ELECTRONIC PUBLISHING, VOL. 6(4), 339–348 (DECEMBER 1993)

Document reuse and document systems

DAVID M. LEVY

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA

email: `dlevy@parc.xerox.com`

SUMMARY

While reuse is currently the focus of much attention in the programming language community, it is also a central, but less noticed, issue in the creation and use of documents, and therefore in the design of document systems. To a great extent, the work of producing new documents, and new versions of old documents, involves reusing pieces of previously existing documents, where reuse involves finding the relevant material, modifying it as needed, and stitching the pieces together. The objective of this paper is to demonstrate how a focus on reuse can shed light on current efforts to build structured document systems and to design and use standards, such as SGML, ODA, and OLE, that address structured and compound documents.

KEY WORDS Reuse Structured documents Compound documents Document standards ODA SGML OLE

> ‘Everything made now is either a replica or a variant of something made a little time ago and so on back without break to the first morning of human time.’ — *George Kubler* [1]

1 INTRODUCTION

Reuse is a subject now of much discussion and debate in programming language design [2]. Object-oriented languages, for example, are touted because of the supposed ease with which pieces of programs can be modified and reused for other purposes and in other environments [3]. During the past year, in the course of doing a study of the state of the art of ‘structured documents,’ I have come to see that reuse is also a central category and concern in the creation and use of documents.[1] To a great extent, the work of producing new documents, and new versions of old documents, involves reusing pieces of previously existing documents, where reuse involves finding the relevant material, modifying it as needed, and stitching the pieces together. This method of production is as old as documents themselves; we have done it with paper-based technologies and are working out how best to perform it with the new digital media and mixed-media technologies.

[1] I am certainly not the first to notice the importance of reuse. In the first pages of [4], Furuta, Quint and André clearly explicate their work on structured documents in terms of reuse — for example, ‘[our] goal in the design of a document preparation system is to identify a document representation that permits the flexible reuse of the document and its components’ (page 20). But as far as I know, I am the first to take reuse as the centre point of my exploration, and to attempt to follow out its implications to structured documents and beyond.

CCC 0894–3982/93/040339–10

Received 15 August 1993
Revised 1 December 1993

My objective in this paper is to demonstrate how a focus on reuse can shed light on current efforts to build structured document systems and to design and use standards, such as SGML [5], ODA [6], and OLE [7], that address structured and compound documents. I begin by laying out some key characteristics of documents, focusing in part on how they come together through the collection and combination of material from diverse sources. In Section 3 I present the basics of reuse. I then use this perspective in Section 4 to analyse structured document standards, and conclude in Section 5 with some thoughts about how the reuse perspective might influence future document research and development.

2 WHAT ARE DOCUMENTS?

Despite the fact that dictionary definitions tend to focus on two senses of documents — that they are written on paper and that they have legal or evidentiary status — the word ‘document’ has in recent times come to denote a much broader class of communicative artifacts, including audio and videotapes, electronic forms, and multimedia presentations. Without trying to provide a complete or all-encompassing definition of documents, I want to focus on three characteristics of this broader class which are important to my story about reuse — namely, that documents are:

1. the localization of communicative material for particular purposes; and
2. instances of recognizable social types;
3. which are realized in a stable form.

2.1 Localization of communicative material

Documents are the coalescence of communicative stuff (text and images) in a place or locale — on a page or screen (in visual form), or on a disk or tape (stored for presentation). Even when the sources may be in some sense distributed, as when parts of the document reside in a hypertext web on multiple servers [8], the material has still been coalesced for the reader into a single unity that is available in one place. This is a key feature of the document, that it pulls together material from a range of sources and makes it locally accessible.[2] Moreover, this coalescence always comes about to satisfy particular purposes or to enable some range of uses. This is crucial, because it is only with respect to use (projected or actual) that we can understand why certain material (content) has been selected or how it has been organized (structured). Document structure is always closely related to use; structure and use are duals of one another.[3]

2.2 Instances of recognizable social types

One of the ways to get at the relationship between document structure and use is via an understanding of document genre [9,11]. Documents come into being as instances of

[2] I prefer the term ‘material’ to ‘information,’ because the former suggests the materiality, the physical embodiment, of the stuff available to us, in contrast with the latter, which suggests a free-floating, disembodied commodity. Nunberg [9], for example, has called attention to the wide-spread, but problematic, notion that information is bottled in documents and decanted into the heads of readers.

