Browsing Intricately Interconnected Paths
Pratik Dave, Unmil P. Karadkar, Richard Furuta, Luis Francisco-Revilla, Frank Shipman, Suvendu Dash, Zubin Dalal
Center for the Study of Digital Libraries and Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
Phone: +1-979-845-3839
{pratik, unmil, furuta, l0f0954, shipman, suvendud, zubin}@csdl.tamu.edu

ABSTRACT
Graph-centric and node-centric browsing are the two commonly identified hypertext-browsing paradigms. We believe that path-centric browsing, the browsing behavior exhibited by path interfaces, is an independent browsing paradigm that combines useful aspects of the two commonly supported cases. Paths have long been recognized as an effective medium for aggregating and communicating information and have been included in various hypermedia systems as alternate metaphors or supporting tools. The Walden’s Paths project promotes path-centric traversal as the primary browsing mechanism over Web-based materials. This paper expands the notion of our paths to include more generalized structures and interconnections across paths. We present an architecture for describing complex networks of such paths. Finally, we discuss the design and present a prototype implementation of the Path Engine, a tool that provides a linear interface for browsing intricately interconnected paths.

Categories and Subject Descriptors
H.5.4 [Hypertext/Hypermedia]: Navigation, User issues.

General Terms
Design, Human Factors.

Keywords
Path-centric browsing, Navigation metaphors, Directed paths, Walden’s Paths, Path Engine.

1. INTRODUCTION
Hypermedia systems support the creation and presentation of large networks of nodes containing information elements, and links expressing relationships between these nodes. Utting and Yankelovich discuss the different display mechanisms and browsing behaviors supported by various hypermedia systems [32]. In their analysis browsers are categorized as either graph-centric or node-centric. Graph-centric browsers display overviews of the directed graphs of nodes and links that form the hypertextual net and permit user browsing via selection from among displayed links. Node-centric browsers, including most present day web browsers, display the information and actions (links) available at the current node(s) and support user interaction solely among the nodes reached while browsing. While graph-centric browsers must address the issue of presenting large information nets coherently and without overwhelming users with unnecessary information [32], node-centric browsers fail to convey information about the size of the hypertext and the location of the user in the hypertextual network [5].

Paths or trails [4][31][35] have long been used in hypertext systems as an interface to large hypermedia networks. Zellweger has suggested the inclusion of paths as “first-class citizens” of their hypertext systems rather than merely as add-on features. Typically, paths are created by users of the system, but may also be generated by the system based on specifications provided by the users. Authors of paths select a subset of the information contained in the hypertextual networks and organize it for use in specific contexts of interest to them and their readers.

Paths present information elements certified by the authors to be relevant to user activity. In comparison with displays generated by graph-centric browsers, path overviews tend to be more narrowly focused upon a theme and thus display information that is more relevant to the users’ context. In comparison with node-centric browsers, paths provide a better context and help readers understand the information at the current node in relation to the underlying concept of the path.

In this paper, we argue that path-centric browsing is an independent browsing mechanism that combines the desired properties of graph-centric and node-centric browsing behaviors. We present an architecture for path-centric browsing over interconnected paths and describe the interface of the Path Engine, our prototype system. The Path Engine provides a linear interface, much like that of a subway carriage map, to navigate a complex network of interconnected paths in multiple simultaneous contexts.

This work expands the notion of paths in Walden’s Paths [27][28], a Web-based implementation of the path metaphor. Our earlier work has addressed issues involving the creation, presentation and management of linear paths. This paper discusses issues that arise when dealing with more generalized path structures.

The rest of the paper is organized as follows: Section 2 provides an outline of related work and a brief overview of our current
implementation of Walden’s Paths. In section 3, we discuss the nature of path-centric browsing. Section 4 presents an architecture for supporting more expressive path structures. This section also describes the design and implementation of a prototype of the Walden’s Path Engine. Section 5 discusses the implications of our support for generalized path structures and presents avenues for future work. Section 6 provides concluding remarks.

2. BACKGROUND
The notion of paths or trails overlaying a hypertextual network is as old as that of hypertext itself. Bush envisioned associative trails with possible digressive side trips as a personal means to remember and organize information for users of the memex and to communicate this information to friends and colleagues [4]. Paths have since been implemented in a variety of systems to serve different purposes.

