Toward a Comprehensive Model of the Content and Structure of, and User Interaction over a Web Site

Anindya Datta[*][†]
Debra VanderMeer[*][†][‡]
Krithi Ramamritham[§]
Sham Navathe[†]

May 22, 2000

Abstract

As the usage of the World Wide Web grows, so does the complexity of web sites and the range of products and services offered on the web. in particular, ebusinesses are interested in tying context-sensitive web services (e.g., services that are cognizant of who a user is or what he is doing on the site) to internal data sources, e.g., information stored in customer profiles or purchase histories, in order to enhance the web site with such functionality as personalized or customized web pages. These companies are also interested in web knowledge discovery with respect to users, e.g., mining user navigation patterns, in order to improve site organization or discover new marketing possibilities.

These new requirements of web sites are all dependent on the underlying representation of the web site, or the state of a user session at the site, or both. In short, these requirements reveal the need for a model of a web site that incorporates not only a representation that supports dynamic page generation, but also the notion of user interaction with the site – all working toward a model of user behavior on a web site.

Existing web site models have been developed either with a specific application in mind, or to support entirely novel page generation and site management strategies. Most importantly, a model of user interaction with a web site is absent in the literature. This paper presents a model of a web site that includes not only the site content and organization, but also user interaction with the site.

[*]Chutney Technologies, Atlanta, GA, http://www.chutneysystems.com
[†]Georgia Institute of Technology, Atlanta, GA 30332
[‡]Contact Author: Debra VanderMeer, 430 Tenth Street, NW, Suite S-105, Atlanta, GA 30318, Tel.: 404-685-8520, email: deb@cc.gatech.edu
[§]University of Massachusetts, Amherst, MA 01003 and IIT, Bombay

1 Introduction

The World Wide Web is becoming a ubiquitous presence in our daily lives – even the little label on a bunch of bananas lists a URL these days. The expanding role of the web in society is leading to the explosive growth of the web across various dimensions: the number of users accessing the web, the number of sites available on the web, and the size and scope of individual sites. In this paper, we are primarily interested in this last issue, the increasing complexity of an individual web site as its size and scope grows.

In the early days of the web, sites tended to be small, and constructed on an ad-hoc basis by hand or perhaps with an HTML editing tool. Now, however, technologies for web site development have evolved to the point where a large team of developers, each with his own specialization, is needed to launch a commercial-quality web site.

One important shift in the area of web site development is the move from static pages to dynamic page generation, in which web pages are created on the fly (using tools such as ASP [7] or JSP [14]) as user requests arrive. Here, the standard model of a site as a set of pages no longer applies, since the space of possible dynamically-generated pages is literally infinite, given the virtually infinite content space.

Along with the growth of complexity of a web site is a concomitant increase in the functionality required of a web site. E-businesses are interested in tying context-sensitive web services (e.g., services that are cognizant of who a user is or what he is doing on the site) to internal data sources, e.g., information stored in customer profiles or purchase histories, in order to enhance the web site with such functionality as personalized or customized web pages. These companies are also interested in web knowledge discovery with respect to users, e.g., mining user navigation patterns, in order to improve site organization or discover new marketing possibilities.

These new requirements of web sites are all dependent on the underlying representation of the web site, or the state of a user session at the site, or both. In short, these requirements reveal the need for a model of a web site that incorporates not only a representation that supports dynamic page generation, but also the notion of user interaction with the site – all working toward a model of user behavior on a web site.

Even though web site modeling has been of recent interest, and a few interesting models, (e.g., [2, 9]) are available in the literature, these models have been developed either with a specific application in mind (e.g., deriving and restructuring hypertext content [2]), or to support entirely novel page generation and site management strategies (e.g., declarative specification of web sites [9]).

While application-specific models are indeed useful, and newer and more meaningful site management strategies are required, a unified model that works in the context of existing dynamic page generation strategies, and is application independent would be nice. Such a model would potentially be useful to both the research as well as the development communities. For researchers, such a model would be the "glue" that enables the envisioning of new scientific methods (such as new web-mining strategies), while still allowing those methods to be applicable in the context of existing web and application servers. For developers, such a model might form the basis of the development of new applications on top of existing web infrastructures. This paper presents such a model; in particular, we present a model of a web site that includes not only the site content and organization, but also user interaction with the site.