[3] For computer scientists, the relationship between structure and use is driven home in the first data structures course where one learns that how one organizes a data structure — for example, as a singly or doubly linked list or as an array — will determine how, and how fast, one can perform various operations, such as searching and sorting. Abelson and Sussman [10] have a particularly nice treatment of the duality of structure and process.

recognizable social types or genres — as newspaper articles, travel brochures, journal articles, purchase requisitions, and so on. Their form tells us a lot about what they are (what their genre is), and knowing what they are telegraphs a great deal to us about how to use them, and what institutions and practices lie behind them. (Indeed, without recognizing the genre of a document, we don't know what to do with it.) I don't need a user manual to read a newspaper; I know that position of article and size of headline signal importance, and I know how to interpret an op-ed article differently than a front page news story. I also have some understanding of the institutional apparatus by which news is gathered and distributed, and this influences the ways I will interpret what I read. Thus genre cues provide an extremely efficient way to supply information about the context of the document and its intended use.

2.3 In a stable form

Another crucial property of documents is their stability, that they remain 'the same' as they travel through space and time, providing a fixed point which people use to orient themselves.[4] In the simplest case, this stability comes about because relatively permanent marks can be inscribed in a surface. An added wrinkle is the ability to make copies of documents, whether by pen, press or copier, so that multiple documents are created, each one of which is stable (unchanging over some period of time), all of which are 'the same.'[5] With some of the newer, dynamic document technologies, though, stability cannot be based on immutability of marks. Indeed, in the case of video, stability is ensured by the fact that 'the same' performance can be shown again and again, even if each performance is necessarily a dance of changing images.[6]

3 DOCUMENT REUSE

While documents provide a measure of stability in an ever changing world, they themselves are subject to change, as are all material and social artifacts. Since documents are communicative artifacts and represent truths about the world, they must be updated as the world changes. Old documents must be continually updated and new documents brought into existence to reflect and address new states of the world. These new documents and new versions often share much material with older documents, just as new states of the world share much with earlier states.

Here then is how reuse enters the picture: The world, though continually changing, is changing incrementally. Much remains the same (unchanged) at any one time, at least at the granularity of description we typically care about. This means that documents only need to be updated incrementally; and incremental updating is more easily achieved when existing material is reused. Compare the use of a typewriter and a word processor. A typewriter typically requires the retyping of an entire document to make all but the smallest changes, while with a word processor, one only changes those portions of the text that need revision;

[4] Latour [12] calls artifacts that exhibit these properties 'immutable mobiles.'

[5] Sameness is of course a notion deserving of some attention. In [13], we define sameness to be interchangeability, or intersubstitutability, of documents, which makes sameness dependent on use.

[6] Another interesting case is that of interactive documents, where form and/or content may change in interaction with the reader/user. To understand what is stable in these instances, we may need to examine the stability of the programs through which these documents are realized. (Program stability, however, is no more a simple matter than is document stability.)

the rest is reused as is. In both cases there is reuse: in the case of the typewriter one reuses portions of the text (an abstraction, not the actual material) while in the case of the word processor one gets to reuse the ASCII representations of the material that has not been changed.[7]

The process by which new documents, and new versions of old documents, come into existence can be further elaborated in terms of the '4 C's': creation, collection, combination, and customization. Creation is the production of new material, accomplished for instance by inputting text via keyboard or writing on a piece of paper. Collection is the identification and gathering together of previously existing material — documents or pieces of documents. Combination is the stitching together of new and old material to form a new unity.[8] Customization is the reworking of this material to fit its new setting. Collection, combination, and customization all involve reuse; only creation introduces new material.

The 4 C's are my attempt to clarify the process by which new documents arise incrementally through reuse. According to this model, all documents, and versions of documents, come into existence through some combination of these operations. Simple cases involving reuse are those in which a single document is modified, as in annotating a memo and passing it on to a co-worker, or editing a digitally represented text to fix a spelling error. More complex cases abound, and are beyond the scope of this paper to describe in any detail. But I think it is clear that many documents are created — and re-created — through complex processes involving multiple acts of collection, combination, and customization. This paper, for example, has come about in part through the reuse of material drawn from talks I have given over the past year; it also incorporates a quotation, which is a conventionalized form of reuse of text.

