Multivalent Documents:
Inducing Structure and Behaviors in Online Digital Documents

Thomas A. Phelps and Robert Wilensky
University of California, Berkeley
{phelps, wilensky}@CS.Berkeley.EDU

Abstract

Digital documents have advanced only slightly from their non-digital ancestors, incorporating a few such innovations as hypertext links and full text searching. Even the much heralded advent of multimedia, just as embedded images and video clips, are rather straightforward digital analogs of books and film. We believe that computer technology makes possible far more interesting and complex digital documents, and propose a new general paradigm that regards complex documents as "multivalent documents" comprising multiple "layers" of distinct but intimately related content. Small, dynamically-loaded program objects, or "behaviors", activate the content and work in concert with each other and layers of content to support arbitrarily specialized document types. Behaviors bind together the disparate pieces of a multivalent document to present the user with a single unified conceptual document.

Examples of the diverse functionality in multivalent documents include: "OCR select and paste", where the user describes a geometric region on the scanned image of a printed page and the corresponding text characters are copied out; video subtitling, which aligns a video clip with the script and language translations so that on the one hand the playing video can be presented simultaneously in multiple languages, and on the other hand the video can be searched with text-based techniques; geographic information system (GIS) visualizations that compose several types of data from multiple datasets; and distributed user annotations that augment and may transform the content of the conceptual document.

In general, a document management infrastructure built around a multivalent perspective can provide an extensible, networked system that supports incremental addition of content, incremental addition of interaction with the user and with other components, reuse of content across behaviors, reuse of behaviors across types of documents, and efficient use of network bandwidth. Multivalent documents exploit digital technology to enable new, more sophisticated document interaction.

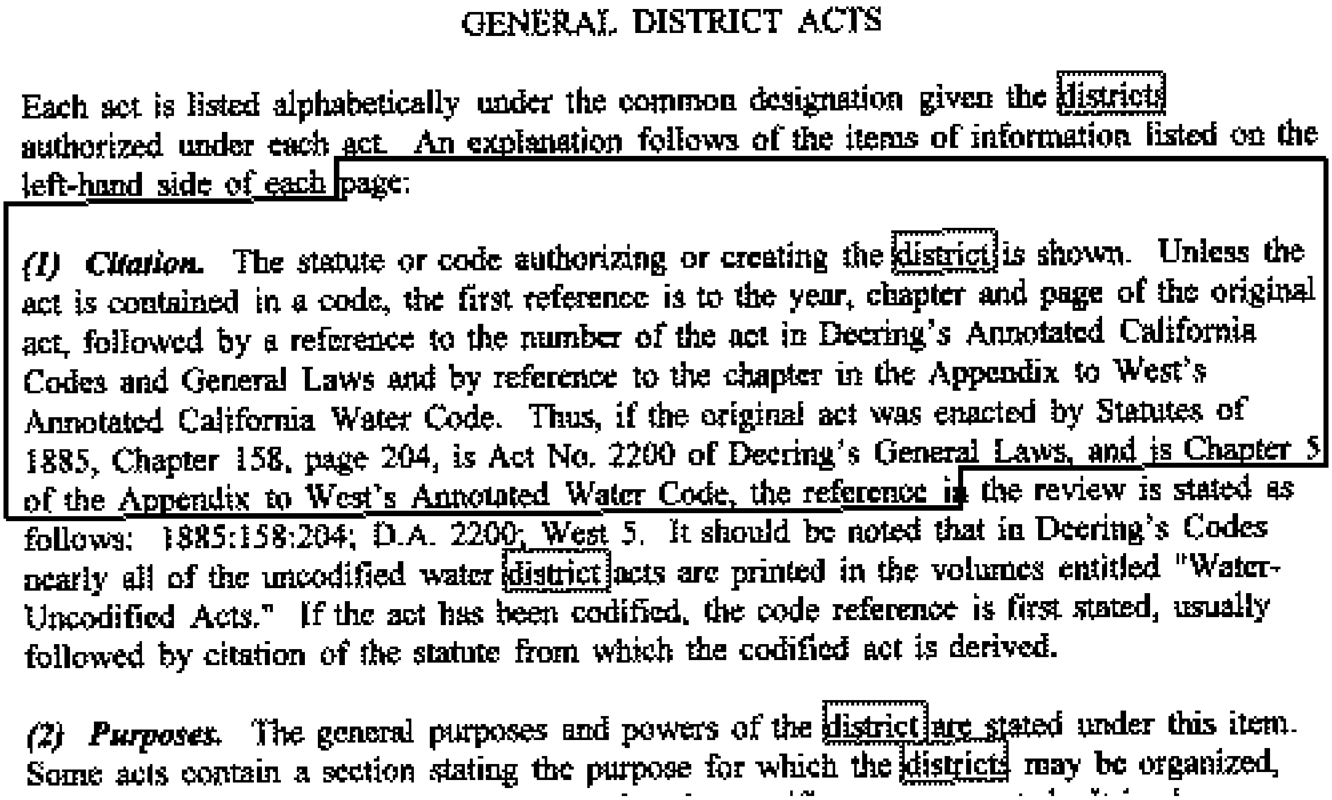
The work reported here was supported in part by National Science Foundation grant IRI-9411334 as part of the NSF/NASA/ARPA Digital Library Initiative.

