examples are also available. Searches can be restricted to a particular kind of content element for more precision; for example, a user can search only in titles, figure captions, or code examples. The content and structure views are synchronized so that a user can quickly navigate to any part of the book by double-clicking on an item in a structure view.

The user's current position in the book is indicated by a highlighted title in the structure view. Numbers in front of each title indicate the number of matches to the current search expression (indicated in the within-book search field at the bottom of the window in Figure 2). This structured search feedback technique, first used by SuperBook (Marcus 1991), has been shown to enhance search and navigation in online publications (Egan et al. 1989).

In the content view in the bottom pane of Figure 2, the text of the book appears in a scrolling window. This display is not page-based; the window can be arbitrarily resized, and the text is wrapped and reformatted as needed. Selectable "See" and "See also" cross references to another section or book (hypertext links) are marked on the screen in a distinctive color. Links within a book instantly scroll the content and structure views within the same window, while links to another book open a new window. Glossary terms, marked by underlining, open a window containing their definitions when they are selected.

IRIS InSight books can contain links to graphics, audio, video, animation, and other digital media elements. A sample screen display appears in Figure 2, but a reader can instead specify that all nontext elements be represented by an icon at the right margin and presented in another window only if selected. Readers can select from a variety of "style sheets" or presentation specifications that change fonts, type size, or other presentation characteristics for a particular book or for the entire library.

The paper clip icon in the right margin indicates a "bookmark," which users can create to mark and return to important locations. Another navigation feature shown in Figure 2 in the lower right corner of the window is a GO BACK button, which retraces the reader's steps to display previously displayed books. Readers can also navigate by selecting hypertext links from entries in an alphabetic "back-of-the-book" index.

Many of the most important features in *IRIS InSight*, including structure views, structure-based search, dynamic reformatting, and alternative presentation styles, depend on its use of SGML. Nevertheless, SGML is both invisible and irrelevant to users. To explain how it is perceived by others, we must first describe the publishing process as it currently exists at Silicon Graphics.

THE PUBLISHING PROCESS AT SILICON GRAPHICS

Almost all of the technical manuals and other customer-oriented information that is delivered with *IRIS InSight* is initially created by technical writers, engineers, and other personnel using *FrameMaker*. Each chapter of a *FrameMaker* book is translated into SGML using a customized *FastTag* translation program and some additional programs written by Silicon Graphics software engineers. The translation process involves much more than a tag-for-tag substitution, because much of the value of getting to SGML is in making hierarchical structure explicit by inserting new tags that serve as containers or wrappers around other tags that identify lower structural levels.

The translation programs identify tagging errors in the *FrameMaker* files that result from inconsistent or nonstandard application of *FrameMaker* tags. The author can repair most tagging errors by editing the original file in *FrameMaker* and rerunning the translation program.

Although it might be more efficient to edit the SGML file directly to fix a tagging or translation problem, this approach has the drawback for most authors of requiring them to use an SGML editor, an unfamiliar and non-WYSIWYG tool. Furthermore, changing the SGML file and not the *FrameMaker* file would create two versions of what is supposed to be the same thing, since the printed versions of documents are produced directly from *Framemaker*.

Insulating authors from SGML by letting them continue to work in a familiar desktop publishing program has its obvious advantages. However, since the program does not interactively enforce compliance with its template or stylesheet, the feedback to the author comes only in batch form after the translation program is run. Even on a fast workstation, an average chapter can take many minutes to translate to SGML.

After authors have resolved all the tagging problems reported by the format translation

programs, they give the "clean" SGML files and the corresponding camera-ready hardcopy to production personnel. At this stage all of the tags in the original *FrameMaker* files have been converted to SGML tags. Hypertext links have been derived from the cross-reference markers created by authors in *FrameMaker*. There is yet no guarantee, however, that the resulting SGML file completely conforms to the document type definition (DTD) that formally defines the structure of a valid document. To validate the SGML files, it is necessary to run them through an SGML parser. Often, the most useful function of the parser is to report logical omissions in the source file, such as the lack of a second-level heading between first- and third-level ones.

The validated files are then run through as a set of "book manufacturing" programs that transform the existing back-of-the-book index into its hypertext counterpart, create the full-text index needed for the full-text search functions, and build the other files needed for efficient viewing and navigation of the book. Graphics and images require their own specialized translation and compression programs before they can be viewed by *IRIS InSight*.

