For open-ended information tasks, users must determine how to best use their time in sifting through many potentially relevant documents. Document triage is the process of rapidly assessing the potential value of documents with respect to the task at hand. Normally, people perform triage using multiple applications in concert: a search engine interface presents lists of potentially relevant documents; a document reader displays their contents; and a third tool—a text editor or a more specialized application—is used to record notes and assessments. To support document triage, we have developed a multi-application environment that combines an information workspace with a modified document reader. This environment infers users’ interests based on their interactions with both applications, coupled with an analysis of the characteristics and content of the documents they are interacting with. It then uses this interest profile to generate visualizations that bring users’ attention to documents or parts of documents that match those inferred interests.
INTRODUCTION

The Web is a vast information resource that may be brought to bear on many different types of activities. Some activities dictate a specific information requirement, such as the location of a restaurant, a stock quote, or the answer to a question. We are concerned with more open-ended information tasks, analysis tasks in particular, where people use the Web to collect a variety of documents for interpretation and synthesis.

The problem in such tasks is that, even with the best search engine and the most effective query formulation, people need to rapidly work through long lists of documents to bring together all the information they need; there is no single document containing one right answer. In fact, as they skim early documents, they may well determine additional information needs that result in further queries and more documents to process [3]. We call this rapid assessment of documents based on their potential value information triage [19] or document triage [2].

Time is frequently a limiting factor in document triage: there are too many documents to assess and too much reading to do. Document triage involves selecting which documents to examine in more detail; locating the parts of documents that are related to the current activity; and keeping track of progress through the collection and recording the documents’ relative worth or taking notes on their content.

For more than a decade, we have been developing spatial hypertext workspaces supporting information analysis and document triage. These workspaces accelerate document assessment by providing users with fast methods for expressing perceived value and relationships among documents. Assessment involves skimming and selective reading, which usually occurs in a separate reader, such as a Web browser or a more specialized reading tool such as Adobe Acrobat or Microsoft Word. Thus, users often interact with multiple applications during triage: for example, a Web browser or information workspace that displays an overview of search results; a reader for viewing the content of individual documents; and possibly a third tool to keep track of what has been read and for recording brief notes or assessments. To better support document selection and skimming, we have begun inferring user interest based on user interaction with multiple triage-related applications [1]. Our past research shows that combining evidence from an information workspace and a document reader improves the overall quality of the user interest model.

This paper describes our efforts to use models of user interest as a basis for visualizations that draw a user’s attention to similar documents and to portions of documents that match these interests. In particular, we discuss how user interactions with documents—including implicit expressions of interest such as scrolling along with explicit expressions of interest such as annotations—can be combined with document and content characteristics and translated into visualizations which may be applied to other relevant documents to simplify the process of triage.

The paper describes an overall architecture necessary to support document triage and the three software tools we modified or developed: a spatial hypertext workspace, an annotation plug-in for Firefox, and an interest profile manager. The next sections survey related work and present our overall approach and architecture. We then describe the three tools in detail. Finally, we present the how the tools work in concert to generate the visualizations that may aid in document triage.
RELATED AND PRIOR WORK

Related work falls into three main categories: research investigating document triage itself; research into identifying and modeling user interest; and research on the use of visual cues to aid navigation.

Document Triage

Whereas sensemaking focuses on the intensive reading of a single document, document/information triage involves engaging with multiple documents at the same time with more cursory and high level readings. Document triage is a process of sorting through (possibly numerous) relevant materials, and organizing them to meet the needs of the task at hand. It is often a time-constrained process that requires quick assessment based on incomplete knowledge [19].

Document triage has been studied from different perspectives. Modeling user acceptance of documents can rely on document features such as title, length of document, presence of images etc. [8]. This modeling can also be based on user behavior during document triage and on the types of external information structures that people build for information that is important to a task (e.g. categories and spatial structures) [2][19][22].

Investigating the differences between the document triage process on paper and electronic media provides a deeper knowledge of what users do during information triage and how they interact with the documents that they evaluate [4]. For example the two media have different affordances: participants reported that annotation was easier on paper, for instance. Moreover, the affordances of tools used during triage (such as spatial hypertext [19] and display configuration [2]) can have an impact on the practice of triage.