1. Introduction

"Multivalent documents" are a new paradigm for the organization of complex digital document content and functionality. The multivalent view of a document rebukes monolithic data formats that attempt to encompass all possible content types of the document in a single, complicated specification, and provide for interaction with them with concomitant complicated editors, often of limited extensibility. In contrast, the multivalent approach "slices" a document into layers of homogeneous content, to which additional layers may be added at a later time at equal status. Functionality is provided by flexible, dynamically-loaded program-objects, called "behaviors", that manipulate the content. Behaviors may communicate with multiple layers and other behaviors, and new behaviors can be added subsequently at equal status.

In conceiving of documents as interacting layers of content and functional behaviors that unite to present a single conceptual document, our paradigm introduces important new capabilities and benefits. By representing a document as multiple components, others besides the original document's author can add additional content and behavior at a later time, even introducing new technology that was unknown during the preparation of the initial materials. By keeping the pieces of content simple, the behaviors that manipulate them are relatively straightforward to define; the components are also more easily reusable. It is also more likely to survive evolutions in technology: 30 year old ASCII files are easily readable today, whereas 15 year old WordStar files are not. By storing the content in distinct pieces, only those pieces necessary to support the requested interaction need be shipped over the network, thus conserving bandwidth. Even if high bandwidth connections to desktop machines are eventually commonplace, this advantage will be valuable for portable wireless computing devices, where

Published in *Proceedings of the 29th Hawaii International Conference on System Sciences /* Maui, Hawaii / January 3-6, 1996 (Best Paper Award, Digital Documents Track)



GENERAL DISTRICT ACTS

Each act is listed alphabetically under the common designation given the districts authorized under each act. An explanation follows of the items of information listed on the left-hand side of each page:

(1) Citation. The statute or code authorizing or creating the district is shown. Unless the act is contained in a code, the first reference is to the year, chapter and page of the original act, followed by a reference to the number of the act in Deering's Annotated California Codes and General Laws and by reference to the chapter in the Appendix to West's Annotated California Water Code. Thus, if the original act was enacted by Statutes of 1885, Chapter 158, page 204, is Act No. 2200 of Deering's General Laws, and is Chapter 5 of the Appendix to West's Annotated Water Code, the reference in the review is stated as follows: 1885:158:204; D.A. 2200; West 5. It should be noted that in Deering's Codes nearly all of the uncodified water district acts are printed in the volumes entitled "Water-Uncodified Acts." If the act has been codified, the code reference is first stated, usually followed by citation of the statute from which the codified act is derived.

(2) Purposes. The general purposes and powers of the district are stated under this item. Some acts contain a section stating the purpose for which the districts may be organized,

Figure 1: This is a scanned image of a printed page, which is the initial digital capture format for most documents in Berkeley's Digital Library Project. In many ways, the user can work with the image of the page as if it were a word processing document. The prototype, which is implemented in Java, refers to a layer of geometrically-positioned characters to provide "select and paste" functionality and incremental searching in a WWW client.

bandwidth and memory will be a concern for the foreseeable future.

As a very simple example of a multivalent document, consider documents that begin their lives as scanned page images. In the UC Berkeley Digital Library Project [27, 26], we have 90,000 pages of such documents from the California Department of Water Resources. It is straightforward to run OCR on such a collection and hence make the documents available for access by full-text search. Indeed, we have done this, and they are available at http://elib.cs.berkeley.edu. However, pages images are just inert pictures of text, with little functionality. In our web interface, users may access the OCR equivalent of a page and do searches or selects from this source. However, it is awkward to move back and forth between images and OCR.

However, consider now our prototype multivalent document implementation. Figure 1 displays the image a scanned page (with telltale page skew). Underlying this image are two layers: the corresponding ASCII characters and their geometric positions on the image (both generated by automatically). A program behavior allows the user to select regions of the image and subsequently paste them into another document via the X selection, and to incrementally search in the image as if it were a document in a word processor. Although this is a relatively simple image, the same techniques would work for text labels embedded in maps and for handwritten text.

