Reusable Hypertext Structures for Distance and JIT Learning

Anne Morgan Spalter
Department of Computer Science
Brown University
Providence, RI, USA
Tel: 1-401-863-7615
E-mail: ams@cs.brown.edu

Rosemary Michelle Simpson
Department of Computer Science
Brown University
Providence, RI, USA
Tel: 1-401-863-7651
E-mail: rms@cs.brown.edu

ABSTRACT

Software components for distance and just-in-time (JIT) learning are an increasingly common method of encouraging reuse and facilitating the development process[58], but no analogous efforts have been made so far for designing hypertext components that can be reused in educational offerings.[1] We argue that such structures will be of tangible benefit to the online learning community, serving to offload a substantial burden from programmers and designers of software, as well as allowing educators without any programming experience to customize available online resources.

We present our motivation for hypertext structure components (HTSC) and then propose a set of pedagogical structures and their building blocks that reflect the categories of lecture, laboratory, creative project, playground, and game[38].

KEYWORDS

Components, design patterns, education, hypertext structure components, interactive graphics, spatial hypertext, structural computing, temporal hypertext

THE NEED FOR HYPERTEXT STRUCTURES

The motivation for HTSCs arises from an ongoing effort at Brown University to leverage the computer's potential for use in education. This 30-plus year effort has included several hypertext projects led by Andries van Dam and others in the Brown University Computer Graphics Group and the IRIS project. This work included HES (Hypertext Editing System) developed with Ted Nelson in the 1960s [16], FRESS (File Retrieval and Editing System), developed in the 1970s and the first hypertext system used to teach a liberal arts course[22], and IRIS Intermedia in the 1980s[37][76], a UNIX-based networked hypertext system with advanced features used for teaching undergraduate and graduate courses

1. However, work has been done in identifying patterns for hypertext [7] and the design of hypertext[60][61][49].

These efforts led to the notion of an *electronic book* with *interactive illustrations*, a new form of textbook that took advantage of the power of hypertext and the power of 2D and 3D interactive computer graphics. This model, however, proved difficult to apply for all but the most determined and privileged educators. Few teachers have the time, inclination, ability, and support necessary to write a textbook or develop interactive software, using either hypertextual or linear formats (although when they do, the results can be extraordinary, as with Professor Thomas Banchoff's electronic text on multivariable calculus [3]).

We could think of no way to make writing a textbook easier, so chose to focus more recently on the illustration aspect, designing, and establishing helpful guidelines for creating, interactive teaching tools that stress exploration and discovery. We called these tools *exploratories* [28] instead of interactive illustrations to better convey our goal of interactive microworlds in which objects have behaviors and users can interact with concepts and phenomena. To complete the sense of being inside an explorable world, we took the text inside the applications.

At first, our exploratories resembled multimedia software: each had several modes and taught a whole sequence of ideas. In these exploratories, *the text was structured by the software design.* For example, in an applet teaching animation, explanations of animation concepts were available in the "explain mode," help text in the "Help" mode, demonstrations in the "show me" mode, and attributions in the "About..." mode. There was no way to see all the text at once or print it out, and no way for users to alter or remove it.

Figure 1 shows an applet for exploring different filters in scaling an image. This scaling applet offers many features for varying the filter shape (including directly drawing a custom profile), uniform and nonuniform scaling, preset examples to explain important filter facts, and several images to choose from. Its interface is structured for different types of interaction and display. A text box brings up both explanations of the concepts and context-sensitive help when the user rolls over different parts of the UI [28].

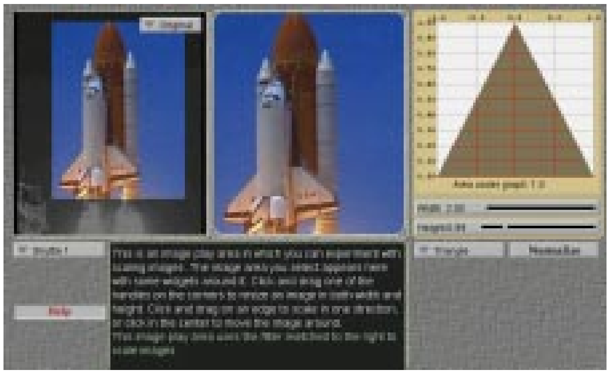

Figure 1: Filtering and scaling exploratory

When it was finished, everyone involved in the filtering applet's creation was pleased with the results. Informal demonstrations to students were also positive. Feeling that we had created a useful exploratory that embodied a "learning through exploration" pedagogy, we did a formal user study in our introductory graphics programming course[10]. Half the class (chosen randomly upon entering the lecture hall) used the applet. The other half acted as a control group and saw a sequence of static pictures generated by the applet.

Contrary to our expectations, the results were ambiguous. There was no clear evidence that the applet helped anyone understand the concepts better (based on a set of varying and randomly ordered test questions administered to both groups). After analyzing feedback questionnaires and conducting interviews, we concluded that the main problems stemmed from a lack of structure, either within the program or an accompanying pages, that would let students know exactly what ideas were being presented, how to tell if they had discovered all the topics or things one was supposed to learn, and that would provide a way to determined if they had learned them.

Not only was the pedagogical result questionable, but the programmer was an exceptional student and he spent over three months (working part-time) creating the applet and working by himself and with others to add text. Other applets took entire summers of full-time student work! The large scale of the projects also meant that the code was difficult to read and reuse, and that a substantial amount of text would have to be written to accompany a online version. While such an approach can be ideal for some circumstances (of both development and use), for most, including for development in a university setting for use by varied audiences, it obviously was not.