The remainder of the paper is organized as follows. We discuss related work in Section 2. In Section 3, we present a model of web site structure and interaction. In Section 4, we discuss the utility of our model, including an example of an application area that has the potential to benefit greatly from our model. Section 5 concludes the paper.

2 Related work

One can identify two distinct streams of research in web modeling: (a) work that considers the entire World Wide Web, and models the web as a whole. Notable pieces of work in this area are [13] and [10], both of which present web data models aimed at allowing declarative querying of the web as a whole for the purpose of enhancing search engines; and (b) models that address an individual web site. The latter of these is the focus of our work.

Some interesting models are available in the literature. However, these models have been developed either with specific applications in mind, or to support entirely novel page generation and site management strategies.

The model in [9] was developed to support new methods for the declarative specification of web sites. A logical model of the content and structure of a web site is presented. Here, the site structure is modeled as a schema, and a page on the site with a page schema. At run-time, web pages are created by accessing site content, stored independently in an underlying database management system, by means of declarative specification. A few notable features of the paper are: (a) a novel, entirely different dynamic page generation mechanism is presented; and (b) it lacks the notion of user interaction with the web site. Our model, in contrast, is consonant with existing and widely used dynamic page generation strategies, and includes, as an essential component, the notion of user interaction with the site.

The model in [2] also uses the notion of a page scheme to define the structure of a site. Rather than generating pages based on a schema (as in [9]), this model was developed to support deriving the structure of a page from HTML text, for the purpose of querying hypertext data and restructuring the query results into new hypertext. Like [9], this work also presents a content and organization model, but not a user interaction model.

In [1], the authors model the organization of a web site with an object-oriented structure called a hypertree, and a set functions that map actual pages (i.e., URLs) to the hypertree. The model presented in [16] presents a logical model of a web site's structure, as well as a presentational model for web content, developed to support a hypermedia application development tool. Again, these models in these papers consider content and organization, but not user interaction.

While application-specific models are indeed useful, and newer and more meaningful site management strategies are required, a unified model that works in the context of existing dynamic page generation strategies, and is application independent would be valuable. Such a model would include not only the site content and organization, but also user interaction with the site. This paper is one of the first in attempting to describe such a model.

3 Site and Interaction model

In this section, we describe a high-level view of a model of a web site and user interaction with such a site. Essentially, a web site consists of *content*, i.e., information displayed to the user upon request, and *structure*, i.e., the organization of the site. A user interacts with the site by making *actions* on the site, i.e., submitting requests to the web server, and viewing the site's *reaction*, i.e., the site's response. Figure 1 shows a high-level view of this interaction. Our model attempts to capture the relationships among the structure, content, and user interaction on a web site.

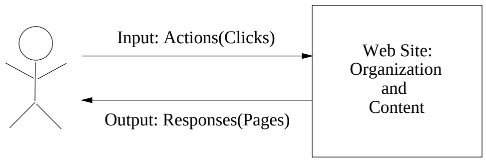

Figure 1: High-level view of site and interaction model

Having given an overview of our model, we now describe each part in detail. We begin with a discussion of the underlying site structure.

3.1 Site model

Fundamental to our site model is the notion of a *catalog*. We regard all web sites as consisting of a catalog, an abstract structure which provides a means of organizing web content on a site. In many cases, particularly for e-commerce-oriented web sites, the hierarchical catalog metaphor is appropriate; the content is naturally organized into categories and subcategories. For example, an e-shopper at amazon.com can browse through the Book category, into the Art subcategory, and from there to the Museum sub-subcategory, where, perhaps, he will purchase a new coffee-table book of paintings by Monet.

Even non-ecommerce sites (i.e., those not specifically selling a set of products) frequently display some form of hierarchy. Consider an online news site, such as the Washington Post's online site [5]. Here, we see a home page and several categories of news, e.g., World, Nation, Metro. While the home page provides links to a small number of top news stories in each category, the majority of content is organized through the site's hierarchal structure. As as entirely non-commercial example, let's consider a personal home page, say, for an academic. The home page shows an image, and provides some basic information. Categories linked from the home page are likely to include Research Interests, Projects, Publications, and Teaching. In fact, it is difficult to come up with an example of a site of any complexity that fails to exhibit some sort of hierarchical structure.