Figure 2 shows before and after images of a table. Of course, the author must choose a particular ordering before committing the table to print. In a multivalent framework, the description of the table's rows, columns, and cell contents can be supplied as a semantic layer. In this way, we can manipulate the table later. The prototype uses advanced document analysis developed in the project that derives the information. In response to, say, a mouse click on a given column heading, the prototype can sort the table by that column and display the results by rearranging the pixels in the image. One could consider adding some spreadsheet functionality to sum columns, for example, or even placing the table in a spreadsheet behavior.

Although a multivalent approach can be taken for simple documents, it is most valuable when applied to complex

Facility	Gross Capacity (Acre-feet)	Surface Area (Acres)	Shoreline (Miles)
Antelope Lake	22,600	930	15
Frenchman Lake	55,500	1,580	21
Lake Davis	84,400	4,030	32
Lake Oroville	3,537,600	15,800	167
Thermalito Forebay	11,700	630	10

Figure 2a: This is a table image from a scanned document. A multivalent framework may associate structural information such as a description of the rows and columns of the table, as well as the ASCII text in each cell.

Facility	Gross Capacity (Acre-feet)	Surface Area (Acres)	Shoreline (Miles)
Lake Oroville	3,537,600	15,800	167
Lake Davis	84,400	4,030	32
Frenchman Lake	55,500	1,580	21
Antelope Lake	22,600	930	15
Thermalito Forebay	11,700	630	10

Figure 2b: With the content available in a multivalent document decomposition, a table behavior can respond to, say, a mouse click on the "Surface Area" heading by sorting the table by that column and displaying the results by rearranging the pixels of the image. (This is a mockup of functionality available in the prototype.)

digital documents. By "complex document" we mean documents that contain a large number of varied types of information and therefore could benefit from sophisticated interactions with users. Examples of possible multivalent documents include subtitled video, which aligns a video clip with the script and language translations so that on the one hand the playing video can be presented simultaneously in multiple languages, and on the other hand the video can be searched with text-based techniques; geographic information system (GIS) visualizations that compose several types of data from multiple datasets; and distributed user annotations that augment and may transform the content of the conceptual document. Consider also program browsing and medical imaging. Fine-grained program analysis for large source trees can consume many hours and involve many types of analysis, e.g., interprocedure register allocation and aliasing detection. The resulting collection of many kinds of diverse but intimately related information about the program sources lends itself to a multivalent perspective, especially if new analyses are added on a regular basis. In medical imaging applications, various images of the human body are aligned with each other spatially and chronologically, and are associated with meta information including instrument settings for each image, patient's name, and medical history. Although narrowly tailored software could written to support any given niche, the general framework provided by multivalent documents leverages the work done in other areas, while fully supporting specialization to particular application domains.

Because the concept of multivalent document decomposition is a subtle abstraction, the paper next presents an extended concrete example illustrating a multivalent treatment of an ancient complex document, the Talmud. In this context we then detail the new levels functionality afforded in the multivalent paradigm. The looser coupling of document components introduces concerns that are examined thereafter. We conclude with a look at related work and statement of the current status of our work.

2. Example multivalent document: the Talmud

To motivate the need for a more sophisticated model for digital documents, consider the Talmud. The Talmud comprises a transcription of the early oral law governing Jewish life and the commentary upon it. As detailed below, the Talmud is a complex document that could benefit from treatment as a digital, multivalent document—a treatment that previous document models are unable to support in all its richness.

2.1 Complex, varied content

Figure 3 depicts a page from the Talmud. Each page is highly structured.[1] The transcription of the oral tradition, the mishna, is rationed to only a small part of the page; this block is located at the upper center (labeled 'A' in the diagram at right). The large majority of the page is given over to commentaries. The most ancient commentary, the gemara (labeled 'B'), follows immediately below the mishna. Surrounding the mishna and gemara are commentaries by other Rabbis ('C' and 'D'). Around the edges of the page can be found cross references to other passages in the Talmud ('a'), cross references to medieval