A notion closely allied with reuse is that of transformation. Transformation of course *is* a form of reuse, since it is the alteration of something, perhaps to suit new purposes or to better address old purposes. And reuse quite often involves the transformation of material — for example, when I move text or a figure to a new document (collecting and combining), it is not unusual for me to modify it to fit its new context.

Once reuse becomes visible as a central feature of document production and use, one sees it everywhere. Thus, the photocopier is a device for reusing documents — for using a single document to make more documents. But the copier doesn't just make 'copies'; the common 'lighter/darker' and 'enlarge/reduce' controls allow originals to be transformed for new purposes. And the copier makes it possible to reuse parts of documents as well; thus we fold originals to copy portions of, rather than entire, documents and cut and paste pieces of copies into new documents. Then too, devices like recirculating document handlers and digital copiers that store electronic masters can be seen to decrease the time and cost involved in reuse.

An appreciation of reuse also sheds light on the debate over the relative merits of paper and electronic documents. Paper, it is my sense, is generally perceived to have a lower reuse value than electronic forms; it is assumed that one can reuse paper to a lesser extent. But the truth may be less clear-cut: While digital representations promote reuse in principle, current practice suffers under severe limitations (more about which below). And paper currently supports at least one reuse technique, annotation, better than does digital.

[7] Of course, we need to distinguish the digital representation from the visual presentations derived from it, on paper or on screen. Strictly speaking, only the former is reused.

[8] A precursor to this formulation was the notion of 'collect-and-stitch,' developed in [14]. See also [15], where O'Day and Paepcke discuss the collection of 'information' to create 'information compounds.'

Research on document scanning and recognition can be seen as one attempt to increase the reuse value of paper; techniques to support the annotation of electronic documents can be seen as attempts to augment the reuse value of electronic documents.

Reuse is of course a central concern in the processing of electronic documents. Text and graphics editors on personal computers allow us to create stable electronic representations, which can be reused in at least three ways: First, from a single electronic representation, an unlimited number of presentations (copies) can be produced on screens and on paper (call this replication). Second, new versions of a document can be made by editing the electronic representation (redaction); here one only changes what needs changing, while the rest of the document remains the same. Third, portions of a document can be easily taken from one document and moved into another (extraction).

Of course, the word ‘easily’ needs to be used with some care. As of a decade ago, it was easy to move material from one document to another so long as one did not cross a system, application, or platform boundary. There was no problem reusing material if I stayed, say, within Microsoft Word running on a Macintosh, but once I wanted to move material from Word to WordPerfect on the PC, say, I was out of luck. One response to this problem has been the development of various document interchange standards, such as SGML, ODA, and OLE, which can be seen as attempting to extend the boundary within which reuse can occur. In the next section, I look at what these standards are trying to accomplish, framing their objectives and capabilities in terms of reuse.

4 STRUCTURED DOCUMENT STANDARDS

Examining the current document interchange standards from the point of view of reuse, I have identified four capabilities or approaches (see Table 1) that are present to varying degrees in these standards: interchange, composite structure, presentation independence, and meta-structuring.

By interchange, I mean the ability to move a representation of a document from one environment to another, for example, moving a Microsoft Word document prepared on a Macintosh to a PC running WordPerfect. This is a form of copying — the attempt to preserve certain features of the document, possibly including how it looks and how it can

Approach	**What is it?**	**Form of reuse**
Interchange	Copying material from one environment to another	Moves material, enabling collection
Composite structure	Specifying how pieces can be put together to make new unities	Enables combination, as well as extraction and recombination
Presentation independence	Representing material independent of presentational characteristics	Basis for transformations to customize to different physical realizations
Meta-structuring	Representing abstractions common to various doc. representation schemes	‘Meta-customization’

Table 1. Approaches to reuse in document standards

Approach	SGML	ODA	OLE
Interchange	Yes	Yes	Yes
Composite structure	Yes	Yes	Yes
Presentation independence	Yes	Yes	No
Meta-structuring	Yes	No	No

Table 2. Approaches to reuse in SGML, ODA, and OLE

be edited. A second capacity is the ability to represent composite structure — to represent something as a set of interrelated pieces. The third approach, presentation independence, is concerned with separating a document's content from its final physical or visual form — often referred to as the distinction between the logical and the layout structure of a document. This allows for two kinds of reuse: the same content can be reused to produce a number of different final forms — paper, screen, CD-ROM, etc. — for different purposes;[9] and the same layout can be reused to present different contents. Finally, 'meta-structuring' attempts to provide the means for generating document representation schemes. Rather than trying to legislate what features of documents should be represented or at what level of structuring, this approach provides a common set of abstractions (for example, for dealing with hierarchy) out of which particular representation schemes can be created.