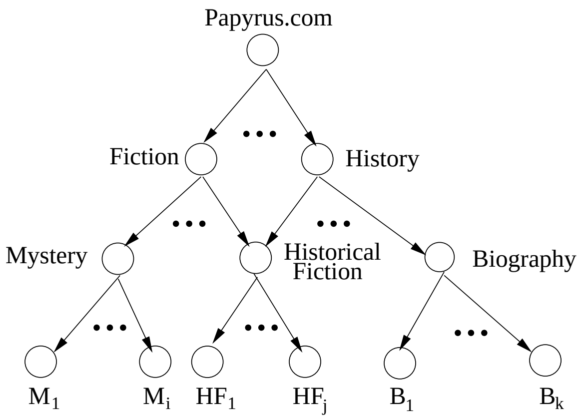

Figure 2: Catalog example

Figure 2 shows an example of a simple catalog for a fictitious online bookseller, *Papyrus.com*. Here, the root of the hierarchy is the *Papyrus.com* home page, which sells several categories of books, including *Fiction* and *History* books. These categories themselves have subcategories, e.g., *Mystery*, *Historical Fiction*, and *Biography*. These subcategories each represent a set of books, e.g., the *Historical Fiction* subcategory represents the set of books $\{HF_1...HF_j\}$.

At the time of a web site's creation, the site's catalog is *explicitly defined*, i.e., all nodes and edges between the nodes are explicitly defined. This provides a simple, yet concrete model of a web site, upon which we can build a discussion of user navigation on the site. We refer to the model of the explicitly defined catalog as the *Simple Catalog Model* (SCM). Figure 2 pictorially depicts the SCM for the *Papyrus.com* site.

We hasten to add, however, that the SCM does not exhaustively define the possible navigations at a site. For instance, given that the user is at the *Fiction* node, it is not necessary that he can only navigate to the nodes reachable (in a single step) from *Fiction* in the SCM. It is entirely possible that the user travels to a node *not* reachable in a single step, e.g., by using the browser's history list. In fact, possible user actions on an SCM define an *Extended Catalog Model* (ECM) that encompasses all possible navigation. We now proceed to present the concepts defined above, i.e., the SCM, the interaction model, and the ECM. To this end, we define the fundamental notions of the SCM, the catalog and the node, more formally:

Definition 1 Simple Catalog Model. *A Simple Catalog Model is nothing more than a graph $G = (V, E)$ in which nodes $v \in V$ represent the actual content of the site, and edges $(u, v) \in E$ represent the organization of the site explicitly defined by the site creator, i.e., the links between nodes. Intuitively, a node v in this graph represents some concept, perhaps an item or category in a product catalog, or a section or subsection of a FAQ.*

Definition 2 Node. *A node is a generalized, abstract structure which serves as the foundation for the catalog model. A node $\mathcal{N}$ is defined as a 3-tuple $\langle L, E, D\rangle$ of descriptive information where:*

- **L** *is unique label, i.e., key, for the node.*
- **E** *is a set of* edge labels*, i.e., labels of nodes $v \in V$ (where v is not the current node), reachable in a single step from the current node.*
- **D** *is a descriptor for the node, containing pointers to objects needed to represent the node in HTML, e.g., links, images and text that describe a the node. These objects are* **elements***, i.e., atomic static items that can be displayed on a web page. There are three basic types of elements (based on [2]):*
  - *An* **information element** *is a text, audio, or video object.*
  - *A* **link element** *is a hypertext link. These links have two parts: (A) a text description of the link destination, and (B) a reference to one of the following:*
    1. *Another node in the product catalog.*
    2. *an external (i.e., outside the web site domain) URL.*

    These link elements correspond exactly to the set of edges **E** *for the node, and provide information to the web server about how to present the edges on a web page to the user. In other words, edges denote the graphical structure of the site, while link elements provide the means to reveal these edges as hypertext links on a web page.*

