we can maintain the same document base in two formats simultaneously, allowing *NeXTSTEP* and *OpenStep* users to create an view the complete documents, and others to access the bulk of the information on the Web.

4 CONCLUSIONS

A Demonstration highlights several lessons learned from the eText Project to date. In the search for a system appropriate for producing interactive textbooks, we designed and built the eText Engine. In designing the educational content of a hypermedia textbook, we saw how support for navigation, interactivity, and open publishing are equally important. We have attempted to balance these three concerns in our custom systems development, use of HTML and the World-Wide-Web, and in the architecture of the Engine.

5 ACKNOWLEDGEMENTS

Special thanks go to Adam Rifkin, for many productive discussions, and to other members of the eText project group: Paul Ainsworth, Svetlana Kryukova, and Rajit Manohar. Last, but not least, thanks to K. Mani Chandy for establishing, leading, and supporting the eText Project.

This research is sponsored in part by the Air Force Office of Scientific Research grant AFOSR-91-0070 and CRPC support for education and parallel scientific applications under cooperative agreement CCR-9120008.

References

[1] Paul Ainsworth and Svetlana Kryukova. A multimedia interactive environment using program archetypes: Divide-and-conquer. Technical Report Caltech-CS-TR-93-36, Computer Science Department, California Institute of Technology, 1993.

[2] T.J. et.al Berners-Lee. The world wide web initiative. In *Proceedings of INET'93*. CERN, 1993.

[3] Simson Garfinkel and Mike Mahoney. *NeXTSTEP Programmin: Step One*. TELOS, a division of Springer-Verlag, 1993.

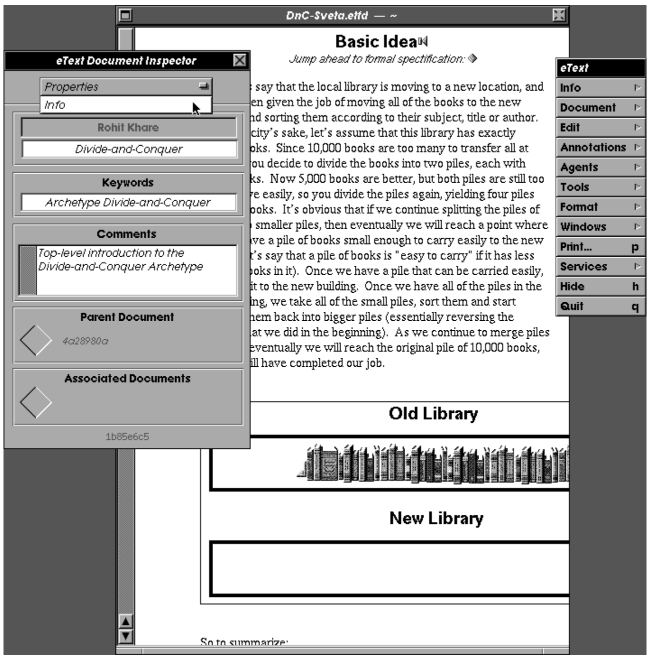
[4] Rohit Khare and the eText Group. The eText engine: An extensible object-oriented hypermedia publishing system. In *Submitted to Proceedings of the 1994 European Conference on Hypermedia Technology*, 1994.

[5] Rohit Khare and the eText Group. The eText project: Creating electronic textbooks. In *Submitted to Proceedings of the 1994 European Conference on Hypermedia Technology*, 1994.

[6] Norman K. Meyrowitz. Intermedia: The architecture and construction of an object-oriented hypermedia system and applications framework. In *OOPSLA '86 Proceedings*, 1986.

[7] Richard L. Phillips. MediaView: a general multimedia digital publication system. *Communications of the ACM*, July 1991.

[8] Adam Rifkin. Teaching archetypical design with an electronic textbook. *Proceedings of the 25th ACM SIG CSE Conference*, March 1994.


Basic Idea

Jump ahead to formal specification:

say that the local library is moving to a new location, and
en given the job of moving all of the books to the new
nd sorting them according to their subject, title or author.
city's sake, let's assume that this library has exactly
oks. Since 10,000 books are too many to transfer all at
ou decide to divide the books into two piles, each with
ks. Now 5,000 books are better, but both piles are still too
e easily, so you divide the piles again, yielding four piles
ooks. It's obvious that if we continue splitting the piles of
smaller piles, then eventually we will reach a point where
ve a pile of books small enough to carry easily to the new
t's say that a pile of books is "easy to carry" if it has less
oks in it). Once we have a pile that can be carried easily,
it to the new building. Once we have all of the piles in the
ing, we take all of the small piles, sort them and start
hem back into bigger piles (essentially reversing the
at we did in the beginning). As we continue to merge piles
eventually we will reach the original pile of 10,000 books,
ill have completed our job.

So to summarize:

eText Document Inspector

Properties: Info

Rohit Khare

Divide-and-Conquer

Keywords: Archetype Divide-and-Conquer

Comments: Top-level introduction to the Divide-and-Conquer Archetype

Parent Document: 4a28980a

Associated Documents

1b85e6c5

Figure 1: *An early* eText Engine *version of the Divide-and-Conquer chapter seen in Figure 4. The first button is a miniature speaker icon; the diamond is a link. The graphic is an inline slide show. The inspector is currently displaying the basic navigational information about this document: author, date, etc, and has diamond-shaped link-wells for dropping links to associated documents.*

dependent application, so the whole model was not designed to scale to handling hundreds or thousands of documents, nor to easily creating new ones. Publishing documents to a wide audience was also limited by the hardware and software requirements of the system.