In their 1999 SIGSCE paper, "Granularity in the Design of Interactive Illustrations,"[33] Exploratory members Gould and Simpson detailed this problem and demonstrated the advantages of a *fine-grained* approach to interactive illustration/exploratory creation. The fine-grained approach breaks the subject matter up into a series of small programs, each of which teaches a single main concept, and embeds them in text. They define granularity as "the conceptual scope covered." Smaller-grained or fine-grained applets take on "small, atomic concepts." The example in Figure 2, for instance, is an applet that teaches just the concept of sampling, with two options, point and unweighted area sampling. The reader is led through a whole series of such applets to learn about the Nyquist limit, weighted area sampling, filter shapes, and seven separate applets for presenting convolution.

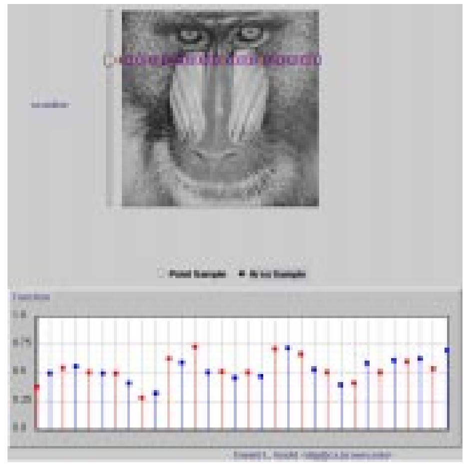

Figure 2: A fine-grained applet

A Shift Toward Components

The fine-grained applet approach removed many of the programming hurdles associated with a program's complexity, from software design issues to the limited time that undergraduates, the chief programmers on this project, had in their schedules. It also served to remove a great deal of the pedagogy from the source code and move it to the Web page, making the applets more flexible for use by others.

The need to repeat certain interface elements, mathematical calculations, and interaction techniques throughout a set of applets, inspired us to think about modularizing our efforts even further. Each fine-grained applet can be thought of as a flexible component of a larger effort, and many of the features of the applets could, we felt, in turn, be recombined to create new versions of existing applets or entirely new ones. In particular, a staff member or upperclassman could program particularly complex portions, such as the math behind some of the filtering methods, and undergraduates (or educators accessing material remotely) could design their own applets. This approach requires software components, pieces of code that can be plugged into different applications and used without modification. A component architecture is achieved by instituting an expected set of named entry points into the code and enforcing a set of naming conventions and introspection capabilities.

Other researchers working on the problem of educational software development have also moved to component architectures[41] and some envision their prime usefulness as suppliers of components rather than authors of complete applications or systems. In their paper "Developing Educational Software Components"[58] Roschelle et al discuss the challenges of creating components with useful cognitive characteristic for educators, such as those furthering the type of constructivist pedagogy[13][51][52][62] that has inspired Exploratories. They envision a potential marketplace in components for small educational software concerns, as well as teachers.

This change from monolithic program to interchangeable, easier to write, interconnectable components has enormous positive implications for the future of distance and JIT learning[42]. But no such granularity analysis has been done for the textual portions of exploratory-type efforts. We knew that the electronic book model was often unrealistic, but also that applets placed on the Web with no supportive text (other than brief descriptions of the content) did not find the larger user base we had anticipated. Through our user test, we found that small areas of changing text within the applet did not offer the type of guidance needed by students, Our most successful applets were those accompanied by significant textual explanation[4]. It was obvious that we needed the text back, but in what form?

We looked closely at our own need to facilitate the hypertext associated with interactive applets and at the lack of a good model for doing so by others, and realized how similar the situation was to our problem with software granularity and reuse. The fine-grained S/E model moved some of the structure of the learning experience out of the applet and into the Web page (by breaking up and defining the "lessons" available in a monolithic implementation), both making the programming easier and making the results easier to reuse by others. Our concept of reusable hypertext structure components (HTSC) further abstracts out structure and pedagogy from the code and places it in the text. The result is that the software components are even more easily reused, the programming is easier because different aspects of the experience are associated in a hypertext structure rather than within a piece of code, the text is supportive of, but separate from the software at hand, and different pedagogical aspects of teaching are more easily adapted to the online world.

Just as viewing a set of software components can help teachers think of new interactive scenarios, so, we believe, perusing a set of HTSCs and seeing examples will not only relieve a design burden, but also inspire thinking about the way that interactive works are used in class, at home, or over the Web.

REUSABLE HYPERTEXT STRUCTURES

Here we present reusable HTSCs for different pedagogical approaches to distance and JIT learning. The structures are described in general terms. Due to the limitations of HTML, our current implementations are effectively guidelines or templates rather than truly reusable structures, but future versions using XML[29] and XLink[75] will more completely fulfill the vision. Other hypertext features such as guard fields [67][39][40], one-to-many links, basic structural formatting, ability to enter mathematical expressions, and use of computational links will also contribute to richer structural definitions[76][19][46] [45][53]Suppose, for example, that the Signal Processing page had guard fields and one-to-many links. The guard fields would enable the system to track a student's state and dynamically generate menus for the set of one-to-many links contained in the document being viewed. These menus would contain a variety of links to destinations including remedial content or extended exploration of the underlying mathematics in addition to standard summary and comprehensive views of the topics being presented.

Building-Block Structures

As in the world of software components, we begin with simple but useful building blocks[15], analogous to basic interface widgets and operations, that can both inspire new ways of thinking about an educational approach and provide tools that make realizing one's vision practical in real-life circumstances. All references to online structure templates and use of the structures in our work can be found at [35].

Island with Gateway(s)

An Island with Gateway(s) structure (see Figure 3) provides a clear boundary between a main interactive learning structure and related internal or external areas. Links off the island are through one or more well-defined gateways. All the pedagogical structures we discuss are Islands with Gateways.