– *A* **form element** *is a portion of a form for user entry. Form elements are collected into* forms*, which are used to facilitate user information entry (e.g., credit card information, customer service requests, etc.).*

To clarify the notion of a node, we present an example. Consider a node for a particular Mystery book. Here, the node's label **L** might be M_i, a unique identifier for the node. The node's edge set **E** might contain node labels for books by the same author, or bestsellers of the same genre. The node's descriptor might contain pointers to several information elements, e.g., a text element containing the book's title, and an image element containing a picture of the book cover. It will also contain pointers to the hypertext link elements corresponding to the edges in its edge set **E**. It might contain pointers to several form elements which will allow users to rate the book, and enter specific comments about it.

Note that the catalog model presented above ascribes an important property to an e-commerce site hosting the catalog. That is, through the notion of *elements*, the model supports the notion of *dynamic web pages*, i.e., pages that are not defined statically using HTML, but rather created on demand by putting together a number of elements on the fly. As the newer web servers and browsers come equipped with dynamic content generation technologies such as ASP, JSP, it is imperative that any meaningful model of an e-commerce site be able to support dynamic page generation.

The above description the SCM provides a basis for our discussion of user navigation through a site, as well as our ECM, both of which are described in the next section.

3.2 User interaction with the web site

User interaction with a web site consists of both input, i.e., clicks on a site, as well as output, i.e., pages returned in response to user clicks. We provide a model of user input in Section 3.2.1 and a model of site output in Section 3.2.3.

3.2.1 User Input: Action

We can think of a user interacting with a web site as a user navigating over the catalog, since the web pages are simply presentations of different views over the catalog. Thus, intuitively, a user can be said to be *located* at some node in the catalog throughout his visit to the site. For instance, assume that a user's i^{th} action at a site is a click on a hypertext link. Then, he is said to be located at the node pointed to by the link chosen in his i^{th} click. Thus, if the user is located at the Fiction node before his i^{th} click, and chooses the link to Historical Fiction as his i^{th} click, then after the i^{th} click he is at the Historical Fiction node.

Clearly, we do not wish to limit our model of user behavior at a web site to simple navigation. Rather, we would like to include other types of behavior, e.g., making purchases. Thus, we generalize the navigation metaphor to a more abstract notion of an *action*, and provide a formal definition of an action on a web site:

Definition 3 Action. *An* action $A \subseteq \mathcal{A}$*, where* $\mathcal{A}$ *is entire space of possible actions on a catalog, is a permissible choice presented to the user at a node. We define six possible actions on a catalog:*

1. *A* **navigation** *action is simply a click on a navigational link. For example, if user* U *is at the home page of an online bookseller (say, Papyrus.com in Figure 2), and wishes to see more information about Fiction, he would undertake this navigation action by clicking on the* *Fiction* *link.*

2. *A* **buy** *action is a click indicating the user's intention to buy an item. On an e-commerce site, this occurs when a user chooses to place an item in his shopping cart. Clearly, this action is available only from nodes which offer items for purchase. Further, an item is only available for purchase from the node that represents it, e.g., the action of purchasing Mystery* M_i *is only available from node* M_i*.*

3. *An* **un-buy** *action occurs when a user removes an item from his shopping cart, in effect undoing a previous* buy *action. This action is available at any point after the user has selected an item for purchase.*

4. *A* **check-out** *action occurs when a user completes the purchase of the items in his shopping cart, thereby actually creating a purchase transaction in the system. This action is available at any point at which the user has at least one item in his shopping cart.*

5. *A* **form-submit** *action sends user-entered information to the web server. This type of action can be* optional *(e.g., a user can choose to respond to a survey), or* mandatory *(e.g., the user must enter credit card information and a shipping address in order to complete a* **check-out** *action, or enter a logon and password to enter a protected site).*

6. *A* **departure** *action occurs when a user departs from the site. Note that an e-commerce site's web server cannot explicitly detect a user's departure, since the HTTP request goes to another site's web server. Typically, a user is considered to have departed a web site after some threshold amount of time has elapsed since his most recent click, and thus, departure can be inferred.*

A subtle, but important property of the above-described action model, in combination with the site model described in Section 3.1, is that it confines a user's possible movements within a site across nodes of the site's catalog (or course, it also models the fact that a user can depart). In other words, the model mandates that at any given time during his visit, a user is on a given node in the catalog, and any action on his part will result in one of the following three results: (a) he will remain on the same node (if, say, he places an item in his shopping cart), (b) he will move to a different node (by clicking on a link), or (c) he will depart the site. Note, however, that the model does not imply that only a single node can be displayed on a page delivered to a user. Rather, the descriptor component of a node, say X, may contain references to other nodes, say I and J. Thus when the user navigates to X, the contents of I and J (or some specified subset of the contents) will be displayed, along with the contents of X.