All four of these approaches address the problem of reuse, albeit in different ways. Interchange is a means of moving material into new environments where it can be collected and combined — and thus reused. Composite structure is a means of specifying how pieces are to be combined — and thus reused; also, by articulating the pieces of something, it allows them to be extracted and (re)used for other purposes. Presentation independence is a way to represent documents for easy customization, to different media and purposes — another form of reuse. And meta-structuring is a way to represent abstractions common to a range of document representation schemes so these abstractions can be used again and again. (Meta-structuring involves a different order of reuse than the other three; it enables the reuse of document representation techniques rather than the reuse of documents.)

All four of these approaches also involve transformation: Interchange involves, at minimum, transforming a document from one format to another. Composite structure involves transformation, since the process of composition *is* a form of transformation. Presentation independence is concerned with representing documents in such a way that relatively low-cost transformations can be performed to produce final forms suited to different media and purposes. And meta-structuring is the creation of a set of common representation abstractions which can be transformed to produce document representation schemes for particular purposes.

Let us look briefly at how these four approaches to reuse are realized in SGML, ODA, and OLE (see Table 2). Since SGML and ODA are closer to one another in purpose and structure than they are to OLE, I will start by comparing SGML and ODA, and then contrast OLE with both of these.

[9] E.g. 'This separation of the document's contents from its appearance enhances the ability to reuse the document and its parts — the same specification can be transformed into the different physical representations needed to meet specific formatting requirements.' ([4], p. 20)

Both SGML and ODA are intended to enable the interchange of documents across environments. Both permit documents to be described as hierarchies of components and both encourage the separation of presentation independent from presentation dependent structure. In ODA, a type or class of document (a genre in the language of Section 2 above) is represented by constructing a generic logical structure, which specifies how instances of that class (specific logical structures) can be made up of hierarchically-organized components. The equivalent in SGML of ODA's generic logical structure is the document type definition (DTD); SGML's equivalent of specific logical structure is the document instance. ODA also includes within it the mechanism to represent a document's layout structure; SGML does not explicitly do so.[10]

ODA is perhaps best thought of as being (or containing) a document description language. Indeed, as just noted, the distinction between logical and layout structure is built into the language. SGML by contrast is characterized as a 'meta-language' [17] — a language (or notation) for creating document description languages. Although the designers of SGML clearly had documents in mind, the lack of document-specific features has encouraged its use to represent other kinds of entities as well.[11]

OLE, however, is neither a language for describing documents nor a meta-language in which to create document description languages; in this respect the notion of meta-structuring seems inapplicable. Rather, Microsoft's 'Object Linking and Embedding' architecture is a protocol aimed at permitting structures from foreign applications to be 'linked to' and 'embedded in' a local application. These structures need not be document components, but OLE seems to have been designed with documents specifically in mind, so that foreign components can be displayed and edited in a containing document. Linked or embedded elements can themselves link to or embed other components, thus producing arbitrary hierarchical (composite) structures.

OLE's linking mechanism is also meant to enable interchange in the sense defined above — 'the ability to move a representation of a document from one environment to another' — but it accomplishes this quite differently than SGML or ODA. Whereas SGML and ODA are based on notions of application independent document description, OLE is intentionally application dependent. In OLE, the display and editing of a document element in a new environment requires the presence and operation of that element's source application. Given this application dependent model, it is not surprising that OLE makes no distinction between presentation independent and presentation dependent material.

At the end of the last section, I suggested that the purpose of these standards is to extend the boundary within which reuse can occur. It is my sense that each of the standards is encouraging somewhat different patterns of reuse; this is the result both of the way it has been designed and the forms of practice that have developed around it. Thus, in practice ODA seems to be primarily concerned with the exchange of whole, compound documents; a complete document is moved into a new environment where it can be reused in one of several ways — typically (re)edited and/or printed. OLE by contrast is more oriented toward the reuse of components — toward collection and combination. SGML is being used in both of these ways: various communities (for example, the airline and automotive industries [19] have been writing DTDs to exchange particular types of documents; others

[10] DTDs, as is often pointed out, can be used to represent layout as well as logical structure. The DSSSL standard [16] is being designed explicitly to permit the representation of layout structure.