1. This description is taken from [13].

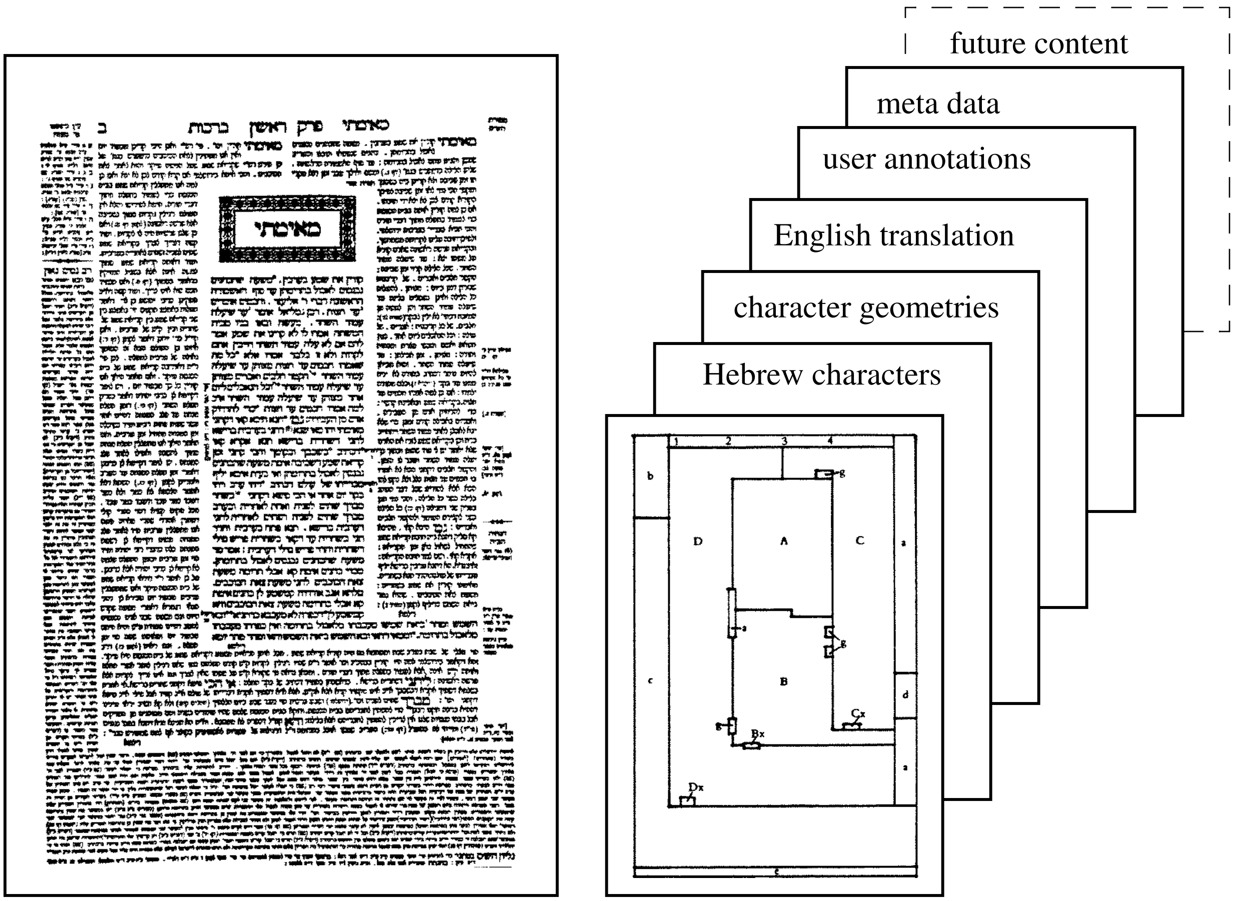


Figure 3: A multivalent perspective regards complex documents, such as the Talmud, as comprising multiple layers of content. Dynamically-loaded behaviors communicate with layers and other behaviors to provide high functionality that can be reused across varied document types and arbitrarily specialized by the user.

codes of Jewish law (‘b’), a textual emendation (‘d’), a key to quotations from the Bible (‘g’), and identifiers such as the page number and chapter title (‘1’ to ‘4’).

As a digital artifact, the Talmud should foremost consist of a scanned image. The layout of various blocks of text is historically significant for the Talmudic scholar, but difficult or at least time consuming to reproduce with text layout software, and therefore the image is essential in order to maintain fidelity with the original. In other documents, too, the primary representation should be the scanned image. In illuminated manuscripts, the pictorial “decorations” are for some purposes more important than the text. Some scholars wish to examine every worm hole, for the text eaten out may have been incorrectly interpolated by the transcriber, and an alternative interpolation may invite a new interpretation of the passage. For handwritten manuscripts, such as Mark Twain’s original draft of Huckleberry Finn with its copious editing markup, obviously there is no complete (digital) substitute for the image. Even with contemporary documents, optical character recognition (OCR) and structural analysis software are not entirely accurate, and the page image remains a necessary fallback.

Other digital information is closely related to the page itself. Clearly the structural map identifying geometric regions is essential to understanding, and the digital version should know of the relationship between page region and its semantic content. For more effective manipulation in text editors, the text of each block should be available as characters. Common OCR software can produce this ASCII (or, for Hebrew, Unicode) character text, but some OCR packages can in addition produce maps of character-to-page position. The various cross references on the page take the obvious digital analog of hyperlinks. Finally, various “meta data” such as the page number and links to the next and previous pages are closely connected with the page.