3.2.2 Extending the Site and Interaction Model

Clearly, the site and interaction models described in Section 3.1 and above allow only for explicit browsing of a site, where a user navigates through the site based on explicit links. For example, a user entering the *Papyrus.com* site (depicted in Figure 2) looking for the Mystery book M_x will need to navigate through the *Fiction* and *Mystery* nodes in order to find his book of interest. This type of navigation on a web site does not reflect the multitude of ways in which a user can move through a web site. We briefly consider a few possible alternative browsing mechanisms available to web site visitors.[1]

1. **Choosing the browser's "Back" button**: A user can navigate in reverse using the browser's "Back" button. In this way, the user receives a copy of the page he viewed previous to his current page. Pages received in response to a "Back" request are typically retrieved from the client-side cache on the user's browser.

   In the context of our running example, the *Papyrus.com* site shown in Figure 2, then, a user who has traversed the link from the *Fiction* node to the *Mystery* node can, by clicking on the "Back" button on his browser, effectively return to the *Fiction* node, even though there is no explicit link from *Mystery* to *Fiction* in Figure 2.

   Note that the "Back" button can be used not only for movement within a site, but also between sites (assuming the most recent click led the user from one web site to another). Using the browser's "Go" list of recently visited pages has a similar effect.

2. **Entering an explicit URL**: A user can, at any point, enter a specific URL, either internal to the current site, or to a different domain. For example, assuming that *Papyrus.com*'s (see Figure 2) URL structure reflects the site's catalog hierarchy, a user entering the string "http://papyrus.com/fiction/mystery" will navigate to the *Mystery* node, even though there may or may not exist a link from the user's current location to that node.

   We note here that choosing an item from the browser's bookmark or history lists has a similar effect. [2]

3. **Entering a search query**: A user interested in a particular book M_y by Agatha Christie might, from the home page of the *Papyrus.com* site (see Figure 2), enter the keyword string "Agatha Christie" in a search form, and receive in return a set of links to books by Christie, biographies about Christie, and perhaps mysteries by other authors. From there, the user can choose the link to his book of interest (assuming the search of the site has returned a link to that book). In this example, the user navigates from the *Papyrus.com*

[1] This list is not claimed to be exhaustive.

[2] Clearly, typing in a URL at a site which uses URLs to post data to the server is difficult, if not impossible, given the cryptic formatting of such URLs. However, this option is always available for addresses in the Bookmark and History lists, which store the URLs.

node to a *search-result* node (not represented in Figure 2), and from there to the node of interest, M_y.

In summary, such navigation poses two major problems for the site and interaction models described in Section 3.1 and 3.2, respectively.

1. It is possible for a user to navigate from u to v, $u \neq v$, $v \in V$, where $(u, v) \notin E$.

2. A user's search query may result in the user moving from his current location, u, to a search-result node, R, which incorporates information from various nodes (yet is not represented by any of them, as mandated by the model described in Section 3.2), and from R to another node $v \in V$.

Thus, we extend the notion of a *navigation* action, described in Section 3.2, to include these types of navigation. In order to handle this extended notion of navigation, we must also extend the site model (presented earlier in Section 3.1) to handle the search-result nodes generated by a keyword search. For this purpose, we introduce the *Extended Catalog Model*, which extends the SCM to include the notions of *virtual edges* and *virtual nodes*. We discuss each of these concepts in turn.

- A *virtual edge* is an edge $(u, v) \notin E$ which is created on the fly in response to the use of one of the alternative navigation mechanisms described above. Consider the case (described above) where the user clicks on the "Back" button and moves from *Mystery* to *Fiction*. Here, a virtual edge $e =$(*Mystery,Fiction*) is created. A virtual edge is visible only to the user for whom it was created, and is "destroyed" after that user has traversed it.