Managers in technical publications and the managers of products served by publications organizations do not need much knowledge of SGML, but they need to be structure aware and SGML aware.

The ultimate test of the process is that the book looks and behaves in *IRIS InSight* as the author expects it will. In some situations a book translates from *FrameMaker* without errors and is validated by the parser, but it still needs some minor adjustment in online formatting. Over time, many of these residual problems have been eliminated by improvements to the *FrameMaker* templates or to the translation programs.

Finally, the *IRIS InSight* files are collected and prepared for mastering and delivery on CD-ROM.

Perspectives on SGML in *IRIS InSight*

Because Silicon Graphics authors work with *FrameMaker* rather than with authoring tools that directly create SGML files (as discussed by Davidson elsewhere in this issue), they need to be "structure aware" but do not need specialized knowledge of SGML syntax or technology. By "structure aware" we mean they need to understand the consequences of their use or misuse of *FrameMaker* templates and tags. The only knowledge of SGML they need is to be able to interpret the error messages produced by the format-translation programs.

Production staff, on the other hand, need more expert knowledge. They work directly with SGML source files and SGML tools and must also maintain the translation programs that convert from *FrameMaker* to SGML.

Silicon Graphics managers in technical publications and the managers of products served by publications organizations do not need much knowledge of SGML, but they need to be structure aware and SGML aware, in a slightly different way from authors. Both kinds of managers need to recognize that the benefits of SGML take time to emerge, and that once they do, they accrue to organizations more than they do to individuals. This realization requires patience and an appreciation that, from individual authors' perspectives, SGML can impose new responsibilities without much perceived value beyond the productivity gains they experience as end users of *IRIS InSight*. Engineering managers who in the past had little understanding of or interest in technical publications seem to appreciate that the *FrameMaker*-to-SGML process imposes an "edit, compile, debug" cycle on authors that has much in common with the work of software engineers.

Project History

It is a convenient simplification to describe the history of the *IRIS InSight* project as involving four primary activities:

1. Project initiation and requirements analysis (late 1990–mid 1991)
2. Design and development (mid 1991–late 1992)
3. Process characterization and institutionalization (late 1991–present)
4. Deployment and enhancement (late 1992–present)

Project Initiation and Requirements Analysis

A project to develop online documentation can be initiated for numerous reasons, and a successful project begins by identifying those that are most important and focusing on them. A major factor in

getting the *IRIS InSight* effort underway in late 1990 was the success of a new initiative to deliver software using CD-ROM. Reducing software distribution costs using CD-ROM made the cost of shippng printed documentation more visible as a cost of sales. We were wary of many people assuming that delivering documentation online was little more than a change of distribution medium, and we spent considerable time educating people about the greater challenge posed by online documentation (Glushko 1989; Kershner and Passarelli 1990).

Our requirements analysis mandated some kind of standards-based approach for online documentation, but we were not in any way predisposed to select SGML.

We recognized that unless the online documents were highly usable, customers would refuse to give up printed documents and there would be little cost savings. A critical part of the internal justification for the *InSight* effort was the impending launch of a low-end (for Silicon Graphics) workstation that would compete with high-end personal computers known for their ease of use. We knew that this expanded customer base has much higher expectations for documentation that the engineering and scientific users who comprised the core of the users for SGI's established product lines. Silicon Graphics couldn't afford an online documentation system that generated more customer support calls than it prevented.

An additional factor was competitive pressure from other workstation vendors. Other vendors were already providing online documentation, and this was beginning to show up in product comparisons. We turned being last into an advantage by conducting a careful competitive analysis and resolving to leapfrog the competition with an aggressive online documentation project that provided new capabilities to users.

Several organizations at Silicon Graphics were considering online documentation, and it was essential for us to focus these disparate efforts into a single company-wide initiative. Convincing the managers of projects for whom online documentation was a third or fourth priority to defer to a project for which it was the primary focus was not difficult, but this consensus building and consolidation took several months.

In early 1991 we conducted a survey of Silicon Graphics developers and customers to ask about their plans and requirements for online document delivery. The results clearly spelled out four requirements:

- Don't use proprietary technology.
- Develop a system that also works on other platforms.
- Add additional functionality that takes advantage of Silicon Graphics' digital media capabilities.
- Use production methods that developers can use as well.