[11] In the automotive industry, for example, a DTD has been written for service information (J2008 [18]), which is intended to be used to represent automobile engine components and their relations.

are using it as the means to represent reusable document components in document databases [20]. But what is perhaps most striking, and distinctive, is the way these forms of reuse are realized through SGML's meta-structuring capability, which permits communities of users to design a particular document description language (and accompanying applications) to suit their own patterns of document reuse.[12]

5 ISSUES AND FURTHER DIRECTIONS

What lessons can we draw from seeing reuse as central to the handling of documents? I think there are implications at both the micro level and the macro level. By the micro level, I mean the movement of material 'pair-wise' between two documents. By the macro level, I mean the transformation of material across multiple documents over larger stretches of space and time.

At the micro level, the central question is: how do we do a better job of supporting users' needs to move material among documents? This will only become a more pressing question as the diversity of systems, media, genres, and applications increases, as it quite evidently will. The main objective is to give users greater ability, and flexibility, in specifying how document material is to be transformed for use (reuse) for new purposes in new contexts.

Current approaches, as in the interchange standards addressed above, tend to assume that users want to make a 'copy' of their documents — to preserve (all) the structure and editability of the source document. Not only is this impossible in all but the simplest cases, but it may not even be desirable when the user is moving material to be incorporated into other documents for new purposes. A better approach would be to allow users to specify what material they want transformed, how they want it transformed, and for what purposes. But to do this in more general ways, we will need more robust approaches to the theory and practice of document transformation. Steps in this direction can be found in [22] and [16].

At the macro level, there is a need to develop better computational tools for tracking the history of modification of documents and the movement of material among documents. Steps have already been taken in this direction — for example, editors that can display newly introduced material [23,24] and publishing systems that keep track of pieces of documents as they are combined to produce a final publication [23]. But there is clearly further to go, especially when we take into account the heterogeneous nature of the world. It is hard enough to track and transform document material across different computational platforms, document representations and standards; how is one to accomplish this when some of the transformations and references are to or from paper?

Moreover, I suspect that getting a handle (literally and figuratively) on the flow of document material will require developing better accounts of, and better computational support for, versions. Do we yet understand what constitutes a version of a document, when and why people create them, how people refer to them, or how they would like to name or find them? In a complex web of document fragments which are being transformed, combined and recombined, what relationship do document pieces bear to versions? What is the relationship between versions of pieces and versions of whole documents? What kinds of support can we provide if we acknowledge that versions can be incomplete and

[12] It is also worth mentioning Adobe's Acrobat [21], which is attempting to extend the bounds of reuse in a different way. Focused on the exchange of document images in support of reading, Acrobat provides a scheme that is intentionally presentation dependent but device independent.

overlapping, or if we entertain the possibility that the distinction between a new document and a version of a document is context and use dependent, and therefore fluid.

One place to look for help in understanding such issues, as a prelude to developing computational support, is the field of textual criticism or scholarship [25,26], which has been concerned with studying and managing the flow of biblical works and secular works of literature. Beginning at least as far back as the Library at Alexandria (founded c. 323 BCE), scholars realized that two manuscripts of the same work might have textual differences, in part because errors were inevitably introduced as manuscripts were copied. They began to develop techniques for identifying and correcting errors — for example, techniques for reasoning about the genealogical relations among manuscripts and for assessing the internal coherence of texts. Over time, textual scholars have served two primary functions: they have acted as custodians of literature, attempting to ensure the integrity of our literary heritage by correcting and commenting on editions from Plato to Joyce; and they have developed a body of theory that allows them to represent, discuss, and evaluate the relationships among documents, including their history of modification and movement. If we want to understand the complexities of document transformation and reuse, this would seem to be a good place to look.

It would also be valuable to look at the work that has been done on reuse in the programming language community — including change management and version control — to see how much of this is relevant to document systems. It is intriguing to notice that the programming language community, a community a mere few decades old, has seized upon reuse as a central topic, while the document systems community hasn't yet done so — certainly not to the same extent — despite the fact that document reuse is as old as documents themselves.