  Clearly, since virtual edges can link any two nodes in the site arbitrarily, we can think of the site, then, as a fully connected graph where some of the edges are defined explicitly at system startup, and others are created (virtually) in the fly as needed. Figure 3 depicts such a fully-connected graph on a subset of the *Papyrus.com* catalog shown in Figure 2.

- A *virtual node* is created in response to a user's keyword search request. This virtual node is connected to the site structure by means of virtual edges. Perhaps an example

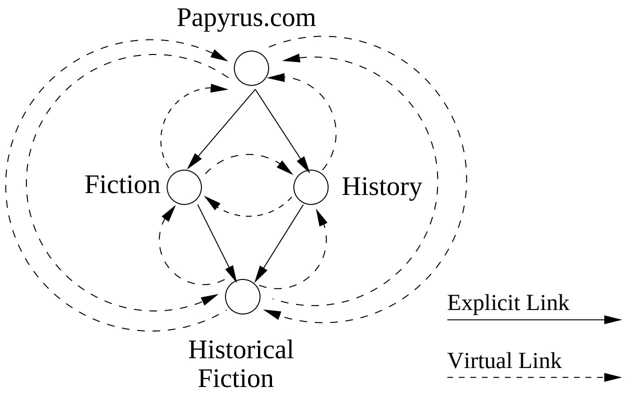

Figure 3: A fully connected graph with explicit and virtual links

will clarify the notion of a virtual node. Consider the case (above) of the user at the *Papyrus.com* home page. Here, the user is interested in books by Agatha Christie, and so enters the string "Christie" in the search form on the home page. In response to the search query, the user receives a page of search "hits" in return. He chooses one of the links

The virtual node *Search-results(Christie)* can be described in terms of the node structure discussed in Section 3.1. We can label this virtual node *Search-result(Christie)*, and generate virtual edges to link the virtual node from the user's current location and to the set of "hit" nodes resulting from the search. Thus, the virtual node's edge set consists of set of virtual edges. The virtual node's descriptor might contain an information element identifying the keywords used in the search, i.e., "Christie", as well as a set of link elements matching the virtual edges. A search form allows the user to begin a new keyword search.[3]

Figure 4 shows an example of the navigational implications of such a scenario. Here, the virtual edge (*Papyrus.com,Search-result(Christie)*) leads the user to the search-result node. From the search-result page, the user can navigate to two book nodes, M_x and M_y. These are linked from the search-result node by means of the virtual edges (*Search-result(Christie)*,M_x), and (*Search-result(Christie)*,M_y).

[3]Note that this permits the chaining of virtual nodes, connected by virtual edges

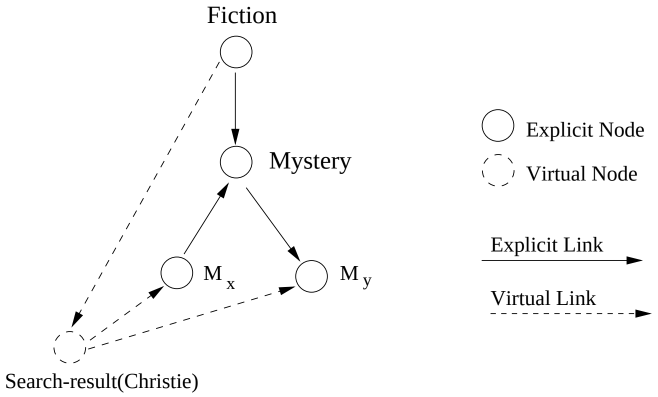

Figure 4: An example of a virtual node

3.2.3 Output: Site Response

As described in 3, in response to a particular action, the web site sends a *response* to the user. This response is in the form of a *web page*.

In sites using dynamic page generation, page structure and presentation is separated from page content. A page typically has a pre-defined *page layout*, i.e., the number, type, and relative position of *page components*(blocks of space reserved for particular types of content e.g., navigation bars, banner ads, etc.) is fixed, while the actual content in the page components (namely, the *elements* discussed in Section 3.1) is chosen on the fly based on the site's business logic and information available at run time.