Textnet paths provide multiple ways to linearize textual and organizational nodes in a semantic network [30]. Hammons and Allinson’s Guided Tours [14] are conditional paths and use a travel holiday metaphor to teach non-formal fields of knowledge. Scripted Paths [34][35] support conditional and programming paths with an automated path playback function. Paths within Guided tours [31], the path mechanism in Notecards, are constructed around a primary spine with digressive side trips [19].

The Perseus project [18] employs paths in an educational environment as a means to teach, think about or to analyze concepts. Storyspace allows hypertext writers to organize lists of cells that match specific criteria into named paths [2][16]. A network of these paths may serve as the index to an underlying complex hypertext structure.

The Webvise Guided Tour System [26], based on the Webvise open hypermedia system, provides tools to generate, edit and navigate guided Web tours using the metaphor of metro (subway) maps. Authors generate metro maps via an overview-based authoring interface. Travelers taking the tour may start at a central metro station or at a station of their choice and take the metro to other stations on the routes of their interest. Readers on a specific metro line get navigational assistance via linear carriage maps that show other stops along the line.

Walden’s Paths (http://www.csdl.tamu.edu/walden) [27][28], our implementation of the path mechanism, is a suite of tools that supports creation, presentation and maintenance [11] of Web-based linear paths. Path-centric browsing is the primary browsing behavior in Walden’s Paths. Typically, paths exercise little control over the structure, content or presentation of (components of) the Web pages to which they refer. Authors of paths locate materials relevant to a topic from the Web and organize them in a path. These Web pages may be oriented towards diverse user

Figure 1. Viewing a path in the Walden’s Path Server.
populations and may each address the topic from a different perspective. Authors of paths add HTML author annotations to each included page to cast these pages into a coherent presentation. The Web page along with the author’s annotation forms a unit of the path called a “stop”.

We have explored the use of Walden’s Paths in an educational setting, where teachers create paths to present curricular materials in a classroom environment. Walden’s Paths has espoused simplicity as a primary design issue for ease of integration into classrooms. Paths are linear lists of stops. Linear structures are the easiest to create, convey, comprehend and represent in both the system as well as the user interface. Soon after they start using the system, teachers and students alike shift focus from the underlying metaphor of the path to the subject matter presented by the path [12]. We attribute this “fading” of technology to the simplicity of the metaphor and of the interface.

Path readers typically begin browsing by choosing a path from a list presented to them. Figure 1 illustrates the navigation interface while browsing a path via the Walden’s Path Server. The interface displays the Web page that this stop includes in the context of the path. The annotation or contextualizing text added by the path author is displayed in the top right part. Readers navigate the path using controls displayed in the top right portion. They may view the pages on the path sequentially by clicking on the “Next” and “Back” images or jump to other pages on the path by scrolling to and clicking on the “paw” corresponding to the position of the page. While on the path, readers are free to follow links on pages to freely examine the information space. While the reader is browsing the information space of the path the control widgets described above are replaced by a single image that links back to the last page visited on the path, thus providing readers with an easy way to return to the path once they are done exploring.

3. PATH-CENTRIC BROWSING

Conklin referred to the problems faced while navigating large arbitrarily structured hypertextual nets as “cognitive overload” and “disorientation” in his survey of the early hypermedia systems [5]. Some hypermedia systems address these issues by restricting the structures that the hypertext may represent. Systems like Augment [8] and EDS [10] impose hierarchy on document structures and included support for hierarchical outline browsing. Large hierarchical structures can be easily managed by collapsing and expanding branches as needed.

Noted hypermedia systems like Intermedia [33], Neptune [8], and Notecards [31] support representation of complex hypertextual structures as networks or directed graphs. Network views containing more than a few nodes frequently overwhelm users [32] rendering static graph-based views ineffective for practical use. Various systems address this issue by providing additional support for navigation. Users may navigate the hypertexts via graph-centric browsers that provide graphical global or partial representations of the hypertext’s node-link network. Intermedia incorporates “Web Views” that present a user’s recent browsing activity and related documents in each of their earlier contexts. Neptune provides graph, document and node browsers that allow users to navigate and manipulate the structure and contents of the hypertext. Users of NoteCards navigate in a node-centric mode by following links from one card to another or in a graph-centric mode by navigating a map of the network via the special “browser” cards.