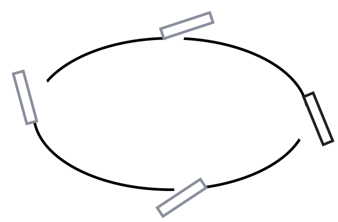

Figure 3: Island with Gateways

Island with Gateway(s) is an important structure for managing the potential complexity of hypertext-based learning structures. On the Web, for example, it is notoriously easy to follow links out of a site without realizing one has done so.

One or more gateways should be clearly defined and should be the only place from which one can leave the island. This helps both to structure the main learning experience (because one is not constantly presented with options for leaving it) and to make further investigation (say through additional resource links) easier by collecting outgoing paths in a single area.

Local Island with Bridge(s)

Local Island with Bridge(s) is, in effect, a nested island. It does not link out of the main structure, but can differentiate essential from optional material or define a tightly integrated area within a more loosely structured set of resources (see Figure 4). Our main exploratory Web site uses Island with Gateways and the Color Web within it is a Local Island with Bridges[35].

.

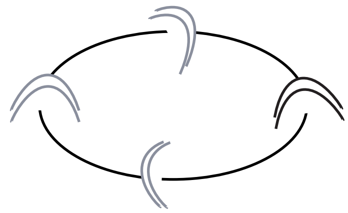

Figure 4: Local Island with Bridges

Locator

A Locator structure provides “You Are Here” information to place the user of a specific educational experience within a larger context. This service is especially important as interactive elements become more fine-grained. Locators can have many forms, including, as shown in Figure 5, a) a linear before-and-after progression, b) a fisheye before-and-after view to accommodate varying levels of detail, c) a position in one of several hypertext trails, or d) a graphical map. Our implementation of this structure includes content stubs that make the templates’ purpose more self-disclosing, much as Trigg’s typed links did. An example of a Locator structure can be seen on our examples site in the Transformation Game.

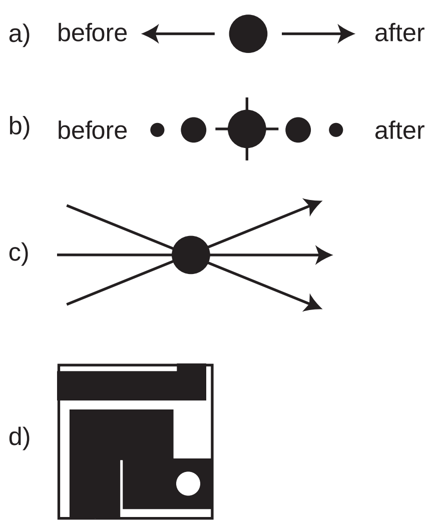

Figure 5: Locator

Notebook

Notebook is a Reflection/Collection structure (see Overview description), that provides a space for students to record their thoughts, lab or lecture notes, game tips, or anything else of relevance. Our current implementation takes a simple first step toward such tools for reflection and provides an easily reused and extended HTML template consisting of a titled starting point with bidirectional links to different, labeled sections, shown in Figure 6. Content stub text encourages users to include images and graphs that can become links within the Notebook. Our longer-term plans include indexing and table of contents tools, as well as embedded interactive graphing and calculating tools[48]. Notebooks can become valuable organizational and portfolio resources for the student, can be used for instructor grading and feedback, and can be posted online to share with others.

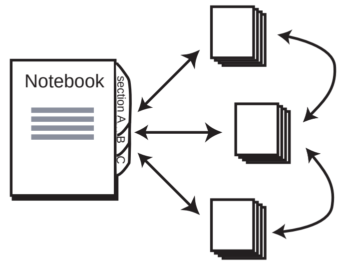

Figure 6: Notebook

Link Types

Link types are used throughout our structures to free the type of link from the text label used in any given implementation and to guide content creation [69][8]. This leaves teachers free to design a page appropriate for their audience without making the structure any less reusable by others or losing touch themselves with the original intentions of the links. In the Overview structure, for example, the headings are link types. Users of the structure create their own text headings. Each link type is followed by a content stub that a user replaces with the content described. Unlike software components, teachers can tailor hypertext components to their own needs without becoming programmers.

Synchronization Links

Our synchronization links are an implementation of temporal linking[14][34][63][68] that bring some of the time-based aspects of live learning and pedagogy to the online world. Linking parts of the structures in time can preserve some of the suspense of classroom discussion and demonstration, and sychronization can use a time-based element, such as video, to drive the pedagogical approach of a distance or JIT learning experience. Such associations can also be used to tie unscripted interaction (such as that taking place in a chat room environment) with specific portions of the predetermined content of the course or lesson.

Overview

Overview is a important structure, present in some form in virtually all of best applet and course examples we could find online. (See, for example, Gamelan’s “cool” educational applets [31].)

Overview uses Local Island with Bridges to situate it within the larger structure. Its structure is a directed graph with bidirectional typed links to substructures that vary for each of its components. Overview is used in examples henceforth. Overview elements specific to different pedagogy structures are discussed in those structure descriptions

.