Interest Modeling

As with many search-like activities, assisting triage requires that we first identify what documents users are interested in. This can be done using explicit indicators, implicit indicators, or a combination of both.

Explicit indicators are based on users directly identifying which documents are interesting, e.g. ratings. Several digital library systems use this approach [20][23]. Since explicit indicators are directly assigned by users, they are easily understood and do not require further interpretation. However, eliciting this information can interfere with user patterns of reading and browsing [7]. Users may also stop providing explicit ratings when they do not perceive an immediate benefit for their efforts [14]. Moreover, users rate far fewer documents than they read [24]. So even though explicit ratings are straightforward and easy to interpret, their use may not provide the information needed to provide assistance during an information task.

Using implicit indicators does not require direct feedback from users. Rather, interest information is collected by recording user behaviors and extracting features of the documents that they interact with. However, this information must be heuristically interpreted to provide the desired interest information. Implicit indicators are collected by software as users interact with the software. These interactions may include user behaviors such as mouse clicks, scrolling, and time spent looking at a particular document as well as user expression such as edits and annotations. The meanings of interactions and expressions may vary among different software applications.
Systems that monitor and respond to implicit indicators model interest in the context of a knowledge structure. This knowledge structure can be pre-existing [5][11] or created based on user behaviors and document features. Some models incorporate characteristics of the larger user population [6][17]. Models that focus more on an individual user can be more task specific [21] or adapt as user activity is observed over multiple sessions [13][15]. User behaviors can be difficult to decipher and use since a behavior can be interpreted in different ways. User expression has the potential to provide more focused information with fewer opportunities for misinterpretation. Annotations such as marks, highlighting and notes can provide more fine grained information for creating a user model [12][29].

These systems generally treat documents as an atomic unit. However, document content tends to exhibit more granular divisions that are used by people when viewing them. These systems also monitor user activity in the context of a single application. Also, real world activity is not limited to the confines of a single application. Recognizing this, we found that models using a combination of the implicit information from multiple applications work better than relying on the information obtained from a single application in document triage [1].

**Visualizations Aiding Navigation**

For users to take advantage of system identified “interesting documents,” they must be made aware of those documents. In the case of standard queries, documents are listed in a ranked order. User models can be used to modify rankings [21]. When interacting with a hierarchical structure where nodes represent documents, the nodes can be zoomed in and out and hidden based on inferred user interest [11]. The ordering of documents does not have to be manipulated to indicate interest. For example, implicit queries in Data Mountain identify documents similar to the one being viewed [10]. These query results are indicated by outlining the documents in green within the Data Mountain. XLibris draws a user’s attention to passages of potential interest based on their query rank by using color and icons to highlight a document overview [29].

**APPROACH & ARCHITECTURE**

This research examines how inferred models of user interest may be used to generate visualizations that aid the user in performing document triage. In particular, we explore how relevant reading and organizing applications can contribute to and share these models to mutually support the activity with different forms of visual feedback.

Our approach to generating visualizations includes four steps:

- inferring a user’s interests from his or her interactions with a collection of relevant documents;
- identifying and representing classes of user interest based on relations among documents of interest, e.g. being assigned the same color;
- identifying documents that are similar to one or more classes of interest; and
- generating visualizations that reflect how these documents are related to the inferred interests.

There are usually three types of applications that people use when they perform triage on a collection of Web-based documents: first, they may use some type of overview application that allows them to specify implicit or explicit queries and examine the resulting sets of Web documents; second, they may use an application to read or scan the contents of the documents;
finally, they may use some type of application that allows them to organize the documents and possibly express some further interpretation of what they have seen. For example, when people are triaging Web documents, these three types of applications might correspond to a browser that displays search engine results, a PDF/html viewer that displays individual documents, and a text editor that allows the user to manipulate valuable URLs.