Because Web browsers only display the information (contents of a Web page) and the available actions (links that can be followed) from the current Web page they are node-centric browsers. Web users cannot view information on linked pages without following the links and receive little guidance about which links in particular will benefit them. Web pages created without any navigational aids for the readers force them to browse the Web in a pure node-centric fashion. Developers of most Web sites, however, provide navigation support mechanisms like static or contextual menus and site maps (simplified hierarchic views of the structure of the Web site) to ease browsing. The menus are typically available from all pages in a Web site and are either included in each page via templates or located in a separate frame. In the absence of this support, Web users may have trouble with navigation, finding information they had located earlier and may inadvertently revisit pages. Park and Kim report that contextual navigational aids help users browse the Web more efficiently [23].

Paths created intentionally by users (or compiled by the system and approved by users) provide a natural human context while browsing hypertexts. Paths are collections for supporting users’ personal mental models. They impose further organization over pre-organized materials without altering the existing structure of the information space, thus adding value to the system. While paths have been implemented in many hypermedia systems [2][14][18][19][30][31][34][35], most systems treat paths as one of the many features in their repertoire, an afterthought rather than a pivotal aspect of the design.

Walden’s Paths promotes paths as the primary navigation metaphor for Web-based materials. The interface of the Path Server displays pages chosen by path authors from various Web sites that they deem appropriate to the readers’ context of use. The ordering of the pages in the paths serves as a clear navigation directive. Thus the path-centric browsing behavior exhibited by Walden’s Paths clearly differs from either the node-centric or the graph-centric browsing behaviors. It affords a greater overview than node-centric browsing and a more focused view of the hypertextual net than graph-centric browsing. Paths are created for use in specific contexts and they provide excellent overviews of relevant information. More importantly, they occlude unrelated materials, thus aiding readers in focusing on relevant information within a much larger and diverse information web.

4. PROVIDING SUPPORT FOR INTERCONNECTED PATHS

The Path Server only allows browsing linear paths. Thus, path authors must express relationships between individual nodes in a single dimension. Relationships between objects (physical or conceptual) are rarely ever that simple; an object may link to many others along various dimensions. While simplicity of the Path Server interface is desired, the simple linear representation of the underlying structure restricts authors in expressing more meaningful relationships between the various concepts included in paths. In this section we present richer path structures that support expression of multiple relationships between stops, within or across paths.


### 4.1 A Scenario of Path Formation

To illustrate the more generalized concept of paths used within the prototype Walden’s Path Engine we will consider an example set of paths built using Web pages dealing with movies.

An instructor of cinema studies is seeking materials for use in her “Introduction to Film Studies” class. She erects the basic framework for her class by creating a path introducing various genres of film in the order in which the class will cover them. Each stop on this path links to another path (a sub-path) about that genre. Each stop on these sub-paths deals with a movie and may appear within different sub-paths if it fits into multiple genres, for example, a stop for the movie *True Lies* would be included in the sub-path for action movies as well as that for comedies. The instructor adds annotations to the stops pinning down the relevance of each of the movie Web page to the genre — a movie may be an instance of “schlock” horror and thus noteworthy amongst other horror movies, or the movie may be the first action movie to garner a total box-office gross exceeding $100 million.

Another instructor is designing a path for her course on films of a specific decade. Her path uses a different perspective than genre: the movies’ dates of release. Other paths might link together movies starring certain actors, or set in Bolivia, causing further intersections. Yet another instructor’s path includes stops about works of art that transcend media forms, for example, literary, theater and movie resources for Shakespeare’s *Hamlet*. Each of these instructors’ paths may share some stops used in the path created for the “Introduction to Film Studies” course, albeit in a different order, possibly with different annotations due to the differences in contexts of their use. Seen from each of these different contextual vantage points, stops take on different connotations.