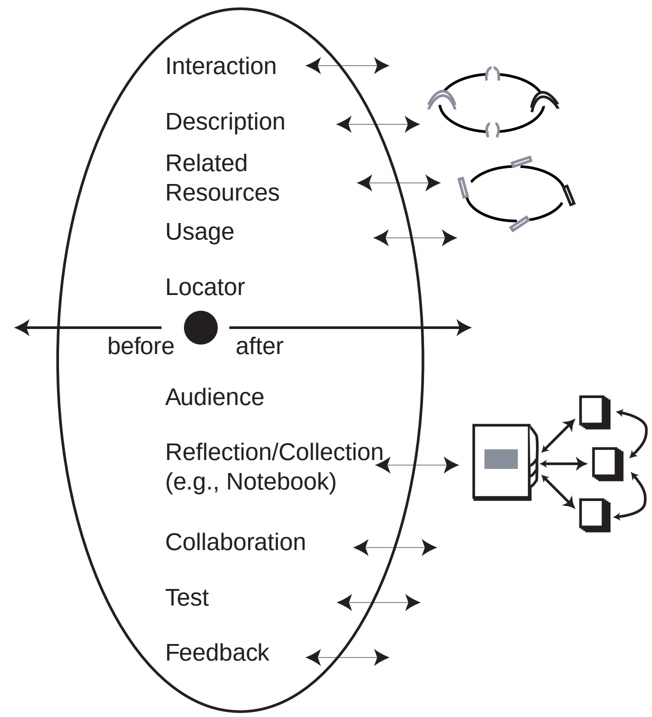

Figure 7: Overview

Interaction: Screen shot(s) and/or text that links to main applet(s) or other interactive area(s).

Description: Brief description (a few paragraphs) of the topic being taught and its importance/relevance. Should have a link to more explanation, and from there links to related topics. Description should use the Local Island with Bridges structure. It can be simple, with a few layers of tree-style links, or arbitrarily complex—for example, a complete hypertext book.

Related Resources: Single-line descriptors that link to areas within and outside the main structure, such as teacher's overheads and/or notes for Lectures, suggested readings for Playgrounds, or reference resource for a Laboratory. Related Resources that link outside the main structure should use Island with Gateways.

Usage: Short technical requirements and directions for using the interactive programs. Can also include links to more detailed instructions and directions for using the main site/structure.

Locator: Implementation of the Locator structure, described above, that gives a high-level overview and often leads to more detail.

Audience: Specifies such descriptors as age-appropriateness, grade level, prerequisites (can be linked to Locator detail), and use contexts, e.g., good for classroom demos vs. good for self-study.

Reflection/Collection: Brief description and link to Reflection/Collection structures such as Notebooks, Lab Notebooks, and Portfolios (discussed in relevant pedagogy descriptions).

Collaboration: Brief description and links to collaborative components such as chat rooms and bulletin boards.

Test: Brief description and link to formal test areas or to self-tests, such as multiple choice HTML forms. Programs such as Macromedia Dreamweaver Attain can automatically generate HTML for various tests formats. Test can also include more open-ended "testing" such as a list of questions to think about.

Feedback: Brief description (such as "mail us with your comments") and link to email, feedback form, or other user evaluation structure.

Additional suggested components for Overview include to *FAQ*s, *online help*, a *news* area, an *index*, *glossary*, and *text search.*

Pedagogical Structures

The following structure descriptions begin with a short working definition of the genre and include an explanatory diagram. Since most of these structures are easy to grasp visually, we do not write at length about how each one should be used, but instead focus on the chief benefits of each structure and any reuse of building block structures.

Lecture

Today it is common practice for college courses to have a Web site containing documents such as the syllabus, handouts, assignments, lecture notes, and even discussion areas. Courses delivered entirely online often have similar structures, with more emphasis on discussion and feedback areas [71]. Because the lecture format is so well established, it immediately suggests integrated building block structures, such as those for note-taking. The structures described next (Laboratory, Creative Project, Playground, and Game) are often found within a Lecture structure.

Our hypertext structure is designed to preserve as much of the live feeling of the classroom experience as possible, while at the same time taking advantage of the potential for using the materials in a nonlinear, self-paced fashion. For example, bi-directional links can be placed throughout the lecture notes and video or audio sequences to allow synchronized but nonlinear navigation. A student can also synchronize their own notes with specific overheads or video placeholders.

.

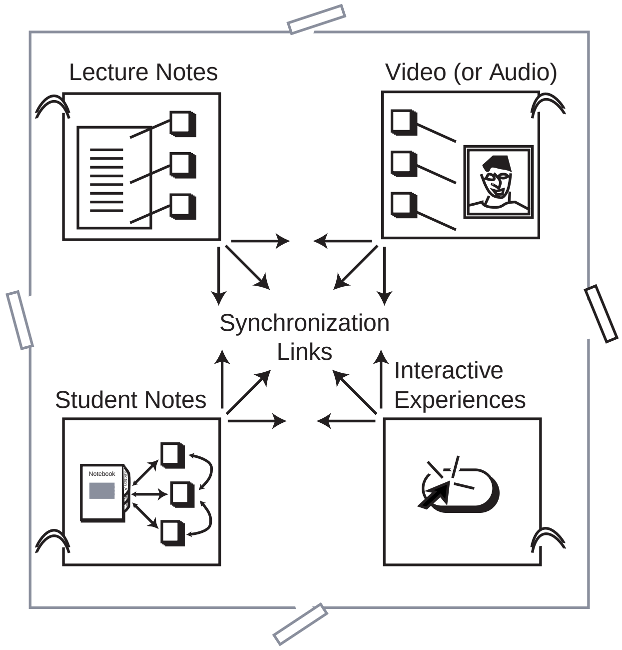

Figure 8: Lecture (without Overview links)

In Lecture, Overview's Description links to lecture notes and other immediately relevant materials within the site. Overview's Related Resources links to resources within the site including video, the student note area, and interactive experiences shown in Figure 8. The key benefits to Lecture are the top-level view provided by Overview, the use of Local Islands with Bridges to define relevant materials, and the synchronization of the notes, video, notebook, and interaction sections. While most online courses offer some version of this structure already, few synchronize its components. For an example of synchronized links, see the Web site of Brown's introductory programming course[12]. In this site, students can click on any overhead and immediately see the relevant portion of the audio, and vice versa. Although the course offers many interactive teaching aids, they are not yet presented in the fashion described here (but they are accessible on the site). The course site uses Classroom2000 software, developed at Georgia Tech [1].