Because the digital version is stored as bits and not limited by physical considerations, we can augment the page with other related information. Most commentaries are hundreds of years old, for only a limited number fit in the dimensions of printed page; obviously in the digital version an arbitrary number of additional commentaries can be added. One might even humbly attach personal annotations to the work. For those learning Hebrew, an audio pronunciation (whether from a text-to-speech system or from canned digital audio files) would be appreciated. For those who do not read Hebrew, a vernacular translation—or translations, into English, French, Japanese, et cetera—are indispensable. Even within a vernacular, academics would appreciate both a literal translation as well as a more fluidly readable version.

2.2 Digital content: fragmented, distributed, heterogeneous

The creative reader can undoubtedly think of more information that could be usefully associated with the Talmud. Indeed, this collection can be made arbitrarily complex. Thus, it would be useful to leave the collection as an extensible set, each piece capturing a *fragment* of the notional document. Similarly, mandating a fixed set of required pieces could impose an onerous burden on the digitizers, who, depending on their objectives, may not use that kind of content and therefore would not wish to invest the work to produce it. As a side benefit of keeping content pieces simple, one maintains the widest choice of tools that can be used to prepare them.

Different types of content have different natural homes, leading to geographically *distributed* collections. Assume that the single, worldwide authoritative version of the scanned page, its structural layout map, the translation into Hebrew characters, and the mapping of characters to geometric positions in the image are all maintained at the digital repository of a scholarly or religious institution. Vernacular translations, however, are more likely to be found in repositories of nations that speak that language. Finally, personal annotations are kept locally.

Currently-identified pieces of Talmud content come in a variety of *heterogeneous* data types. Among them are the page image, textual transcriptions and language translations, hyperlinks, audio clips, a bidirectional structural mapping between symbolic labels and the image, a bidirectional mapping between characters and their position on the image, various meta data, user annotations, and active program code that will be explained in the next section. This implies that document models like SGML [11] or HyTime [17] that merely coordinate a subset of these types will not suffice. As Furuta points out [10], describing the structure and interrelationships of the wide range of objects in the digital library will require powerful new metaphors.

2.3 Multivalent document approach

In short, a comprehensive digital Talmud is composed of a variety of fragmented, geographically distributed, and heterogeneous—but intimately related—pieces of content. In multivalent document terminology, each of piece of content is called a *semantic layer*. The layer metaphor is meant to imply that pieces may not have a visual representation, that only bound together do they form the notional document, in contrast to the more-independent nodes in a hypertext or a compound document.

In particular, the user should interact with the collection of layers as single conceptual document, as a unified and functional whole. Returning to our Talmud example, a user will want to interact with the image representation of the page for reasons described above, while at the same time manipulating it as if it were an ordinary text document. For example, the scholar studying a multivalent version of the Talmud will be able to select a region on the image and grab the corresponding character representation to paste into another document. This “OCR select and paste” takes coordinates from the image, routes them through the position-to-character mapping, and finally extracts the Hebrew characters from the OCR stream. It would be possible to define many different kinds of such actions, such as extracting the corresponding aligned text (at sentence or paragraph boundaries) from an alternative language layer, say, English or Japanese, or automatically augmenting the text with its provenance, including the edition of the Talmud, page number, chapter title and so forth.

The layers of a multivalent document can be ordered into a composite filter, much like UNIX pipes. This can be useful in correcting the results of the optical character recognition software. Good OCR software is approximately 98% correct. For extremely important documents care would be taken to correct this output by hand. But with the cost of OCR correction running two to five dollars per page [2], it would not be cost effective to invest such effort for the great mass of scanned documents. Ideally, consumers of the documents should be able to vote on the importance of a correct textual transcription. In a multivalent representation they can. For the personally important document, an individual can locally override the OCR extraction function with one that generates the corrected version, either by entirely replacing the content or by filtering the incorrect content through a program that corrects the few incorrect characters on the fly. This filter can be submitted to the official repository of the document for verification and subsequent incorporation into its master version.

This ability to freely "wire" connections between layers is useful in other types of documents as well. For documents with mathematical equations, the selected region in the image could be delivered as the TeX mathematical markup, PostScript layout, or as executable code, whether in Lisp, C, Mathematica/Maple, or another computer language. Bibliographic entries could similarly be delivered in BibTeX or EndNote database format. Many of these alternative representations would have to be composed by hand and attached to the document as another semantic layer. For those that can be derived by machine, one can store an abstract representation of the information and request the desired concrete version from the associated program filter.