To consider an example, authors of various paths may wish to elucidate some of the many contexts in which the epic 1959 movie *Ben Hur* can be viewed. An author creating a path on big budget films may focus on the production costs and grandiose sets used for the movie. Another author creating a path on Oscar-winning movies may annotate the stop with information about the various categories in which the movie won Oscars and its importance in being the all-time top award winner. An author creating a path about Charlton Heston’s movies may include information that describes Heston’s role in this movie and compare it with his roles in other movies. Similarly, authors creating paths on topics in other subject areas like “The Rise of Christianity” or “The Roman Empire” may point to *Ben Hur* and focus on its biblical references or the Roman treatment of Jews depicted in the movie. The various contexts in which a movie may be viewed are limited only by the author’s imagination and the attributes that she wishes to focus on.

Visually representing densely interconnected paths becomes increasingly difficult as the number of paths and thus the dimensions of analysis increase. Creating visual representations of all possible interconnections between various paths (also including other paths whose stops are linked in order of box-office gross, film studio, and so on) devolves into little more than a race to find methods to overlap the various lines of connection without smothering the stops themselves in the process.

Different paths may present a set of stops in differing order possibly leading readers to varying interpretations of the paths’ content. In hypertext, as in cinema, concepts are expressed by juxtaposition of content and the users infer coherence and meaning by linking these on the basis of their semantic content [17]. Readers of Joyce’s *afternoon, a story* may come away with radically different impressions depending upon the links they follow during their reading session(s) [15]. None of these impressions are right or wrong; they simply reflect varying views conveyed by a hypertextual story. Rosenberg describes how readers of hypertexts attempt to impose a narrative structure on episodes—combinations of users actions on the hypertext (actemes)—to form sessions of coherent activity [25]. In this sense the sessions are like poems, none of them are right or wrong in the absolute sense [20]. Authors may create paths that consist of scenes from films like *Pulp Fiction* or *The Usual Suspects* presented in sequences other than those in the movies to provide a different experience of these movies to their viewers.

Within metropolitan areas travelers ride subways every day and city transportation planners do manage to produce rail system maps. Different subway lines share stops, run alongside one another, and even cross paths occasionally. However, there are two important differences between subway lines and our contextual paths. The linkages between physical train stops are spatially oriented in the real world; train lines simply cannot take up more space than is available between the buildings, streets, sewers, and other objects. Due to their geographic grounding, railway routes rarely intersect quite so often as paths in our conceptual spaces do.

As we all too often learn upon first visiting a new city, subway maps themselves are often not particularly easy to use. They frequently appear as convoluted masses of lines and circles, presenting the complex interconnections between various lines without providing much clarity on the details of individual lines or stops. Examples of route maps available on the Web include those for the New York City subway,1 London’s Tube,2 and the Seoul Subway.3 (Copyright restrictions prohibit us from reproducing these images in this paper.) Much like graph-centric hypertextual browsers, as the number of nodes increases, information complexity becomes an omnipresent concern. While this view is complete and coherent it does not address the immediate needs of a person traveling on one of the lines. On the other hand, providing information about the traveler’s current location via a pure node-centric model would prove too limiting: merely knowing what the next stop is, without any awareness of what stops exist beyond that point would be inadequate. Urban transportation boards provide a solution oriented around the path-centric model. Once aboard the train a traveler need only concern herself with the stops visited along that line and perhaps at which of them she might transfer onto a different line. These route-specific maps are themselves presented in an easy-to-comprehend linear form, only providing information that a traveler is likely to find helpful.

---

1 http://www.nycsubway.org/maps/route/
2 http://www.thetube.com/content/tubemap/images/large_print_map03.gif
3 http://www.metropla.net/as/seou/seoul-map.htm
4.2 Architecture
The Webvise system addresses the issue of representing complex route maps in a top-down manner, starting with the overall structure of the tours and interplay between them [26]. Authors may create overview maps to a desired level of complexity via the provided graphical guided tour editor.

We take the bottom-up approach to creating complex path structures by defining the primitives and combining them to form higher-level structures. This approach is similar to that of ZigZag in which all structures are built from free-floating cells in an n-dimensional space [22]. The architecture separates the management of information elements and links, similar to the architecture of Sun’s Link Service [24].