Thus our architecture must handle the requisite types of applications to perform the basic triage activities (presenting search results, examining individual documents, and organizing the search results according to an assessment of their value). It must also include an Interest Profile Manager that accumulates and analyzes implicit and explicit indicators of user interest from the different types of applications and generates the appropriate visualizations for each. Finally the architecture must have a provision for storing interest profiles so they persist across sessions. The applications must each engage in two-way communication with the Interest Profile Manager in a way that is extensible so that additional applications can be added as appropriate; for example, we expect different reading applications to be added to the architecture to extend the capabilities offered by our proof-of-concept Web page reader.

Figure 1 shows this overall architecture. In the implementation we discuss in this paper, we simplify the general architecture by collapsing the application used to present the search results and the application used to organize them into a single application, the Visual Knowledge Builder (VKB). VKB has a built-in connection to Web search services and is designed to support organizing documents. The reading application is represented by the WebAnnotate, which is intended as a proof-of-concept, rather than as a fully-functional reader. We have implemented the shared Interest Profile Manager (IPM) and a means of storing the interest profiles across sessions; user activity in VKB and WebAnnotate is thus stored in the IPM and drives the visualizations generated for either application.

![Figure 1. Communication between document triage applications.](image)

The next sections briefly describe the two applications that we have used as a testbed, VKB and WebAnnotate, focusing on the features that are important for triage. These descriptions are followed by a discussion of the IPM and how it collects and analyzes records of user activity to generate cross-application visualizations.

**VISUAL KNOWLEDGE BUILDER**

The Visual Knowledge Builder (VKB) is a spatial hypertext workspace for collecting, analyzing, and organizing documents [27]. Much like its predecessor, VIKI [18], documents in VKB are
represented by visual objects displaying metadata; these document surrogates are arranged in a hierarchy of two-dimensional spaces called collections. This paper introduces VKB 3, a major revision that includes much greater capabilities for working with Web-based information, such as the combined thumbnail and metadata views shown in Figure 2.

Figure 2. Web documents, represented as objects combining thumbnails and metadata, are organized in VKB.

Figure 2 shows the workspace that results from a Yahoo search on “antimatter” that returned 20 Web documents. The hypothetical user has moved articles he perceives to be valuable as definitions of antimatter and arguments for its existence into a pile and colored them blue. He has created a similar green pile to categorize articles about antimatter and its potential for space propulsion Web pages that refer to the antimatter energy drink and other documents unrelated to the current task remain in the original collection generated by the user’s search (titled “Yahoo Search < antimatter >”); he has changed them to dark gray so he knows to ignore them (although he could have also deleted them). He hasn’t looked at the remaining articles yet. It is likely the workspace will grow to contain more collections as our user works with the documents and discovers the need for further exploratory searches; he may also create new collections to categorize additional documents and manage the space.

Prior versions of VKB are described in detail in [26][27][28]. The application is written in JAVA and runs on MS Windows, Apple’s OS X, and Linux and VKB 2 is available for download at http://www.csdl.tamu.edu/VKB. VKB 3, shown for the first time here, is in use internally and later will be released publicly.

For the current investigation of triage, VKB 3 has been augmented with new capabilities to communicate with the Interest Profile Manager (IPM). As users open, move, color, delete, or otherwise modify document objects in the workspace, these actions are sent to the IPM to be
used in inferring user interests. VKB also receives information on user interests from the IPM and uses this information to provide additional visual cues.

VKB 3 also introduces object layers. Each VKB object can include multiple display layers. Web document objects have a main layer, which can be configured to show various combinations of metadata (title, URL, etc.) along with a thumbnail of the Web document, and a system layer. This layering enables a user to informally express preliminary interpretations of document characteristics and value (via color and other visual attributes) in the main layer, while the system uses visual attributes of the system layer to provide cues, hints, or suggestions without interfering with user expression.

VKB sends the user’s visual and textual edit information to the IPM along with identifying metadata, in this case a URL. These visual events include move, resize, change background color, change border color, and change border width. Textual edits include new attributes and values set by the user and freeform textual annotations.

WEBANNOTATE

When a user opens a document in the VKB workspace, it is displayed in the Firefox browser (or Microsoft’s Internet Explorer if so desired by the user). To further facilitate triage, we have developed a Mozilla Firefox add-on called WebAnnotate that provides basic annotation capabilities, collects data on users’ interactions with documents, and uses interest data returned from the IPM to create visualizations that enhance document skimming and reading.