These results seemed somewhat surprising at first. Many people expected that developers would want a system that competitively exploited Silicon Graphics' superior digital media technology. However, since many companies that develop applications for Silicon Graphics computers also develop for other Unix computers and the Macintosh, they needed an online documentation approach that would not be limited to Silicon Graphics' line. Developers wanted to be able to create a single set of documents that could be viewed on all of the computers on which they sold software.

IRIS InSight would obviously have to be based on standards. But which standards?

Design and Development

Our requirements analysis mandated some kind of standards-based approach for online documentation, but we were not in any way predisposed to select SGML. We were probably predisposed **not** to use it because we initially knew much less about it than the two alternatives we considered. The first was Adobe's *Postscript*, the de facto standard for formatted page description already in use by Sun and other vendors as the internal representation for online document. The second was Microsoft's *RTF* (Rich Text Format), the ASCII interchange format for *Word* widely used in the PC marketplace.

We spent several months studying the technical and business issues, and the engineering team demonstrated that either *Postscript* or *RTF* would satisfy Silicon Graphics' requirements for delivering online documentation. But what finally led us to choose SGML was our competitive analysis of the

online documentation offerings of other workstation vendors and the survey of Silicon Graphics developers. Only SGML would enable the shift from a vendor-centered "here's our online documentation" strategy to a customer-centered "here's an online documentation viewer that you can also use" strategy.

Once we chose SGML and began our system design, we had the luxury of several "build vs. buy" decisions. Because SGML is a vendor-independent standard, many companies are building SGML-based products to support various parts of the publishing process. We decided that it would let us ship a product faster and it would be less proprietary if we relied in part on third-party tools. We chose *FastTag* from Avalanche Development for format translation and the *DynaText* toolkit from Electronic Book Technologies to satisfy some of our indexing and presentation requirements.

IRIS InSight takes advantage of digital media capabilities to present video, audio, 3-D graphics, and animation on Silicon Graphics computers, but multiplatform developers can use the basic *DynaText* viewer to deliver some of this information on other platforms with no changes to the object files. Without *DynaText*, we would be unable to meet the cross-platform requirement we identified in the 1991 survey of Silicon Graphics developers.

The detailed design emerged over time as a result of a thorough review of the literature, a detailed analysis of design alternatives, and extensive usability testing.

One initial design goal was to maintain fidelity of content between the printed manuals and those we delivered online. By this we meant that we would not require page fidelity—an exact correspondence between online and printed versions in page numbering, layout, or formatting—but that the structural organization would be the same (Glushko et al. 1988). The first version of *IRIS InSight* we started shipping in late 1992 more or less met this goal.

However, as we have moved more books into *IRIS InSight*, we have come to recognize that content fidelity is not entirely achievable, nor should it be. We have discovered some structural or logical aspects of our printed books that we cannot preserve in the online versions. For example, we have no easy way, other than screen snaps, to duplicate printed graphics that have text callouts or the sidenotes that appear in many of our printed books.

On the other hand, once a critical mass of information entered *IRIS InSight*, users started to rely on the online version as the primary one, so complete content fidelity would limit us from enhancing the online books where it added value. We increased the descriptiveness and number of internal headings in books because of the effective use that the *IRIS InSight* structure view makes of them (Glushko et al. 1988). We have begun to incorporate 3-D graphics, animation, audio, and video into *IRIS InSight* books. We expect that future releases will lead us further from any correspondence with paper documents.

The detailed design emerged over time as a result of a thorough review of the literature, a detailed analysis of design alternatives, and extensive usability testing. (We highly recommend Horton 1990, Kahn et al. 1990, and Marcus 1991). We recognized early that acceptance of *IRIS InSight* depended on users' willingness to rely on online documents as their primary source of information.

Human Factors engineers from the customer research and usability group conducted three usability studies with both experienced customers and novice users. The studies were designed to ensure that *IRIS InSight* would be preferred by users and that it wouldn't generate calls to customer support. These studies contrasted several specific design alternatives and objectively measured performance on information-retrieval and problem-solving tasks. The most important of these alternatives were the "granularity" or "node size" into which online books should be organized and the appropriate behavior of hypertext links—for example, whether to scroll within the same window or open a second window.

These carefully controlled experiments led to the final design in which books were the most salient online unit and in which each distinctive link type had a unique behavior. User performance with the final design was equal to that for the printed manuals, a significant achievement given the limited experience users had with it. Finally, users strongly preferred *IRIS InSight* to the printed manuals.