ACKNOWLEDGEMENTS

I am grateful to many people for generously contributing their time and knowledge to this study. I would particularly like to thank Annette Adler, Jean-Marie de La Beaujardière, Dan Brotsky, Allen Brown, Heather Brown, Mary Dalrymple, Niklas Damiris, David Elliott, Alice Wilder Hall, Al Howe, Gregor Kiczales, Paul Kennedy, John Lamping, Mike MacKay, Larry Masinter, Jill Miller, Makoto Murata, Alan Perey, Sally Peters, Harold Shinsato, Lucy Suchman, and Randy Trigg.

REFERENCES

1. G. Kubler, *The Shape of Time*, Yale University Press, New Haven and London, 1975.
2. *Software Reusability*, eds., T.J. Biggerstaff and A.J. Perlis, volume 1, ACM Press, New York, 1989.
3. G. Booch, *Object Oriented Design with Applications*, Benjamin Cummings, New York, 1991.
4. R. Furuta, V. Quint, and J. André, 'Interactively editing structured documents', *Electronic Publishing*, **1**(1), 19–44, (April 1988).
5. *International Standard ISO 8879: Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML)*, ed., International Organization of Standardization, Geneva/New York, First edition, 1986.
6. *International Standard ISO 8613: Information Processing – Text and Office Systems – Office Document Architecture (ODA) and Interchange Format*, ed., International Organization of Standardization, Geneva/New York, 1986.
7. Microsoft Corporation, *Object Linking and Embedding: OLE 2.0 Design Specification*, 1993.

8. T. Berners-Lee, R. Cailliau, J-F. Groff, and B. Pollermann, ‘World-Wide Web: The information universe’, *Electronic Networking: Research, Applications and Policy*, **1**(2), (Spring 1992).
9. G. Nunberg, ‘The places of books in the age of electronic reproduction’, *Representations*, **42**, 13–37, (Spring 1993).
10. H. Abelson and G. Jay Sussman, *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge, 1985.
11. J. Yates, *Control Through Communication*, The Johns Hopkins University Press, Baltimore and London, 1989.
12. B. Latour, ‘Visualization and cognition: Thinking with eyes and hands’, *Knowledge and Society: Studies in the Sociology of Culture Past and Present*, **6**, (1986).
13. D.M. Levy and K.R. Olson, ‘Types, tokens and templates’, Technical Report CSLI–92–169, Center for the Study of Language and Information, (June 1992).
14. D. Brotsky, D.M. Levy, and D. Rabin, ‘Yeti: A graphics editor based on ‘collect-and-stitch”, in *EP92 Text processing and document manipulation*, eds., C. Vanoirbeek and G. Corey, pp. 211–221, Cambridge, UK, (April 1992). Cambridge University Press.
15. V.L. O’Day and A. Paepcke, ‘Understanding information needs in technical work settings’, Technical Report HPL-92-123, Hewlett-Packard Laboratories, (1992).
16. *International Standard ISO/DIS 10179: Information Technology – Text and Office Systems – Document Style Semantics and Specification Language (DSSSL)*, 1991.
17. J.M. Smith, *SGML and Related Standards*, Ellis Horwood, New York, 1992.
18. *SAE J2008*, Society of Automotive Engineers, 1993.
19. M. Walter, ‘Update on the ‘quiet revolution’: Report from SGML ’92’, *The Seybold Report on Publishing Systems*, **22**(7), 13–17, (December 1992).
20. G. Alexander and M. Walter, ‘A fresh look at SGML: The conventional wisdom changes’, *The Seybold Report on Publishing Systems*, **20**(7), 3–16, (December 1990).
21. P. Ames, *Beyond Paper: The Official Guide to Adobe Acrobat*, Adobe Press, Mountain View, CA, 1993.
22. A. Brown, S. Mantha, and T. Wakayama, ‘The declarative semantics of document processing’, in *Proceedings of the First Conference on Principles of Document Processing*, Washington DC, (October 1992).
23. Frame Technology Corporation, San Jose, CA, *Framemaker Version 3.1*, 1992.
24. Xerox Corporation, Palo Alto, CA, *Xerox GobalView Network Computing Environment*, 1990.
25. W.P. Williams and C.S. Abbott, *An Introduction to Bibliographical and Textual Studies*, Modern Language Association of America, New York, 1992.
26. *Introduction to Scholarship in Modern Languages and Literatures*, ed., J. Gibaldi, Modern Language Association of America, New York, 1992.