WebAnnotate supports simple forms of annotation on HTML documents. While sufficiently functional for our purposes, WebAnnotate’s annotation model is not unique. In particular, Microsoft Word, Scrapbook, Annozilla, and Annotea were valuable references in designing the annotation features of WebAnnotate.

Creating Annotations

Once users activate the annotation toolbar (Figure 3), they can highlight and underline text and can create colored sticky notes to place on the current page. WebAnnotate’s sticky notes are editable translucent text boxes that can be moved to anywhere on the HTML document. Highlights and underlines designate a span of anchor text that is of interest or importance. All forms of annotation may be of any color.

Figure 3. Annotation toolbar of WebAnnotate

Figure 4 shows annotations created on a HTML document in WebAnnotate. WebAnnotate saves the annotations separately from the HTML document. Users can choose to save the annotations locally or via a Web service. Whenever a user opens an HTML document, WebAnnotate determines if there are any annotations saved for that document by checking locally or with the service. If any annotations are found, WebAnnotate regenerates them.
As annotation information is collected, it is conveyed to IPM. This information includes the color and type of the annotation as well as terms associated with the annotation (i.e. the anchor text).

**User Events and Document Attributes**

Our prior study of document triage practice [1] showed that data that describes users’ activities while they read, for example the time they spend on a page and how much they scroll, in conjunction with document attributes, such as document length or the number of embedded hypertext links, can be a meaningful source of evidence for inferring user interest. Thus the second major role of WebAnnotate is to collect this type of data on users’ interactions with Web pages (such as scrolling, mouse clicks, and changes in focus) and the corresponding data about the characteristics of the web pages themselves (page length, number embedded links, and number of images). This information is aggregated and communicated to the Interest Profile Manager.

**INTEREST PROFILE MANAGER**

The Interest Profile Manager (IPM) plays the central role in inferring user interest during document triage. The IPM collects information about interest-related activity from the multiple applications involved in document triage. This information is aggregated and saved in the user’s interest profile. Based on the interest profile, the IPM estimates user interest and broadcasts this interest to the participating applications.

Thus, the IPM acts as an interest profile server, while document triage applications act as interest profile clients. Any application that can be modified to include the interest profile client software interface can communicate with the IPM. Currently, VKB 3 and WebAnnotate include this interface as a proof of concept.
Representing User Interest

Each application provides a user with unique ways of interacting with information that can be used as a basis for inferring his or her interests. It is this type of interaction data that is sent to the IPM. For efficiency, applications send aggregated low-level activity data (such as scrolling data) to the IPM after major events. For example, WebAnnotate, sends data to the IPM when the document is replaced in the Web browser; this aggregated data includes the duration the document was displayed on the screen, scrolling information, any annotations created or modified, and other reading- or document-related characteristics.

Although each application has unique information that may be used to gauge human interest, this interest assessment needs to be sharable among the different applications to be useful to the triage process as a whole. The IPM depends on an abstract XML representation for receiving interest-related information from applications and for broadcasting inferred interest to client applications. Because we realize that we cannot foresee all of the ways different applications will allow users to interact with documents, the representation is extremely general and extensible. Thus an interest profile takes the form of a document identifier, application identifier, and a list of attribute/value pairs that is provided by each application. In this way, new applications only have to inform the IPM of their attributes when registering. Based on the registration, the IPM sets up the appropriate data structures to provide interest information based on application’s unique set of attributes and actions. In the current implementation, an Interest Profile is a list of documents and document segments, user activity associated with each document or segment, a term vector that characterizes each document or segment, and a set of visual and metadata features for each document or segment.

Inferring Interest

Client applications (currently VKB 3 and WebAnnotate) send the IPM records of user activity and document attributes. The data from each application are different, but they can be used similarly to infer the user’s varying interests. The different types of interaction are used as expressions of user interest or lack of interest (e.g. object deletion). The document attributes, such as assigned color, are used to determine relationships among documents.

There are two main processes the IPM uses to infer user interests: one to determine the descriptor for a class of document or annotation and a second to aggregate interest in that class. To aid in the creation of descriptions of document classes, the IPM includes term vector and metadata analysis capabilities as well as text tiling [16] capabilities to allow clients and the IPM to do text analysis at the sub-document level. To infer the level of interest in document classes, the IPM includes interest estimators and aggregators based on prior use of the tools (e.g. [1]).