Although no predetermined set of content pieces are required, the structural representation will prove to be an important component. It will allow other layers to communicate at a higher level of abstraction. For instance, rather than associating each cross reference in the Talmud with an explicit hyperlink, a programmatic behavior could intercept, say, mouse clicks on these regions, as determined by consulting the page schematic map, and for them extract the OCR text to identify the destination point of the link. In contemporary scholarly documents, this facility could connect bibliographic reference with their full entries without littering the document with hardcoded hyperlinks. Such hyperlinks are good for the sole purpose for which they were created, whereas time invested in creating a semantic structural layer can be reused for other purposes.

With the ability to arbitrarily transform the document, user annotations can be much more interesting than usual. Some existing annotation systems force the user take a "snapshot" of the screen, upon which arbitrary graphical figures can be drawn, while more advanced systems compose textual comments from multiple users to points in a text document. Extending these capabilities, multivalent annotations can arbitrarily augment the content or interaction from remotely accessed documents with local semantic content layers and program code. As regards the Talmud, one is instructed to read each commentary as if it is the only one on the page. This can be realized with a query to the structural map to identify the desired geometric region on the image and a call to utility program tools to automatically elide all other commentaries.

Corresponding to the fact that multivalent documents are segmented into layers, the key entity of the architecture is an object-oriented *behavior*. Behaviors encapsulate the interaction with information of a specific type. In operation a behavior can directly manipulate its own data, send messages to other behaviors to request information or action, and communicate with the user through a graphical user interface toolkit. Behaviors for a particular document are dynamically loaded over the network on demand. Behaviors are typed like classes and methods in an objected-oriented programming language, and a dynamically constructed type graph governs interaction between behaviors. The architecture of multivalent documents is considered at length elsewhere [20].

3. Issues

3.1 Misalignment of Layers

The intimately related but decoupled nature of multivalent documents introduces a concern: a change to document could cause other layers to become misaligned. Although difficult to fully combat in the most general case, an array of strategies can mitigate the practical impact of this possibility. In some cases, a layer of content does not change: chances are that the PostScript of a Berkeley technical report from 1988 will never be updated. In the case of a scanned page of text, its structure and the textual transcription, changes in the scanned page can be fully propagated through to the others by completely recomputing them. For other cases, the layer can be versioned at the server. Attachments to that layer also record its version number, and clients can then either request the specific version, or the new version along with a list of changes from the requested version so that the layers can be realigned. An ad hoc association between layers on volatile and uncooperative non-versioning servers can record context in order to increase chances reorienting itself after a mutation. Finally, some layers are unaffected by changes to an associated layer. Anything referring to the mishna on a page or a position relative to the mishna, for instance, is related on an abstract level, and when the geometric location the mishna changes, the relationship on the symbolic level is undisturbed—in fact it has the desirable property that it is transparently updated to operate properly with the new information.

3.2 User Interface Coherence

Another problem is how to present a comprehensible user interface in the face of competing behaviors from varied sources. In a sense, learning the space of interaction with a new document specialization is little different from learning a new application program, and the same techniques used in graphical user interfaces could be applied. That is, functionality common to most multivalent documents, searching perhaps, would be standardized, and other functions could be placed into menus of commands.

Behaviors that introduce interaction modes would first be required to register the user interaction gestures in which they are interested, and only modes with disjoint sets of gestures would be allowed to coexist; enabling one automatically disables conflicting behaviors. Alternatively, a Toolglass and Magic Lens [5] style of interaction would seem to complement a multivalent document approach perfectly. Toolglasses carry collections of Magic Lenses, and the user invokes a tool on a document by placing a lens over the affected area and operating through the lens. Multivalent documents would provide the content and a set of tools; Toolglasses could organize them and a Magic Lenses would provide an excellent interaction paradigm.

4. Related Work

4.1 Compound Documents

A comparison with compound document systems like OpenDoc [15] and Microsoft's OLE [7] highlights the distinctive decomposition of the multivalent approach. In analogy to operating systems, the relationship between compound documents and multivalent documents parallels that between processes and threads. That is, compound documents organize multiple editors in the same document but at a much coarser level of granularity. In the compound approach, a geometric region of the document can be devoted to a particular editor, which can introduce new content and functionality into the container document. Although OpenDoc allows an editor to communicate with other editors to access their data and request actions, both OpenDoc and OLE fundamentally view the document as a set of very loosely coupled data types spatially arranged. Content is coupled with functionality and screen real estate, and each functional unit operates primarily with a single, complex data type. In the multivalent approach, complexity is built up from simple content and most interesting behaviors operate on multiple layers of content and with other behaviors.