Laboratory

A laboratory session, like a lecture, is a well-known learning and research format. A lab can offer a range of restrictions, from step-by-step instructions and prescribed user actions to more free-form, playground-like experiences. Most laboratory simulations available today fall toward the highly restricted end of this spectrum. The heart of our Laboratory hypertext structure is shown in Figure 9.

As usual, the main structure has an Overview (not shown here). The focal point, as with the Lecture, is a synchronized link set between a Reflection/Collection structure, the Lab Notebook, and a set of Experiments (assumed in our case to be interactive).

The Lab Notebook is an implementation of Notebook tailored to specific Lab needs. Currently, this distinction must be made by the user of our general Notebook structure, but we are experimenting with Java elements for a Lab notebook that would offer basic graphing and calculation.

We do not propose a specific Experiments structure at this time.

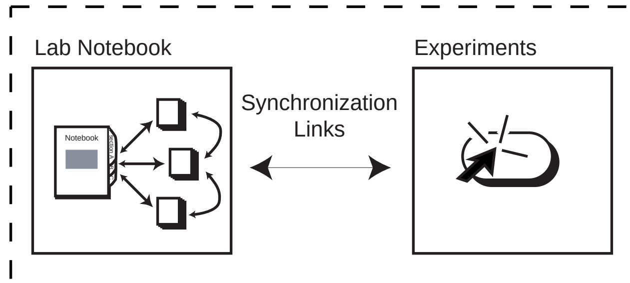

Figure 9: Laboratory

Creative Project

Creative Project, like Laboratory, links an interactive area, the creative workspace, with a Portfolio Reflection/Collection structure. A creative project is always supported by substantial student-teacher interaction, since without any feedback it is difficult to grow and learn as a creative individual. Some criteria for self-assessment is therefore vital for creative projects to be more than entertainment or useful only for especially self-critical or talented learners. Such criteria can be built into the project in some manner or may be present in the surrounding hypertext structure, for example in the Test structure, linked off the Overview and from the Creative Project Overview's Interaction section.

The Portfolio could be a Notebook-like structure, but, especially for large numbers of images, a separate database program is more realistic. Canto Cumulus, which has an online version, for example, offers standard key-word searching, thumbnail views, printing options, annotation areas, and media management tools.

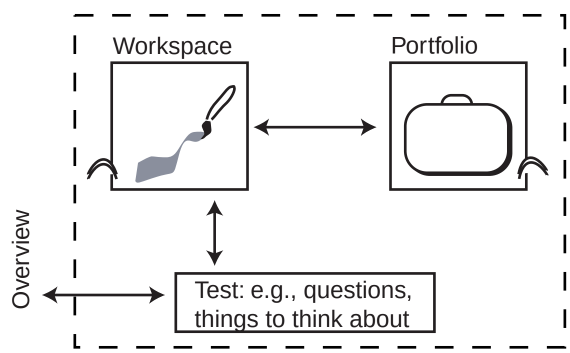

Figure 10: Creative Project

Playground and Game

Playground and Game do not use any new structures. Both use Overview, and Playground could also use a Notebook or

Portfolio. The pedagogies are defined not so much in the hypertext structures used, but in the interactive software and the text content.

A Playground can be a lot like a Laboratory, for example, a place to experiment and observe. The use of Laboratory's structured experiments and the Lab Notebook, however, suggest a particular way of guiding the student. In a Playground, the hypertext structure should provide support but maybe not in the form of assignments or exercises. Because the spirit is one of open-ended investigation, the Test section can play a vital role, helping users confirm that they have indeed learned something. Test might have "questions to think about," rather than a series of problems to be solved and turned in for a grade. On our structure templates and examples site we show a Playground hypertext structure for the filter and scaling exploratory shown in Figure 1.

In a Game, one learns either through direct or incidental techniques. In either case, the Overview structure helps to ensure that the educational goal is explained up front so users can decide whether the subject matter and approach is appropriate for their needs. The pedagogy is inherent in the interactive software so less needs to be represented in the hypertext. For example, a Game has, in effect, built-in testing as the user progresses toward the goal or through the levels of difficulty, so the Test link from Overview would probably not be emphasized. While Playground might have "questions to think about," a game could have tips, hints, and special challenges that guide the user to explore certain topics or methods. On our structure templates and examples site we show a Game hypertext structure for a game that gives students an intuitive feeling for the order of the geometric transformations (i.e., translation, scaling, and rotation).

Using and Reusing the Structures

We have found these structures useful in thinking about our own work (see examples at [35]), but the question remains of their general viability and the extent of their resusefulness by others. We hope that readers of this paper will experiment with our resources and ideas and begin a dialog about their experiences.

FUTURE WORK

Our near-term plans are to bring the strengths of XML and XLink to bear on our structures and their reusability. We have recently finished designing a repository for Java Bean educational components[5] and will soon start one for hypertext structures as well. By making a repository for HTSCs public, we hope to accelerate the pace of their development.

We plan to apply our structure ideas to 3D desktop and virtual reality[72] educational programs, using ideas from spatial hypertext to develop three-dimensional forms as structure components. A locator in a virtual world, for example, could draw on the extra dimension to give more complex information than is possible on a desktop machine. Drawing on Landow and Kahn's example[43], we hope to make structures that ease the transition between platforms, languages and applications.

RELATED WORK

From the beginning, with FRESS[22] and later Intermedia[76], hypertext systems have been used for classroom education[26][9][18][20][23][66]. More recently, Walden's Paths builds upon earlier work with scripted documents[77] and guided tours[70] [44] to provide classroom teachers with structures for using existing WWW materials[30]. In a report on their experiences with actual classrooms, they describe some of the persistence and versioning issues intrinsic to this domain[64].