For example, consider the three dark blue VKB objects in Figure 2. The IPM will first retrieve the documents at each of the three URLs, aggregate the term vectors for those documents, and identify the metadata that is common across the three objects. This results in a characterization of the document class. Next, the IPM will calculate an interest rating for each document. It does this by combining the user activity associated with each document across both of the triage applications; in other words, the IPM combines activity data for the document surrogate as it was manipulated in VKB with the corresponding activity data for the document as it was viewed and annotated in WebAnnotate. Finally, the three individual interest values are aggregated—currently they are summed—to quantify the user’s interest in the document class.

The IPM’s user interest estimator module includes multiple algorithms for estimating user interest based on the collected interest-activity information. This enables us to try alternate
reasoning techniques; as users become more familiar with the IPM, they may select the combination of algorithms that works best for them.

**Recognizing Similar Documents**

Once the IPM has generated a list of document classes and the user’s relative interest in these classes of documents, this information is broadcast to the registered applications and can be used to generate visualizations that distinguish the documents or parts of documents that fit into these categories. By assigning different colors (or other visual features) to document objects in VKB and by using multiple annotation colors in WebAnnotate, users are expressing implicit classifications for their own later use. Currently, we have only explored the use of such visual features for distinguishing document classes as these classes have the advantage that they have relatively intuitive mappings to the adaptive visualizations that are described in the next section.

**VISUALIZATIONS**

As we discussed previously, the IPM provides VKB and the WebAnnotate-augmented browser with a list of document classes. For each document class there is an estimation of user interest, an aggregate representation of the documents in the class, and the feature(s) that distinguish the class.

This information is used to generate visualizations that can either be applied directly to new objects or document segments within the same application, or visualizations can be translated in a more or less intuitive way to be applied to analogous situations in the other application. Thus, there are VKB-to-VKB visualizations, VKB-to-WebAnnotate visualizations, WebAnnotate-to-VKB visualizations, and WebAnnotate-to-WebAnnotate visualizations. These four forms of adaptive visualization are described below through a triage scenario. In this scenario, the user is in the process of collecting resources for a paper on antimatter and its potential use in space propulsion systems and is working through the documents in Figure 2.

**VKB to VKB**

In our earlier scenario, the user has classified the retrieval results in the workspace as either green, blue, and dark grey, according to his perception of what is in the documents and whether they are useful for his paper on antimatter. Given this user expression, VKB 3 uses the results from the IPM’s analysis to indicate the likely color/classification of the remaining documents.

In this case, the term vectors for each uninterpreted document are compared to the three classes. If a document is close to one of the three clusters and not to the other two, the system layer is colored to be the same as the document objects in that class. The saturation of the color indicates the system’s confidence, determined by the degree of match between the document term vector and the class term vector. Figure 5 shows examples of the resulting visualization; objects with system layers that are shades of green, blue, or gray indicates the strength of similarity. Objects for which classification is ambiguous are unchanged.
Figure 5. Colors in objects’ system layer indicate similarity to documents given that color previously by the user in VKB or similarity to document contents annotated in WebAnnotate. Some objects are unaffected and others take on various shades of green, blue, red, and grey depending on the strength of the similarity.

**VKB to WebAnnotate**

In VKB, color is applied to objects in the system layer to refer to the classification of whole documents; WebAnnotate uses the same visual classifications to direct the user’s attention to portions of documents that are textually similar to the class.

When notified of the IPM’s document classifications, WebAnnotate uses the IPM to compare the term vectors for document segments of its current document with the classifications that were identified based on user expression in VKB. The IPM informs WebAnnotate of the matches, and the resulting text segments are given a lightly colored thick underline to indicate the similarity. Figure 6 shows two segments with a green underline and two segments with a blue underline.
Figure 6. The lightly-colored thick underlines indicate similarity of document segments to classes of documents expressed in VKB.