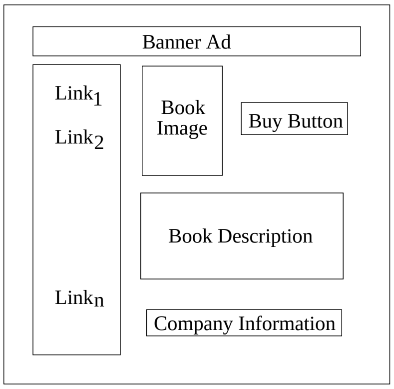

Figure 5: An example of a page

For instance, consider a page at our fictitious online bookseller, *Papyrus.com*. Assume this page represents a product node in our catalog (see Figure 2), perhaps the Mystery book M_i. Figure 5 shows an example of what a page layout for this book might look like. Here, the page shows an image of the book's cover, and provides a text description of the book. A "Buy" button allows the user to select the book for purchase. A set of links provide navigational opportunities to the user. The page also displays a banner ad and provides a link to information about the company.

In order to limit the complexity of site construction and maintenance, a site will typically have a limited number of page layouts, each designed for a special purpose. For example, an online bookseller's site may define a page layout for products, and another for categories of products. The web pages composed from a given page layout will have the same structure and "look and feel", but with different content.

We provide a more formal definition of a page:

Definition 4 Page. *A* page *is an abstract object containing a set of* page components*. A page component is a visual presentation of site content in two-dimensional space. We represent a page component with a 4-tuple of information:*

- **ID** *A component's* identifier *distinguishes it from other components in the page layout.*
- **TYPE** *A component's* content type *determines what type of element should be displayed in the component's space on a page.*
- **LOGIC** *A component's* business logic *tells the page generator how to choose the component's content. The content itself is drawn from the node descriptor of a user's current node.*
- **POSITION** *A component's* relative position *provides information on the placement of the component on the page. Note that the placement of components on a page is not necessarily disjoint; components may potentially overlap one another.*

Having described extensions to our site and interaction models, we are now ready to discuss a unified model of site interaction.

3.2.4 A unified model of user behavior on a site

Based on the site and interaction models in Sections3.1 and 3.2, we model user behavior on the site as a *session*, i.e., a sequence of $\langle action,\ node \rangle$ pairs, where each pair denotes a user click (i.e., action) on the site.

For example, a session modeling a user navigating from the root of the *Papyrus.com* catalog (shown in Figure 2) to book H_3, purchasing H_3, and then departing from the site might consist of the following sequence: $\langle N_{Papyrus.com}, N_{Fiction}, N_{HistoricalFiction}, N_{HF_3}, B_{HF_3}, D \rangle$, where N_i denotes navigation to node i, B_j denotes buying item j, and D denotes departure from the web site.[4]

4 Utility of the model

Having given a thorough description of our site interaction model, we now describe an area of research and development where our model can be of particular value – web site analysis. Web usage analysis essentially refers to methods of knowledge discovery of user navigation patterns on web sites through the use of data mining.

Several commercial tools area available for the purpose of web usage mining, e.g., [11, 15, 19], and much work has been published in the academic literature in this area, e.g., [3, 18, 17], most of which is focused on discovering user navigation patterns based on web logs. A tool for improving site organization through the use of web usage mining is presented in [12].

Most work in this area is based on web server log analysis, which suffers from several severe limitations:

- Rather than tracking high-level semantic actions, the web server log stores file hits, where each page served to a user may contain several files, and result in several log entries. In [6], the authors provide a thorough discussion of the problems of mining web logs, and propose data preparation methods to help mitigate some these problems. However, the underlying problem – that user navigation on the site is mapped to the invalid notion of static pages

[4]A more complete model of site interaction includes the notion of site output, thereby permitting analysis of user reactions to specific stimuli at the site. However, this is significantly more complex than the length limits of this forum will allow, and is thus left to future work.

– still exists.

- E-businesses often want to combine transactional and navigational data for analytical purposes. However, these two types of data are stored separately; matching purchase transactions to user sessions in the web log is all but impossible.