The surging demand for JIT and distance learning is driving the search for better ways to develop and distribute online education[36][55][42][73]. Our approach addresses that need, treating the WWW as a distribution medium, with a focus on providing reusable structures for embedding educational materials.

Educators, both teachers and technologists, have long been concerned with issues of reuse—using teaching materials in widely different contexts[13][51], and flexibility—tailoring educational materials to the needs of individual students[52]. Unfortunately the track record has been less than thrilling [17]. Part of the problem is that creating educational materials is extremely labor and high-level skill intensive; most teachers simply do not have resources to investigate and use new technologies. Current efforts to address these issues include Brown's educational software seminar[11], ESCOT (Educational Software Components of Tomorrow)[25], E-Slate components[27], ACOT[62], and pedagogical patterns[54]. We extend those approaches by treating the link structures as objects in their own right. Currently these are simple HTML templates, but, as we describe in future work, our ultimate goal is to provide hypertext structures as fully functional hypertext components.

Reuse and components have been part of software engineering goals from the earliest days of computers[47][74]. Recently, design patterns[2][32][56] have provided structured metadata descriptions that facilitate the reuse of computational algorithms and strategies. Their success has spawned an ever-expanding series of attempts to mine[21] diverse areas of human endeavor for useful patterns, including hypertext design[49][60][61] and education[54]. None, however, have as yet dealt with the notion of reusable hypertext structures. We have consciously used a pattern-directed approach to identifying reusable hypertext structures.

In "As We Should Have Thought"[50] Nuernberg et al. asserted the primacy of structure over data and identified the need for a structural computing paradigm. They pointed out that hypertext models such as spatial hypertext[65] cannot be adequately addressed by the more traditional paradigms[57]. Bernstein's reports on his results with hypertext pattern mining[7] provide a useful start in the direction of structure identification. Our work directly furthers the objective of structural computing by continuing the work of identifying and classifying structures.

ACKNOWLEDGMENTS

Thanks to Mark Bernstein, Roger Blumberg, Dan Gould, Jean Laleuf, Elli Mylonas, Peter Nuernberg, Dave Peck, and Allen Renear. This work is sponsored by Adobe Systems, Inc., Advanced Network & Services, Alias|Wavefront, The Center for Advanced Scientific Computing and Visualization, the DOE, IBM Corporation, Intel, Microsoft, the NSF Science and Technology Center for Computer Graphics and Scientific Visualization, and Taco, Inc.

REFERENCES

1. Abowd, G., Pimentel, M., and Ishiguro, Y. 1. Supporting Life-Long Learning Through Automated Capture of Educational Experiences, submitted to Journal of Continuing Engineering Education and Life-Long Learning (IJCEELLL, a UNESCO Journal) (under review), 1999.
2. Alexander, C., et al. A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York, 1977.
3. Banchoff, T. Interactive Laboratories for Calculus of Several Variables. http://www.geom.umn.edu/~banchoff/Calculus/Calculus.html
4. Beall, J., Doppelt, A., and Hughes, J. Developing an Interactive Illustration: Using Java and the Web to Make it Worthwhile, in The Internet in 3D: Information, Images and Interaction, Earnshaw, R. and Vince, J., editors. Academic Press, San Diego, 1997, pp. 55-64.
5. Bean House repository, just released. http://www.beanhaus.org.
6. Bernstein, M. Conversations with Friends: Hypertext with Characters, in Proc. of International Workshop on Hypermedia Design 1995, (Montpellier, June 1-2, 1999), Springer Verlag, 1996.
7. Bernstein, Mark. Patterns of Hypertext, in Proc. of Hypertext '98 (Pittsburgh, June 20-24), ACM Press, pp. 21-29.
8. Bernstein, M. Hypertext Now: Link Types: A Commentary on Randy Trigg's 1983 PhD Thesis. 1999, http://www.eastgate.com/HypertextNow/archives/Trigg.html.
9. Blumberg, R. NetTech Forum on Hypermedia Teaching and Technology, STG (Scholarly Technology Group), Brown University, December 1996. HTTP://www.stg.brown.edu/edu/HTT96/
10. Brown University Computer Graphics course for undergraduates, October, 1999. http://www.cs.brown.edu/courses/cs123/
11. Brown University Educational Software Seminar, October, 1999. http://www.cs.brown.edu/courses/cs092/.
12. Brown University Introduction to Object-Oriented Programming and Computer Science, October, 1999. http://www.cs.brown.edu/courses/cso15/
13. Bruner, J. The Process of Education. Harvard University Press, Cambridge, 1960, 1977.
14. Buchanan, M. C.and Zellweger, P.T..Specifying temporal behavior in hypermedia documents, in Proc. of ECHT '92 (Milan, Nov. 30-Dec. 4), ACM Press, pp. 262-271
15. Catlin, K.S., Garrett, L.N., and Launhardt, J.A. Hypermedia templates: an author's tool, in Proc. of Hypertext 91 (San Antonio, Dec. 15-18), ACM Press, pp. 147-160
16. Carmody, S. et al. A Hypertext Editing System for the S/360. Pertinent Concepts in Computer Graphics. Faiman, M. and Niervergelt, J., editors. Urbana: University of Illinois, 1964. pp. 291-330.
17. Cuban, L. Teachers and Machines. Teachers College Press, New York,1986.
18. da Silva, D., Van Durm, R., Duval, E., and Olivie, H. Adaptive navigational facilities in educational hypermedia, in Proc. of Hypertext 98 (Pittsburgh, June 20-24), ACM Press, pp. 291-292.
19. Davis, H., Hall, W., Heath, I., Hill, G., and Wilkins, R. Towards An Integrated Information Environment With Open Hypermedia Systems, in Proc. of the ECHT '92 (Milan, Nov. 30-Dec. 4), ACM Press, 1992.
20. DeBra, P. and Calvi, L. 2L670: a flexible adaptive hypertext courseware system, in Proc. of Hypertext 98 (Pittsburgh, June 20-24), ACM Press, pp. 283-284.
21. DeLano, D.E. Pattern Mining, in The Patterns Handbook: Techniques, Strategies, and Applications, Linda Rising editor. Cambridge University Press, Cambridge, 1998.
22. DeRose, S. J. and van Dam, A. Document Structure and Markup in the FRESS Hypertext System. Markup Languages: Theory and Practice 1,1 (Winter, 1999), pp. 7-32.
23. deVries, E., Tiberghien, A., and Petitot, G. Learning Processes and Knowledge Representation in the Design of Educational Hypermedia, in Proc. of International Workshop on Hypermedia Design 1995, (Montpellier, June 1-2, 1999), Springer Verlag, 1996.
24. 1999 EC/NSF Workshop on Human-Centered Computing, Online Communities and Virtual Environments, ACM Interactions, Vol VI.5, pp 9-16, Sept/Oct 1999.
25. ESCOT: http://wise.sri.com/escot/Home.html.

