Publishing Evolving Metadocuments on the Web

Andruid Kerne, Madhur Khandelwal, Vikram Sundaram
Interface Ecology Lab | Center for Digital Libraries
Texas A&M Department of Computer Science
College Station. TX, 77843. USA.
+1 979.862.3217

{andruid, madhur, vikram}@tamu.edu

ABSTRACT

Metadocuments are documents that consist primarily of references to other documents, and elements within them. Our active browsing web visualization tool generates an evolving series of navigable metadocument snapshots over time. The granularity of browsing is shifted, from documents to the finer grained information elements, which are metadocument constituents. The program conducts expression-directed automatic retrieval of information from the web. It performs procedural visual composition of the information elements to form spatial hypertext. The user can express interest and design intentions through direct manipulation interactions with the visualized information elements. As prior versions of the tool lacked the capabilities of save and load, they were entirely process-oriented. The metadocuments existed only as transient states. This paper is an early report on our new metadocument authoring and publishing capability, and its potential uses. Saved metadocuments can be published on the web. Once published, they can serve both as static navigable metadocuments, and as the jumping off point from which the information space represented by the collected elements can continue to evolve.

General Terms: design, human factors,

Keywords: metadocuments, collections, spatial hypertext, adaptive hypertext, navigation, procedural visual composition

1. COLLECTING INFORMATION ELEMENTS INTO METADOCUMENTS

Users of hypertext often need to collect references to significant places that they encounter while browsing. These collections are metadocuments. They are documents that consist primarily of references to other documents, and elements within them [2]. They consist both of references by name, such as image elements and hyperlinks, and by value, in the form of textual quotations. We call the image references and quotations *information elements*. For each information element, in addition to any embedded hyperlinks, there is always an implicit reference back to the original document, which we call its *container*. Schraefel articulates the importance to metadocument authors of the connection between an information element and its container [7]; when we collect information elements from the web, we want to be able to easily return to the sources of the quotations. Each information element can be thought of as an enhanced bookmark.

2. EXPRESSION-DIRECTED GENERATIVE BROWSING

Our program [3] generates collections of information elements automatically, over time. The process starts with seeding. Content from N initial seed URLs is downloaded. These may refer to static or dynamic web pages, including search engine queries. The seed content is decomposed into information elements.

The program builds a model of the structure of the information, and of the user's interests [5]. The graph structure of the web is represented, along with a term-based textual index of information elements and document references. Each node includes attributes that record the user's expressions of interest.

The program automatically, periodically engages in several dynamic operations, based on the model. It chooses information elements to display, document references to crawl to and download, and regions of the screen for the placement of successive elements.

Once elements are onscreen, the user may use the positive grab tool to express interest, and the negative grab and cut tools to express disinterest [4]. These expressions are propagated through the model via spreading activation [6].

3. VISUAL COMPOSITION

Procedural

The program performs procedural composition, according to the model of user interests, as information elements enter the spatial hypertext (See figure.). Each new element covers up the region of least importance. Elements are successively transformed. Alpha gradients create visual connection between juxtaposed elements. Desaturation is performed as elements age. Negative elements are blurred. A result is layering that facilitates overall legibility. The use of procedural visual transformations differentiates the program from other spatial hypertext tools such as VKB [8].

Direct Manipulation

Our tool also allows users to engage directly in visual and semantic composition. The cut tool allow users to remove information elements they are not interested in. Grab allows them to arrange elements in ways that express organization. Resize enables the creation of visual hierarchies. Another tool enables users to navigate from the spatial hypertext collection visualization back to browsing the container or hyperlink associated with an information element, in a traditional browser.

Metadocument tools since Bush's Memex [1,2,7,8] have realized that authors can add meaning to their collections through annotations. Our program enables annotations through its text tool. Annotations are a special form of information element, because they may not be directly associated with another document container. In our implementation, they are automatically collected in a special user session container, as well as in the metadocument itself.

HT'03, August 26-30, 2003, Nottingham, United Kingdom.
ACM 1-58113-704-4/03/0008.

Procedural visual layering in a metadocument referencing Perseus and International Children's Books digital libraries.