4.2 Active Documents

Many systems activate the digital page with program script control, including Active Tioga Documents [24], Embedded Buttons [4] and Computational E-mail [6] to list but a few. In general these systems connect a script to a specific region of the document; in these regions the document is active, and elsewhere the document retains its ordinary properties. In the multivalent approach, active regions vary according to what behaviors are active, and the content of that region varies according to what semantic layers are available. Others systems, such as Henry [22] and Firefly [8], coordinate external viewers or editors through narrow interapplication interfaces to induce on the heterogeneous content composite hyperlinks and temporal behavior. In the multivalent approach, document behavior is programmed in the native object system, but access to advanced tools is maintained: Dedicated, sophisticated analyzers and editors can be used at document preparation time to compute or compose information, and the results are stored for manipulation at runtime by less heavyweight code.

4.3 Multiple Representations

Many multivalent documents, including the Talmud, use multiple representations in the service of one or few presentations. This inverts the Smalltalk model-view-controller paradigm [14] which maintains a single shared model in support of multiple views.

RightPages [23] models scanned journal article pages "as three planes of information: the image, OCR text, and page layout". As well, it maintains the representations distinctly (on disk) and as a (C++) object internally. RightPages is an example of a multivalent perspective but in the form of a custom system for a fixed and small number of layers.

In order to edit scanned images of text, Image Emacs [2] constructs on the fly a map of connected components that roughly corresponds to characters—sidestepping the time and complexity of optical character recognition—and operates through this layer in reformatting "characters" as requested by editing commands. Multivalent documents share with Image Emacs the characteristic that both manipulate the document through layers of partial information, without demanding the "native" or source representation. Although a multivalent behavior could also identify the connected components on the fly also, its bias is toward caching computationally intensive analysis for use by portable lightweight code.

4.4 Annotations

Annotation systems maintain a distinction between a "base" document and the additional annotation material, a distinction that is generalized in a multivalent document. It is useful to be able to store annotations separate from the base document for several reasons: the source of the document may not be mutable (e.g., it comes from a read-only server or a CD-ROM), the data format may not be amenable to storing annotation information (a digitized sound clip meant to be streamed as raw data through a speaker driver may not have accommodations for annotations), and it should be possible to annotate shared documents without viewing annotations made by others. TkMan [19], a graphical browser for UNIX manual pages, allows the user to mark regions as if with a yellow highlighter marker. Highlighting information is stored in a user's personal database and merged with manual pages as they are shown; when a manual page changes, as determined from its file modification date, the user is asked whether or not to throw out the highlights wholesale, but no attempt is made to adjust the highlights to the modified document. Adobe's [25] supports textual attachments tied to a geometric position on the page, and annotations from several users can be combined into a composite page, but no attempt is made to adjust to a modified base document. More advanced annotation systems like CoNote [9] and ComMentor [21] tie textual annotations to particular locations in the text stream, compose annotations from multiple users, and the latter system is robust against changes to the original document.

5. Current and Future Work

As part of research effort of the University of California, Berkeley's Digital Library Project [27, 26], we are initially applying multivalent document decomposition to California Department of Water Resources technical reports. Like the Talmud example, these reports consist of scanned page images that are analyzed for both structure and characters transcription. As more content such as tables and maps can be extracted intelligently, it will be added as additional layers. The other current focus of research is a distributed annotations system. Although no different from other behaviors at an abstract level, an annotations system needs a friendly user interface for such common annotation styles as adding a textual comment or hyperlink, graphically embellishing an image, and eliding a region of content.

With the goal to make the digitized collection widely available, we deliver materials over the World Wide Web [3] as much as possible. Although Mosaic [1] and Netscape [16] are sufficient for simple viewing of simple media like images and text, full multivalent interaction requires a browser that can execute the program-objects, which in our case are written in Sun's C++-like Java language [12]. The basic document model of the HotJava browser mimics a compound document in that rectangular regions of the document controlled by dynamically-loaded program code; we are taking this browser as a base and modifying its source code to support the multivalent model. All document content and program code is stored in the object-relational database Illustra, the commercial version of Postgres.

Figures 1 and 2 depict functionality currently available in a prototype implementation. In the future, the Project's rich collection of geographic materials will take advantage of the multivalent framework articulated above in the flexible delivery over the network of functionality found in dedicated geographic information systems (GIS). Currently, authoring can be done at two levels: the expert Java level and the easier but more restricted user annotation level. Future work will provide intermediate stages between these extremes. One possibility is a rules systems that the user could program to invoke a program behavior (as a black box) in response to patterns identified in document content. Also under consideration is a scripting language level that would expose primitives implemented in Java to a very high level glue language like Tcl [18].