26. Ess, C. The pedagogy of computing: hypermedia in the classroom, in Proc. of Hypertext ‘91 (San Antonio, Dec. 15-18), ACM Press, pp. 277-291.

27. E-Slate Exploratory Learning Environment: http://e-slate.cit.gr/.

28. Exploratories project, Brown University Computer Graphics Group: http://www.cs.brown.edu/exploratory, Exploratories user studies: http://www.cs.brown.edu/exploratory/writings.html; Scaling and Filter Shape applet: http://www.cs.brown.edu/exploratory/applets/ScaleIllus/scalingApplet1.html; Combination Additive and Subtractive Color applet: http://www.cs.brown.edu/exploratory/applets/ColorMixCombo/comboMixing1.html October, 1999.

29. Flynn, P. Understanding SGML and XML Tools. Forward by Steven DeRose. Kluwer Academic Publishers, Boxton, 1998.

30. Furuta, R., Shipman, F., Marshall, C., Brenner, D., and Hsieh, H. Hypertext paths and the World-Wide Web: Experiences with Walden's Paths, in Proc. of Hypertext '97 (Southampton, April 6-11), ACM Press, pp. 167-176.

31. Gamalan educational applets page. Those with little blue fan icons are “cool”. 1999, http://www.gamelan.com/directories/pages/dir.java.educational.html

32. Gamma, E., Helm, R., Johnson, R. and Vlissides, R. Design Patterns. Addison-Wesley, New York, 1995.

33. Gould, D., Simpson, R. and van Dam, A. Granularity in the Design of Interactive Illustrations, in Proc. of SigCse '99, (New Orleans, March 24-28) pp. 306-310.

34. Hardman, L., van Ossenbruggen, J., Mullender, K., Rutledge, L., and Bulterman, D. Do you have the time? composition and linking in time-based hypermedia, in Proc. of Hypertext 99 (Darmstadt, February 21-25), ACM Press, pp. 189-196.

35. HTSC (HyperText Structure Component) examples. Online HTSC templates and their use. October, 1999. http://www.cs.brown.edu/exploratory/HTSC/

36. IMS Internet Architecture for Learning. Standards for online learning systems arising out of Educause work, October, 1999, http://www.imsproject.org/

37. Intermedia home page, STG (Scholarly Technology Group), Brown University, 1999. http://landow.stg.brown.edu/HTatBrown/Intermedia.html

38. Jones, A., Bagford, L. and Wallen, E. Strategies for Teaching. Metuchen, NJ: The Scarecrow Press, Inc., 1979.

39. Joyce, Michael. Storyspace as a hypertext system for writers and readers of varying ability, in Proc. of Hypertext 91 (San Antonio, Dec. 15-18), ACM Press, pp. 381-388

40. Joyce, M. Of Two Minds: Hypertext Pedagogy and Poetics. University of Michigan Press, Ann Arbor,1995.

41. Klein, R. and Encarnacao, L. M. A Web-based framework for the complete integration of teaching concepts and media in computer graphics education, in Proc. ED-MEDIA'97 (Calgary, June, 1997).

42. Koning-Bastiaan, M. Connected and Scalable: A Revolutionary Structure for Online Communities, in EOE (Educational Object Economy) web site, 1999. http://www.eoe.org/.

43. Landow, G. and Kahn, P. Where's the Hypertext? The Dickens Web as a System-Independent Hypertext, in Proc. of the 1992 European Conference on Hypertext, ACM Press, 1992, 149-160.

44. Marshall, C. and Irish, P. Guided tours and on-line presentations: how authors make existing hypertext intelligible for readers, in Proc. of Hypertext '89 (Pittsburgh, Nov. 5-8, 1989), ACM Press, pp. 15-26.

45. Marshall, C. and Shipman, F. Searching for the missing link: discover implicit structure in spatial hypertext, in Proc. of Hypertext ‘91. (San Antonio, Dec. 15-18), ACM Press, pp. 261-275.

46. Marshall, C., Shipman, F., and Coombs, J. VIKI: spatial hypertext supporting emergent structure, in Proc. of ECHT '94 (Edinburgh, September 18-23), ACM Press, pp. 13-23.

47. McIlroy, M. D. Mass Produced Software Components, in Software Engineering, Naur, P. and Randell, B., editors. Scientific Affairs Division, NATO, Brussels, January, 1969, 138-151.