The advent of the *World-Wide-Web* [2], particularly *Mosaic*, made publishing possible for our project. We refocused our development on creating HyperText Markup Language documents, and were successful with porting text structure, links, and some media types, but could not present the interactive ideas we had developed earlier. We had some success with using X toolkits to reimplement one of our simulations as an X application, and we are exploring the new capabilities of Forms, but we felt a real need to integrate the capabilities of the the two variants of our textbook.

2.2 The eText Engine

We planned a document-centric authoring system that could support hypermedia documents with fully interactive components, but could also automatically publish a subset of these documents to the Web, and other formats in the future.

The eText Engine is being developed under *NeXTSTEP* [3], an advanced object-oriented development environment, and is shown in Figure 1. For creating, viewing, and converting documents, it is comparable to a word processor, with multiple documents, multiple undo/redo, rulers, WYSIWYG, printing, faxing, spell-checking, drag-and-drop, and so forth. New annotations can be created in a variety of ways. For example, an audio annotation can be created by importing an audio file, pasting audio data from another application, dragging in a sound icon, or choosing a menu command. Click on the sound-icon, and the Inspector displays the waveform, editing, and playback tools. Similar ease-of-use applies for other media types, even running custom simulations. Linking is achieved through a drag-and-drop operation; simply create a bookmarked region, and drag it out to another document.

3 APPLICATIONS

The three variants of the textbook we have worked on have taught lessons in their applications to our twin challenges of education and publishing.

3.1 Education & Reference

We initially deployed the *CraftMan* version to teach an undergraduate course at Caltech. From student feedback we observed that the interactive features were a valuable part of the experience. Contrasting this with later experience with the *Mosaic* version, we noted that developing custom extensions should be an integral part of an authoring environment.

Navigation is simplified by the regular structure of Archetypes over the three-level Archetype, Application, and CaseBook implementation hierarchy. However, scenarios for an expanded textbook along with a much wider selection of implemented codes all require more powerful navigational models, particularly for exploring the text. Full-text search is the beginning of the process for a professional programmer trying to find a relevant Archetype for parallelizing code; associated documents need to be retrieved, links made from point-to-region, and incorporation of information outside the hypermedia system. Teaching scenarios included guided tours, personalization, and user-tracking navigation. Many of these lessons have been incorporated into the Engine's architecture and implementation.

3.2 Hypermedia Publishing

The compound-document model used at the core of the Engine outputs Enhanced Text Format documents (ETF) using Microsoft RTF for formatted text and a directory for object annotations to store component data (images, audio, simulation state). This mechanism can be remapped to produce other formats, such as HTML. By automating the process,

eText: Electronic Textbooks

Rohit Khare and the eText Group
The eText Project at Caltech
MSC 256-80, Computer Science
Pasadena, CA, 91125
fax : (818) 792-4257
e-mail: khare@cs.caltech.edu

ABSTRACT

The eText Project intends to create an “electronic textbook” for parallel software engineering. The Project has produced teaching and reference materials using custom applications, World-Wide-Web and Mosaic, and a new authoring and publishing environment, the eText Engine. The Engine is a user-friendly, object-oriented, compound document editor that provides a document-centric metaphor for hypermedia, as well as a flexible testbed for developing custom interactive media objects. Demonstrations of each of the systems discussed within highlight the use of hypermedia publication and education.

KEYWORDS: Compound Document Architecture, Hypermedia, Education, Publishing, Object-Oriented Design, User Interface Design

1 INTRODUCTION

Caltech’s Archetypes eBook Project was established to create an “electronic textbook” to teach computer science concepts such as parallel programming. In our vision, a student can be guided by the textbook, an intelligent, multimodal, interactive knowledge base, as well as professionals. The eText Engine was developed when we determined that existing tools for authoring (and publishing) large hypermedia documents were not suited to presenting our content.

The Archetype software design methodology is the domain of a larger community of computational scientists, at Caltech and elsewhere. Our goal is to build a support system for that research, trying to balance the competing requirements to provide interactive simulations for teaching undergraduates, a professional-grade development support system, and publishing the system as widely as possible. For more information on the project, consult [1, 4, 5, 8].

2 SYSTEMS DEVELOPMENT

To some degree, the eText Project is systems-driven. Our vision for the text has been largely shaped by the capabilities of current hypermedia tools. As we developed the editorial structure of our text (see §3), we also surveyed the field and prototyped several systems, which eventually led to the creation of our own tool, the eText Engine.

2.1 Prototypes

In the early stages of the project, our plans were defined by the features of three systems we worked with. We studied, but did not adopt, database-oriented systems, small-focus systems (“card”, “frame”, and “web” metaphors), hyper-“text” systems that could not support interactivity, time-line based presentation packages, or systems not currently supported (e.g. *Intermedia* [6]).

MediaView [7], running on *NeXTSTEP* was our initial inspiration. One demonstration of *MediaView* featured an interactive graphics paper, including narration from the author, graphic and textual annotations, and a working simulation of the actual lighting model embedded in the document. From this, we appropriated a compound-document metaphor, and the need for interactivity. Rather than an animation or even a video of the lighting model simulation, we noted that the power came from actually implementing the model, right there for the reader to play with. However, *MediaView* did not have any support for linking documents or navigation in general, nor continued support from its authors.

Xanthus *CraftMan*, also on *NeXTSTEP*, is an interactive, interpreted prototyping environment we used to develop our first chapters. With it, we refined our ideas about interactive media by developing more algorithm visualizations, simulations, and quizzes. We began to attack the navigation problem, but each “document” we created was really an in-