Our model, in which both transactions and navigation are represented as semantic actions on the site, has the potential to have a great impact on this area. For example, mining semantic user interactions with a site can potentially yield some interesting behavior patterns that have thus far been difficult to discover using current techniques. This, in turn, can lead to new customization techniques for web site interaction with users. In summary, our model has the following implications:

- It provides a representation technique for modeling/structuring a web site. As such, a corresponding language can be defined to describe a site.
- The site description can be used as a catalog that may drive the process of data acquisition related to user navigation and actions at a site.
- The model provides a better insight into user behavior at a site than currently available models. As such, it is a better choice for analyzing user access patterns.

5 Conclusion

In this paper, we present a unified model of a web site, including the site data and structures, as well as user interaction with the site. Based on these models, we develop a notion of user behavior on a web site. We also describe several relevant examples of the utility of our model.

Our plans for future work include extending this model to incorporate the notions of time spent on a page and the effect of server output (i.e., web page content) on user behavior. We anticipate using some modeling work from active databases related to events and event algebras to further enhance our model.

References

[1] G.O. Arocena and A.O. Mendelzon. Weboql: Restructuring documents, databases, and webs. In *Proceedings of the Fourteenth International Conference on Data Engineering*, pages 24–33, Orlando, Florida, February 1998.

[2] P. Atzeni, G. Mecca, and P. Merialdo. Semistructured data in the web: Going back and forth. *SIGMOD Record*, 26(4):16–23, 1997.

[3] A.G. Buchner, M. Baumgarten, S.S. Anand, M.D. Mulvenna, and J.G. Hughes. Navigation pattern discovery from internet data. In *Proceedings of WEBKDD'99: Workshop on Web Usage Analysis and User Profiling*, 1999. http://www.acm.org/sigs/sigkdd/proceedings/webkdd99/toconline.htm.

[4] R.G.G. Cattel. *The Object Database Standard ODMG-93.* Morgan Kaufmann Publishers, San Francisco, CA, 1994.

[5] The Washington Post Company. washingtonpost.com. Available at http://www.washingonpost.com.

[6] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[7] Microsoft Corporation. Active server pages. Available at http://msdn.microsoft.com/library/backgrnd/html/msdn_aspfaq.htm.

[8] D. Florescu. Personal communication, May 2000.

[9] D. Florescu, A. Levy, D. Suciu, and K. Yagoub. Optimization of run-time management of data intensive web sites. In *Proceedings of the 25th VLDB Conference*, pages 627–638, September 1999.

[10] L.V.S. Lakshmanan, F. Sadri, and I.N. Subramanian. A declarative language for querying and restructuring the web. In *Proceedings of the Sixth International Workshop on Research Issues in Data Engineering - Interoperability of Nontraditional Database Systems*, pages 12–21, February 1996.

[11] MINEit Software Limited. Easy miner. Available at http://www.mineit.com/products/easyminer/, 2000.

[12] L. Faulstich M. Spiliopoulou. Wum - a tool for www ulitization analysis. In *The World Wide Web and Databases, International Workshop WebDB'98*, pages 184–203, March 1998.

[13] A.O. Mendelzon, G.A. Mihaila, and T. Milo. Querying the world wide web. *International Journal on Digital Libraries*, 1(1):54–67, 1997.

[14] Sun Microsystems. Java server pages. Available at http://java.sun.com/products/jsp/.

[15] net.Genesis. net.analysis. Available at http://www.netgen.com/.

[16] P. Paolini P. Fraternali. A conceptual model and a tool environment for developing more scalable, dynamic, and customizable web applications. In *Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology*, pages 421–435, Valencia, Spain, March 1998.

[17] C. Shahabi. Knowledge discovery from users web-page navigation. In *Proceedings of RIDE'97 - Seventh International Workshop on Research Issues in Data Engineering*, 1997.

[18] M. Spiliopoulou, L.S. Faulstich, and K. Winkler. A data miner analyzing the navigaitional behavior of web users. In *International Conference of ACAI'99: Workshop on Machine Learning in User Modelling*, 1999.

[19] Web Trends. ebusiness intelligence solutions. Available at http://www.webtrends.com/products/Intelligence.htm, 2000.