4.2.1 Stops
A stop is a unit of information within the framework and represents some concept or idea as embodied by a given Web page. Stops support perspective-specific presentation of concepts by storing multiple information resources and annotations. Multiple resources allow the system to potentially select the optimal resource for each client’s display capabilities or to gracefully accommodate an atrophied or broken hyperlink.

Multiple annotations within each stop provide a mechanism to better contextualize a stop within different paths. Each path that includes the stop can either reuse the stop’s resource and the default annotation or provide a path-specific annotation. This enables path authors to reuse resources entered by others and to
implicitly delegate the responsibility for keeping these resources up to date to another author.

To use a specific example, consider a stop describing the living habits of elephants both in the wild and in the zoo. Such a stop is relevant within both a path about large animals or one exploring the different animals one might find in a zoo. The context within which we encounter the stop and annotations upon the stop will provide a framework within which a reader may approach the topic.

### 4.2.2 Links
Links connect stops to other related stops. These links “know” what path they begin from and where they lead — and in this way the links themselves provide directionality and a means to navigate within or across paths. While these links are directed, they can be followed backwards.

Sun’s Link Service [24] leaves the definition of the links’ endpoints to the integrating applications. Links may point to nodes or to specific end points (identified by markup) within the nodes. Microcosm [6] supports linking to locations within nodes without embedding markup in the data nodes. To use the terminology defined by the Dexter hypermedia reference model [13] our architecture supports component-to-component links. The links are anchored to opaque stops; they do not anchor to spans within stops. Links store their source and destination stops, as well as the source and destination path. Thus, links may lead from a stop on one path to a stop on another path just as easily as between two stops on the same path. As paths are considered to be primitives, our system treats these links as the basic “untyped” links. Conklin calls these “Referential” links [4], while DeRose’s taxonomy refers to them as “Relational links” [7].

The system provides support for adding types to links by embedding attribute-value pairs into link objects. While the architecture does not restrict the number of types a link may have, we currently implement only a “hierarchic” type. These links embody Conklin’s “Organizational” links. Links of this type point to the head of a path expounding upon the content of the stop it originates from. They can be understood as an opportunity to “learn more about” the concept explored within a given stop without the leaving the present context of the reader’s path. In DeRose’s taxonomy these links are called “Sequential links”. When these links are followed, the system retains a record of the digression and allows the reader to return immediately to her place in the former path. In contrast, a regular link leading to a stop on another path may just as readily represent a chance to wander across paths and visit sub-paths to learn more about topics of interest. To avoid having to load the entire path network a priori we use client-server communication to enable real-time responsiveness by retrieving information about the current stop “just-in-time” as the reader navigates along a path.

A reader navigating a path may continue to follow links to further stops along the same path or if the links allow may elect to change to a different path. These cross-path jumps can either represent an expansion (a hierarchical linkage) of a given concept, with the understanding that she may wish to return to the earlier context, or a complete shift of context to a new unrelated path.

Figure 2 illustrates a scenario in which a student enrolled in the “Introduction to Film Studies” class is browsing the class material that her instructor has prepared. The top left part of the interface contains widgets to help her navigate the path structure. The student is currently browsing a path titled “Action Movies”, presently on the second stop along that path, and that the path consists of 15 stops.

### 4.2.3 Paths
Where stops may be said to be concepts, paths are contexts running through these concepts. The structure of a path is specified through the links that connect a set of stops together. When two paths share stops, the paths are said to be intersecting. The architecture does not prohibit or promote any particular structures.

Stops, links and paths are assigned globally unique identifiers when they are created so that any individual object can be traced back to its originating server as needed. This feature allows paths to include path fragments that are distributed across a network.

### 4.3 Design and Prototype of the Path Engine
The architecture does not restrict the expression of any path structures and paths could, in theory, be expressed as general directed graphs. The Trellis system provides a general mechanism for expressing paths through documents and their components [29]. Guided Tours in Notecards are paths built around “spines”, which present the main concept, and permit “side trips” to pursue interesting digressions or explore materials in greater detail [19].