Process Characterization and Institutionalization

Just as we knew we couldn't design a product with only vague requirements for putting documents

online, we knew we had to carry out a systematic "reengineering" effort to examine the existing end-to-end publication process. In part this caused us to discover problems we didn't know we had, such as the extent to which authors often invent ad hoc tags to meet the visual requirements for camera-ready copy rather than use only the official templates and tag sets. Systematizing and restructuring the publication process was made more difficult by Silicon Graphics' periodic reliance on contract authors to meet publication schedules, especially for major product releases, because contract authors often work from remote locations and are otherwise removed from the mainstream publications culture.

We gradually recognized that SGML was having a profound institutional impact at Silicon Graphics and was qualitatively changing the publication process. We were imposing significant new responsibilities on technical publications personnel.

SGML changes the relative importance of formatting and structure. When authors are charged with delivering camera-ready copy, they are tempted to insert formatting tags until the document looks just right. A new requirement to deliver valid SGML files directs an author's attention to more precise structural tagging that gives customers the advantages of online presentation but which requires more work. Authors value the flexibility provided by the multimedia capabilities of *IRIS InSight*, but only if they are given the time to take advantage of them.

We chose an incremental strategy for moving to SGML that attempted to minimize the disruption to existing methods and technology. Because Silicon Graphics had just made a major investment in technology and training to standardize technical publications on *FrameMaker*, we knew we had to design a publication process based on format translation to SGML from *FrameMaker* rather than adopt SGML-based authoring technology directly. This strategy also preserved Silicon Graphics' investment in using *FrameMaker* to create hardcopy documents.

We brought in consultants to conduct document analyses to help us standardize our use of *FrameMaker*. This was a critical step because authoring standards are essential if automated translation to SGML is to work. We discovered that although different writing groups at Silicon Graphics had their own templates, we were able to extract a common document architecture and encode it in a single DTD.

The DTD is the linchpin of an SGML application because it formally defines the structure of a valid document instance. It establishes the target for format translation and is the basis for alternative presentation styles. But what authors see most directly are the templates in *FrameMaker*, and any changes to these templates ripple into the DTD and the software for format translation and online presentation.

At the time we didn't fully appreciate this central importance and vulnerability of the DTD in the overall publication process. For months several people on the team worked on different pieces of the end-to-end process, and a change that might have been an improvement in authoring templates or other parts viewed in isolation would become a destabilizing bug to other pieces. It took several months for us to stabilize the production process.

Training and retraining of authors and production personnel has been essential. Initially our authors attended courses taught by popular industry consultants. But as we have learned more about SGML and online documentation, there has been a major shift in training to very specific "here's how we do it at Silicon Graphics." Our needs can no longer be met by the broad and relatively unspecific training available from outsiders. Instead, our authors have translated what they learned into Silicon Graphics-speak, and it now exists in a detailed publications style guide as an *IRIS InSight* book. This book enables authors to become aware of SGML and *IRIS InSight* even if their books are not going online right away. Authors and production personnel have also instituted one-on-one and group training in which they teach each other how to write for online presentation and how to build books successfully.

The production process still needs to be more transparent for authors. We are currently working on tools for authors and production personnel that will hide much of the complexity of SGML, especially the nuances of the format translation and indexing tools.

Deployment and Enhancement

Bundling *IRIS InSight* and its core document library as part of system software and charging for printed manuals overwhelmingly biased the cost-benefit equation in favor of online documentation. This economic incentive encourages customers to consider relying on online documentation as their primary version.

We believe that focusing on issues of putting text

online before we moved to incorporate digital media into the books was critical to the success of *IRIS InSight*. Maintaining this conservative tack was difficult at times, but we had seen other online documentation efforts fail because they were seduced by the excitement and hype about multimedia, leading to books with "Macbeth multimedia"—full of sound and fury, but signifying nothing (our apologies to Shakespeare) (Glushko 1992). We knew that *IRIS InSight* would never be scalable, cost effective, or usable by our authors or customers if we relied on handcrafted production techniques of the sort encouraged by premature experimentation with multimedia (Alschuler 1989, Glushko et al. 1988).

SGML enabled Silicon Graphics to design and develop* IRIS InSight, *a system for delivering online documentation, while achieving its objective of significantly reducing the costs of providing that information to customers.