WebAnnotate to VKB

As users annotate documents they examine in WebAnnotate, the contents, type of annotation, and color are communicated to the IPM. This data is aggregated with the annotations to other documents to create classes based on the color of the annotations, or combinations of annotation color and type (e.g. red underlines represent a class). Thus, for each class, a term vector of keywords from the annotated text is broadcast to the triage applications.

As with VKB to VKB visualization, these term vectors are compared against the contents of the Web documents represented by objects in the VKB workspace, resulting in a similarity rating of not related, low, medium, or high to each class. The border color of the system layer of the document object is then assigned the color of the best matching class with the transparency level determined by the interest level.

Consider the case where our user continues his task by annotating documents on antimatter propulsion. Passages relating to the design of antimatter engines are annotated in green while passages about antimatter production and its cost are annotated in red. This results in the addition of a red and a second green interest class to the blue, green, and gray document classes that were already in place from the prior use of VKB. Figure 5 shows the visualizations in VKB that result from these WebAnnotate annotations. Two objects have been assigned red or light red system layers because they are more similar to the class containing documents about antimatter production and its cost.
WebAnnotate to WebAnnotate

If annotations reflect a user’s interest in a particular concept, it may be valuable to see other discussions of the same concepts later in the document or in other documents. This was the idea behind XLibris’ ability to propagate annotation ink to throughout a document [25]. As a generalization of this technique, the annotations created in WebAnnotate can be used to automatically identify portions of later documents that are similar to passages that have been read and annotated.

As in the case of the visualizations that are propagated from WebAnnotate to VKB, the same interest classes are defined based on the color, type and content of annotations. To identify segments of new or unread documents to bring to the user’s attention, these classes are then compared against the segments of the document currently displayed in WebAnnotate generated by the text-tiling algorithm. When a match is identified, a thick underline of the appropriate color for the class is used to signal the similarity to the user.

CONCLUSIONS AND FUTURE WORK

We have developed an architecture that supports triage across multiple applications. The central element of this architecture is the Interest Profile Manager, which receives information about user activity from the individual applications and broadcasts inferred user interests back to the applications. The IPM consolidates functionality necessary to characterize user interests, including the ability to collect, parse, and determine similarity among common forms of Web documents. In our example instantiation of this architecture, the IPM communicates with a Visual Knowledge Builder workspace and an annotation-enabled Web browser. The examples in the prior section show how the actions of users in either of these applications can generate assistive visualizations in both applications.

To extend this infrastructure with additional applications, the applications must be able to record and aggregate user activity and communicate it to the IPM and/or receive and use broadcasts from the IPM. Applications need not do both; it may make sense for an application that incorporates a non-interactive visualization technique to receive information about inferred user interests without sending any information to the IPM about user activity. Similarly, an application may be interactive, and may offer considerable insight into a user’s interests, but it may not make sense to modify anything in that application accordingly; for example, if the user is writing a paper while she is performing triage, the topics that emerge in the paper may be a very effective source of interest profile data.

The classification of documents of into different user interests in the current IPM is based solely on explicit user expression in a single application. For example, documents or subdocuments that a user colors red will generate red visualizations for documents or subdocuments considered similar. Other applications have identified classifications of documents by clustering the documents based on textual analysis or image processing [9]. Such capabilities may help determine when the user has multiple interests that are expressed using the same color or when the user has used different colors to express the same interest.

To make the IPM more readily extensible, the IPM needs to incorporate an abstract model that characterizes the expressive and presentational capabilities of applications. For example, such a model would specify that VKB allows users to assign colors to document surrogates to informally express their interest in a document, their understanding of what a document is about, or a general assessment of its worth to them. By contrast, WebAnnotate displays documents’ contents; thus any user expression of interest or other interpretation conveyed through annotations happens at a sub-document level. Components of such an application model include
the granularity of the information presented, persistent forms of user expression, transient forms of user interaction, and visualization methods supported.

Because our work to date has focused on the architecture, the efficacy of the visualizations we currently use to support document triage must still be evaluated through experimental or observational means. Such an evaluation is planned and will inform not only the redesign of the visualizations, but also the refinement of the IPM’s inference engine. It is not anticipated that these results will influence the overall IPM architecture or basic capabilities.

REFERENCES