Our experiences with Walden’s Paths [28] have suggested that linear paths with sub-paths that depart from the various nodes of the main paths are sufficiently expressive, yet easy to work with. Walden’s Path Engine is designed to support browsing of densely interconnected linear path structures with sub-paths (expressed by hierarchical links) branching off from them. The sub-path is a path in its own right and may contain other nested sub-paths. This hierarchical structure containing linear sub-paths is further complicated when we consider that stops on the sub-paths may also connect in unpredictable ways with stops on paths and other sub-paths. While the underlying path structure may be more complex than the subway map shown in figure 2, the Path Engine retains the proven linear interface of the Walden’s Paths Server to shield users from the complexity of the intricate underlying network of paths. The Path Engine borrows the notion of using linear simplification of relevant information in complex structures from the subway maps but it does not employ the metaphor of “taking the subway”. The interface supports navigational aids that are essential when browsing conceptual spaces (especially if the users have a goal) that are not analogous to problems faced by users of subway systems.

The non-intersecting linear paths supported by our earlier Path Server are easily represented in files. To better support non-linear and complex structures, the Path Engine stores the stops, links and paths in a relational database system. This permits the system to readily determine membership relationships across paths and stops via simple queries. Readers of the Path Engine may easily wander across paths and visit sub-paths to learn more about topics of interest. To avoid having to load the entire path network a priori we use client-server communication to enable real-time responsiveness by retrieving information about the current stop “just-in-time” as the reader navigates along a path.

Figure 2 illustrates a scenario in which a student enrolled in the “Introduction to Film Studies” class is browsing the class material that her instructor has prepared. The top left part of the interface contains widgets to help her navigate the path structure. The student is currently browsing a path titled “Action Movies”, presently on the second stop along that path, and that the path consists of 15 stops.
Beneath the path’s title a menu lists available destinations proximal to this stop. The reader has the opportunity to continue along to the next stop within the “Action Movies” path or to change to one of two different non-hierarchical paths: “Movies of the 1980s,” and “Mel Gibson Movies.” These paths begin elsewhere and intersect the “Action Movies” path at this stop. Had a hierarchical link been available the title of that sub path would be listed within the menu under an explanatory banner explaining that she can “Learn More About This”. In addition, the menu provides a link for the reader to continue along the present path.

The “Continue Earlier Path” option visible to the right of the logo informs the reader that she has reached this path by following an earlier hierarchical link. To recall the information about the earlier context in which she had been browsing, she lets the mouse pointer hover over the available “A” link. Information about the earlier path that she was browsing, and may return to, is displayed in a tool tip. In this case it was a path about “Movie Genres.” Should she progress to another hierarchical link and follow it to a successive sub-path a “B” link will appear beneath the “A”.

The stops succeeding or preceding her current location in the path are all available for her to visit immediately. Furthermore, the line joining the numbered stops extends past the 5th stop to emphasize that the path continues for more than 5 more stops. Clicking on the “Next 5” button will bring up stops numbered 6 through 10. The line joining these would extend past the 6th stop to the left along with a corresponding “Prev 5” button. At any point the reader may skip to a stop further (or earlier) along the present path, by clicking on the button corresponding to the stop.

To the right of the operating panel we can view the author-provided HTML annotation for this stop within this path’s context. If the user switched to the “Mel Gibson Movies” path, a different annotation would appear in this panel, presumably one that puts this film in the context of other Mel Gibson films. In addition, the operating panel would change to display the options available to the reader from that path.

The reader is free to wander off the path entirely by clicking on hyperlinks within any Web page displayed in the lower part of the window. Upon doing so, the navigation widgets will be replaced by a single prominent “Return to path” link. When the reader is done exploring the Web, she may click on this link to restore the system state from which she initially wandered off the Path Engine’s domain. This is performed, as in the earlier Path Server, by using client-side browser scripting and dynamic HTML operations. Thus, a reader retains complete control over her browsing experience and may freely explore concepts not contained within the realm of stops.

Looking more closely at the numbered buttons available to the reader, stop 5 along this path is visually distinguished by a button with a border to indicate that a hierarchical link is present at that page. Pages with further information available in the form of sub paths may thus be readily identified without the need to manually progress through the intervening stops. The interface also realigns the current stop to the central position, if possible. Thus the Path Engine provides a moving window around the current stop, allowing the user to visit nearby stops via a single mouse click.