6. Conclusions

We have described a new model for digital documents called multivalent documents. It takes a fine-grained, object-oriented perspective on digital documents that is well suited to a networked world of rapidly evolving document technology. This paradigm views digital documents, especially complex ones with varied content and styles of interaction, as comprising homogeneous layers of semantic content. Content is activated by program behaviors that are dynamically loaded from geographically distributed locations. Behaviors specialize interaction for classes of documents, and unify semantic layers to present the user with a single conceptual document. Both content and functionality can be augmented at a later time in a way that places new material at equal status with existing material. Although a custom system could be written for any given document niche, a multivalent framework leverages work done for other multivalent documents. A multivalent perspective is useful for a wide variety of complex documents, including the Talmud. We believe that a multivalent document approach to complex digital documents enables new, sophisticated functionality and content for complex digital documents.

Acknowledgments

7. References

[1] Marc Andreessen. NCSA Mosaic technical summary, May 1993.

[2] Steven C. Bagley and Gary E. Kopec. Editing images of text. *Communications of the Association for Computing Machinery*, pages 63–72, December 1994.

[3] T.J. Berners-Lee, R. Cailliau, J-F Groff, and B. Pollermann. World-Wide Web: The information universe. In *Electronic Networking: Research, Applications and Policy*, volume 2, pages 52–58. Meckler Publishing, Westport, CT, USA, 1992.

[4] Eric A. Bier. EmbeddedButtons: Documents as user interfaces. In *Proceedings of the ACM Symposium on*

User Interface Software and Technology, pages 45–53, November 1991.

[5] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and Magic Lenses: The see-through interface. In *Proceedings of SIGGRAPH '93*, pages 73–80, August 1993.

[6] Nathaniel S. Borenstein. Computational mail as network infrastructure for computer-supported cooperative work. In *Proceedings of Computer Supported Cooperative Work (CSCW)*, pages 67–74, November 1992.

[7] Kraig Brockschmidt. *Inside OLE 2*. Microsoft Press, Redmond, WA, 1995.

[8] M. Cecelia Buchanan and Polle T. Zellweger. Specifying temporal behavior in hypermedia documents. In *Proceedings of the ACM Conference on Hypertext (ECHT '92)*, pages 262–271, November 30–December 4 1992.

[9] Jim Davis. CoNote. http://dri.cornell.edu/pub/davis/annotation.html.

[10] Richard Furuta. Defining and using structure in digital documents. In *Proceedings of the First Annual Conference on the Theory and Practice of Digital Libraries*, June 1994.

[11] Charles F. Goldfarb. *The SGML Handbook*. Clarendon Press, 1990.

[12] James Gosling and Henry McGilton. The Java language environment: A white paper, 1995.

[13] Barry W. Holtz, Ed. *Back to the Sources: Reading the Classic Jewish Texts*. Summit Books, New York, 1984.

[14] Glenn E. Krasnet and Stephen T. Pope. A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, pages 26–49, August/September 1988.

[15] Chris Nelson. OpenDoc and its architecture. *The X Resource*, 1(13):107–126, 1995.

[16] Netscape Communications Corporation. Netscape. Commercial Software, 1994.

[17] Steven R. Newcomb, Neill A. Kipp, and Victoria T. Newcomb. The hytime hypermedia/time-based document structuring language. *Communications of the Association for Computing Machinery*, 34(11):67–83, November 1991.

[18] John Ousterhout. Tcl: An embeddable command language. In *1990 Winter USENIX Conference Proceedings*, 1990.

[19] Thomas A. Phelps. TkMan: A man born again. *The X Resource*, 1(10):33–46, 1994.

[20] Thomas A. Phelps and Robert Wilensky. Multivalent documents: architecture and applications. Submitted to *Proceedings of the First ACM International Conference on Digital Libraries*. Association for Computing Machinery, March 1996.

[21] Martin Roscheisen, Christian Mogensen, and Terry Winograd. Interaction design for shared World-Wide Web annotations. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI' 95)*. The Association for Computing Machinery, May 1995.

[22] Mario J. Silva. Active documentation for VLSI design. Technical Report UCB/CSD 94-843, University of California, Berkeley, 1994.

[23] Guy A. Story, Lawrence O'Gorman, David Fox, Louise Levy Schaper, and H.V. Jagadish. The RightPages image-based electronic library for alerting and browsing. *IEEE Computer*, pages 17–26, September 1992.

[24] Douglas B. Terry and Donald G. Baker. Active tioga documents. Technical Report CSL-90-6, Xerox Corporation, Palo Alto Research Center, June 1990.

[25] Mark Walter. Acrobat 2.0: Adobe moves up market, beyond ad hoc document delivery. *Seybold Report on Desktop Publishing*, 9(1):3–10, 1994.

[26] Robert Wilensky. Toward work-centered digital information services. Submitted to *IEEE Computer Special Issue on Digital Libraries*, 1995.

[27] Robert Wilensky. UC Berkeley's digital library project. *Communications of the Association for Computing Machinery*, 38(4):60, April 1995.