4. SAVE | LOAD

Saving the state of our metadocuments requires, at a minimum, saving the visualized information elements, their containers, and the attributes of the model that represent the user's interests. The set of containers is emitted first, as a convenience for the program, so that it will be available during the load process, when the information element representations are parsed.

As is standard practice, we are using XML to save metadocument state. Generating the structures of nested and linked objects is a bit cumbersome for programmers building XML DOM trees from the object oriented data representations that programs use, and especially for the reverse translation. By imposing conditions that procedurally map the names of XML elements and attributes to Java objects and instance variables, and using Java's reflection API, we have created a general, extensible framework for automatically conducting these transformations. We intend to detail the inner workings of this framework in a future publication, and to open source the code.

Some surprising complications arise in regenerating the visual composition from the saved metadocument. The fact that the web is unstable and that the containers and images may suffer 404 not found errors, is to be expected. What is more complex is that, since download operations need to proceed in parallel, elements may not arrive in their original visual stacking order. Currently, we just let attributes from the model sort the stacking order out dynamically, as it would in the course of the program's usual, one element at a time, operation.

5. DOWNLOAD | PUBLISH

After saving session metadocuments, user/publishers can easily move them onto web servers. This is equivalent to what authors of regular HTML documents do when building a site. We have added the capability to retrieve recombinant metadocuments via http, instead of from the local file system.

What was left then for us, was to add a publisher's mechanism that enables integration of these metadocuments into regular web sites via the standard `href` tag. The mechanism consists of a new, fixed information space seeding page [5]. References to it include the URL of the published metadocument, and other optional metadata, as arguments (Example: `http://csdl.tamu.edu/collagemachine/launch/regenerate.html?metadocument=http://mrl.nyu.edu/andruid/perseusIntlChildrens.xml&start=play`). The user is brought to a standard startup page, which offers options such as the size of the metadocument visualization window. The URL arguments and the options, which are processed via client-side JavaScript, are among the runtime parameters passed to an applet.

Evolving Metadocuments

An optional publisher's runtime argument determines whether the published metadocument is started as dynamic and evolving (playing), or as initially static (paused), giving the user time to see the visualization exactly as designed by the previous metadocument author. In either case, the user can use the tape recorder metaphor controls to switch between active play and static paused. The user has access to the same expression and design tools as the original author, and can save and publish her own versions of the metadocument. One potential use is to enable publishers of large digital library collections to offer specific navigable evolving views of their libraries. Users can derive and share their own views. Another is for collaborators to collect and share compositions of significant materials.

6. CONTINUING WORK

A focus of our work during this cycle is to give metadocument authors and experience-oriented browsing users more control of information element retrieval and composition. One part of this has to do with the identification and retrieval of related information in response to the user's expressions of interest. We continue to integrate techniques from information retrieval and computational linguistics. Another aspect has to do with enabling information to be brought into the design space via drag and drop. We are working on modifying Mozilla, so that information elements dragged into a metadocument state will include container metadata.

7. REFERENCES

[1] Bush, V. As We May Think. Atlantic Monthly, July, 1945.

[2] Furuta, R. Shipman, F., Marshall, C. Brenner, D., Hsieh, H.. Hypertext paths and the World-Wide Web: experiences with Walden's Paths. *Proc ACM Hypertext*, 167-176, 1997.

[3] Kerne, A. *CollageMachine*. http://csdl.tamu.edu/collagemachine

[4] Kerne, A. CollageMachine: Interest-Driven Browsing Through Streaming Collage, *Proc Cast01: Living in Mixed Realities*. Bonn, Sept 2001.

[5] Kerne, A., Sundaram, V. A Recombinant Knowledge Space, *Proc Cosign 03: Conf on Computational Semiotics*, in press.

[6] Pirolli, P., Pitkow, J., Rao, R.. Silk from a Sow's Ear: Extracting Usable Structures from the Web", *Proc CHI 1996.*

[7] schraefel, m.c., Zhu, Y, Modjeska, D. Wigdor, D., Hunter Gatherer: Interaction Support for Creation and Management of Within-Web-Page Collections. *Proc WWW 2002.*

[8] Shipman, F., Hsieh, H., Airhart, R., Maloor, P. Moore, J.M., The Visual Knowledge Builder: A Second Generation Spatial Hypertext, *Proc ACM Hypertext*, 113-122, 2001.