5. DISCUSSION AND FUTURE WORK

The Path Engine prototype does not allow for branching paths, although the underlying architecture can accommodate them. This apparent limitation has several benefits. It enables the system to determine the length of the paths and thereby, provides readers an idea of their location within the current path as well as in all other contexts (earlier paths) as they navigate.

Furthermore, branches can be simulated within the prototype’s capabilities. Figure 3 illustrates a simple case where a branched structure (a) can be converted into a form (b1) or (b2) supported by the Path Engine by converting one of the branches into the spine and creating a new gray path with the gray-filled stops as its end points. Some structures and applications inherently require support for more complex representations than this prototype supports. For example, features like the conditional links in StorySpace [3] are not supported in the current prototype. Using conditional links in educational paths will enable teachers to automatically guide students from various grade levels to different branches of a path instead of writing different paths for each grade level. Zellweger argues that it is easier to manage and present fewer, but more powerful, paths than to manage a large forest of simpler paths [35]. Bernstein presents a variety of hypertext patterns found in literary hypertexts [1]. Simulating some of these structures for example, the Counterpoint, the Contour or the Mirrorworld is likely to test the skill and patience of most authors when using this system. We are exploring mechanisms to support easy authoring and presentation of rich patterns for literary hypertexts.

Providing comprehensible global views of these high-dimensional and highly interconnected structures is a non-trivial task. We are presently exploring methods to visualize these structures effectively and also assessing what types of views will be most useful to readers and authors. Having means for visualizing these structures is also crucial to the development of intuitive visual environments for path authoring. Trellis [29] and Webvise [26] provide simple and elegant 2-D interfaces to support the creation of directed graph structures. The simplicity of these interfaces unfortunately also limits the expressivity and size of the structures that may be created. caT (context-aware Trellis) [21] addresses this issue by providing hierarchically organized 2-D authoring environments.

Figure 3. Simulation of simple fork and join structures.
spaces. We are also investigating appropriate paradigms and metaphors for designing usable authoring environments.

We are also identifying issues related to creating and browsing paths whose components are distributed across networks. Our studies of the authoring practices employed by teachers with Walden's Paths found that they often created short paths — most containing fewer than 15 stops [27]. Path fragments, such as these, potentially authored by teachers at different locations can be shared across the network to reduce the time needed for creating new paths.

6. CONCLUDING REMARKS
Nodes in large hypermedia networks may be related to each other in more contexts than are made explicit by the default organization of the network. Paths manifest these implicit connections for author-specified contexts and add value to these networks by performing as another layer of organization over the pre-structured materials. Path-centric browsing, the browsing behavior embodied by Path interfaces, has two major advantages over other navigation strategies. Path overviews tend to include more focused information and hide the irrelevant information from the users, thus helping users to focus on the intended activity. They also present nodes in the immediate context of their neighboring elements, as well as the larger context of the entire path. While paths have been incorporated in hypermedia systems for decades, they have been viewed as supporting tools rather than as primary navigation mechanisms.

Continuing the work begun by the Walden’s Paths project, we are exploring strategies for specification, presentation and effective use of more expressive path structures. Over time, the paths created to connect the nodes may repeatedly intersect to form dense networks over the nodes in the underlying hypermedia system. Path intersections represent the meeting points between the various concepts and contexts they represent. In this paper we have presented an architecture to support the expression of structurally rich, deeply interconnected paths. The inherent problem in presenting complex connections is effectively displaying them. Our prototype system, the Path Engine, uses a simple linear display strategy, somewhat similar to that used in subway cars to present only the most relevant information at any time. As our investigation into the various issues progresses, the results will be used to improve the architecture and the prototype presented in this paper.

7. ACKNOWLEDGMENTS
This material is based upon work supported by the National Science Foundation under grant numbers DUE-0085798, DUE-0121527 and IIS-0219540.

8. REFERENCES


and Hypermedia, Hypertext ’00 (San Antonio TX, May-June 2000), ACM Press, 236-237.