Now that we have a robust process for converting print-based books to online, we can confidently provide the *FrameMaker* templates, the DTD, and the associated format translation and producton software to those wanting to use *IRIS InSight* to become online publishers on their own. Since *IRIS InSight* text files are completely compatible with those used by *DynaText* on other Unix platforms, on the Macintosh, or under *Microsoft Windows*, a multiplatform stategy for online documentation is completely viable. *IRIS InSight* can use any SGML DTD, so users remain free to choose another document architecture if they need one for a specialized application. However, by providing a core of SGML technology tailored for *IRIS InSight*, we have greatly reduced both the cost and the risk of adopting SGML.

Summary

SGML enabled Silicon Graphics to design and develop *IRIS InSight*, a system for delivering online documentation, while achieving its objective of significantly reducing the costs of providing that information to customers. But the transition to SGML involved a steep learning curve. We had to define an SGML-based publishing process in terms of steps or functions like authoring, format conversion, indexing, viewing, and information management so that we could select or design software to support each activity. This process characterization took place iteratively because we didn't have a complete end-to-end perspective when we started. When candidate tools were identified and "plugged into" a tentative end-to-end solution, the requirements for other pieces were often unknowingly changed or constrained. We didn't fully appreciate at the time that the publishing process was both a technological and an organizational creation that would require significant changes at Silicon Graphics.

The expertise required to design and develop *IRIS InSight* was substantial. We hired numerous consultants, but their participation then imposed the additional task of integrating their different perspectives and expertise into what we were doing on our own.

The substantial increase in the visibility of SGML technology and concepts today and the benefit of hindsight make it easy to say that we made a good decision to use SGML. It is certainly viable today for real applications, and we hope that our case history provides some stepping stones on a successful path to using it. Nevertheless, we advise those reading this article to recognize that even for us, SGML was only a means and not an end for online documentation. The ultimate goal was always usable and cost-effective online documentation.

Acknowledgments

The *IRIS InSight* project involved dozens of people in software engineering, technical publications, release engineering, usability testing, and customer support organizations at Silicon Graphics. Avalanche Development, Electronic Book Technologies, and several external consultants played essential roles, as did the Silicon Graphics developer companies who participated as survey respondents and beta sites. We hope they accept our collective thanks here and appreciate our concern that if we named them individually, we might inadvertently leave someone out.

References

Alschuler, L. 1989. "Hand-crafted hypertext: Lessons from the ACM experiment." In E. Barrett, ed., *The society of text: Hypertext, hypermedia, and the social construction of information*, pp. 343–361. Cambridge, MA: MIT Press.

Egan, D., J. Remde, L. Gomez, T. Landauer, J. Eberhardt, and C. Lochbaum. 1989. "Formative design-evaluation of SuperBook." *ACM Transactions on Information Systems* 7(1): 30–57.

Furnas, G. W. 1986. "Generalized fisheye views." In M. Mantei and P. Orbeton, eds., *Human factors in computing systems*, pp. 16–28. New York: ACM.

Glushko, R. J. 1989. "Transforming text into hypertext for a compact disc encyclopedia." In *Proceedings of the ACM Conference on Computer-Human Interaction–CHI '89*, pp. 293–298.

Glushko, R. J. 1992. "Seven ways to make a hypertext project fail," *Technical Communication* 39: 226–230.

Glushko, R. J., M. D. Weaver, T. A. Coonan, and J. E. Lincoln. 1988. "'Hypertext engineering': Practical methods for creating a compact disc encyclopedia." *Proceedings of the ACM Conference on Document Processing Systems*, pp. 11–19.

Goldfarb, C. F. 1990. *The SGML handbook*. Arlington, VA: Graphic Communications Association.

Horton, W. 1990. *Designing and writing online documentation*. New York: John Wiley & Sons, Inc.

Kahn, P., J. Launhardt, K. Lenk, and R. Peters. 1990. "Design of hypermedia publications: Issues and solutions." In R. Furuta, ed., *Proceedings of the International Conference on Document Manipulation, and Typography*, pp. 107–124. Cambridge University Press.

Kershner, K., and B. Passarelli. 1990. "Compact distribution." *Unix Review* 8, no. 10: 64–69.

Marcus, A. 1991. *Graphic design for electronic documents and user interfaces*. New York: ACM.

Raymond, D. 1992. "Flexible text display with *Lector*." *Computer* 25, no. 8: 49–60.

Remde, J., L. Gomez, and T. Landauer. 1987. "SuperBook: An automatic tool for information exploration—Hypertext?" In *Hypertext '87 Proceedings*, pp. 175–188.