48. Morgan, D. Dept. of Materials Science and Engineering, MIT. email correspondence, October 20, 1999.

49. Nanard, M., Nanard, J., and Kahn, P. Pushing reuse in hypermedia design: golden rules, design patterns, and constructive templates, in Proc. of Hypertext ‘98 (Pittsburgh, June 20-24), ACM Press, pp. 11-20.

50. Nuernberg, P., Leggett, J.J., Schneider, E.R. As we should have thought, in Proc. of Hypertext '97 (Southampton, April 6-11), ACM Press, pp. 96-101.

51. Ohanian, S. Garbage Pizza, Patchwork Quilts, and Math Magic: Stories about Teachers Who Love to Teach and Children Who Love to Learn. W.H. Freeman and Co., New York, 1992.

52. Papert, S. Mindstorms: Children, Computers, and Powerful Ideas, Second Edition. Basic Books, New York, 1993.

53. Park, S. Structural properties of hypertext, in Proc. of Hypertext ‘98 (Pittsburgh, June 20-24), ACM Press, pp. 180-187.

54. Pedagogical Patterns Project. Reported at OOPSLA'97 education workshop, July, 1999. http://www-lifia.info.unlp.edu.ar/ppp/

55. PITAC Report. President's Information Technology Advisory Committee Report to the President: Information Technology Research: Investing in Our Future, National Coordination Office for Computing, Washington, Feb. 1999. http://www.hpcc.gov/ac/report/

56. PLoP - Pattern Languages of Programs conferences home page, August 5, 1999. http://st-www.cs.uiuc.edu/~plop/

57. Reinert, O., Bucka-Lassen, D. Pedersen, C., and Nuernberg, P. CAOS: a collaborative and open spatial structure service component with incremental spatial parsing, in Proc. of Hypertext 99 (Darmstadt, February 21-25), ACM Press, pp. 49-50

58. Roschelle, J., DiGiano, C., Koutlis, M. Repenning, A., Phillips, J., Jackiw, N., and Suthers, D. Developing Educational Software Components. IEEE Computer 32(9) (September, 1999), pp. 50-58

59. Rosenberg, J. The structure of hypertext activity, in Proc. of Hypertext 96 (Washington, March 16-20), ACM Press, pp. 22-30.

60. Rossi, G., Schwabe, D., and Garrido, A. Design Reuse in Hypermedia Applications Development, in Proc. of Hypertext '97 (Southampton, April 6-11), ACM Press, pp. 57-66

61. Rossi, G., Schwabe, D., and Lyardet, F. Patterns for Designing Navigable Information Spaces, in Pattern Languages of Program Design 4, Harrison, N., Foote, B., and Rohnert, H. editors. Addison-Wesley, Reading, MA, 2000. In press.

62. Sandholtz, J.H., Ringstaff, C., Dwyer, David C. Teaching with Technology: Creating Student-Centered Classrooms. Forward by Larry Cuban. Teachers College Press, NY, 1997.

63. Sawhney, N., Balcom, D., and Smith, I. HyperCafe: narrative and aesthetic properties of hypervideo, in Proc. of Hypertext 96, ACM Press, pp. 1-10.

64. Shipman, F., Furuta, R., Brenner, D., Chung, D., and Hsieh, H. Using paths in the classroom: experiences and adaptations, in Proc. of Hypertext 98 (Pittsburgh, June 20-24), ACM Press, pp. 267-270.

65. Shipman, F., Marshall, C., and LeMere, M. Beyond location: hypertext workspaces and non-linear views, in Proc. of Hypertext 99 (Darmstadt, February 21-25), ACM Press, pp. 121-130

66. Spoehr, K. T. Enhancing the Acquisition of Conceptual Structures through Hypermedia, in Classroom Lessons: Integrating Cognitive Theory and Classroom Practice, Kate McGilly editor. MIT Press,1996, pp. 75-101

67. Storyspace. Getting Started with Storyspace for Macintosh V. 1.5, Eastgate Systems Press, Watertown, 1996, Storyspace for Windows, V. 1.7.5, Eastgate Systems Press, Watertown, 1999.

68. Stotts, D. and Furuta, R. Temporal hyperprogramming. Journal of Visual Languages and Computing 1(3), (Sept. 1990), pp. 237-253.

69. Trigg, R. A Network-Based Approach to Text Handling for the Online Scientific Community. Ph.D. Thesis, University of Maryland, College Park, 1983. http://www.parc.xerox.com/spl/members/trigg/thesis.

70. Trigg, R. Guided tours and tabletops: tools for communicating in a hypertext environment. ACM Transactions on Office Information Systems 6(4), (1988), 398-414.

71. UC Online. Includes a tour of their course interface, October 1999, http://www-cmil.unex.berkeley.edu/online/about.html, tour at http://www-cmil.unex.berkeley.edu/online/tat2/html/ocment.html

72. van Dam, A. Beyond WIMP, in IEEE Computer Graphics and Applications. Vol. 20, No. 1 Jan. 2000, in press.

73. Waite, W. and Simpson, R. The Lab and the Web: Transforming the Sophomore Experience, in Proc. of ASEE '96, 1996.

74. Wilkes, M., Wheeler, D. and Gill, S. The preparation of programs for an electronic digital computer, with special reference to the EDSAC and the use of a library of subroutines. Addison Wesley Press, Cambridge, 1951.

75. XLink Standard, World WIde Web Consortium Working Draft 26 July 1999.

76. Yankelovich, N., Haan, B.J., Meyrowitz, N.K., and Drucker, S.M. Intermedia: the Concept and the Construction of a Seamless Information Environment. IEEE Computer 21(1), (1988).

77. Zellweger, P. Scripted documents: a hypermedia path mechanism, in Proc. of Hypertext '89 (Pittsburgh, nnn), ACM Press, pp. 1-14.