D-Lib Web Magazine, October 1998. See http://www.dlib.org/dlib/october98/dasher/10dasher.html, ISSN 1082-9873.

Collaborative Information Space Analysis Tools

Robert Neches, Sameer Abhinkar, Fangqi Hu, Ragy Eleish, In-Young Ko, Ke-Thia Yao, Quan Zhu, Peter Will
{rneches, fhu, iko, kyao, zhu,will}@isi.edu

University of Southern California
Information Sciences Institute
Marina del Rey, CA 90292

Abstract

The DASHER Project at USC/ISI has focused upon helping organizations with rapid-response mission requirements. Such organizations need to be able to quickly stand up tiger teams backed by the information, materiel, and support services they need to do their job. To do so, they need to find and assess sources of those services who are potential participants in the tiger team. To support this very initial phase of team development, the project has developed information analysis tools that help make sense of sets of data sources in an intranet or internet: characterizing them, partitioning them, sorting and filtering them. These tools focus on three key issues in forming a collaborative team: helping individuals responsible for forming the team to understand what is available, helping them structure and categorize on the information available to them in a manner specifically suited to the task at hand, and helping them understand the mappings between their organization of the information and those used by others who might participate. DASHER's Information Space Analysis Tools are unique in combining multiple methods to assist in this task. This makes the suite particularly well-suited to integrating additional technologies in order to create specialized systems.

Keywords: collaborative information analysis, integrated usage of services, electronic commerce

1. Introduction

The focus of DASHER (Defense Acquisition Services for High Performance Electronic Commerce) is the rapid formation and utilization of *task-oriented* information repositories in order to help organizations with rapid-response mission requirements. Such organizations need to be able to quickly organize rapid response teams (sometimes called "tiger teams"), backed by the information, materiel, and support services they need to do their job. To do so, the team and its organizers need to find and assess sources of those services and work with them to define the specific nature of the services provided. Prototypical examples include an engineering team formed to take an IFF (Identification Friend-or-Foe) device from design to deployment in weeks, a team coordinating technical, contractual and transportation requirements for an emergency purchase, or a disaster relief mission trying to assess locally available services.

This paper looks at information search and retrieval from the perspective of team formation. Our tools help to make explicit the implicit collaboration between service seekers and service providers. They do so by helping users to understand the relationship between the terms that they use to describe their *requirements as seekers of information and services*, versus the terms and categories that are used by *potential providers of information and services*. This helps initiators of a team to develop an explicit understanding of the relationship between their personal framing of the task and the categorizations used by others with whom they might work.

In contrast to many other efforts in information search and retrieval, which emphasize retrieval and are focused on finding a specific document, the DASHER Information Space Analysis Tools focus on making sense of *sets* of data sources: characterizing them, partitioning them, sorting and filtering them. The goal behind this effort is to provide tools that help in developing and assessing alternatives (be they alternatives for goods/services sought or alternatives for candidate providers considered). The information analysis tool set being developed is unique in combining multiple methods to assist in understanding, particularly natural-language text extraction techniques and ontology-based categorization. It provides user interfaces for set-manipulation operations and for visual display of partitionings, giving users means of identifying topics of interest to them and sorting the space of data sources accordingly.

One of the most significant features of the tools is that these capabilities give a user multiple means of *characterizing* the contents and topics of the document set. These include the ability to wrap net services, such as Yahoo [Yahoo], to extract

explicit formal taxonomies that have been developed as a common way of structuring the world. The methods also include the ability to extract *implicit* organizing features of the world, such as terms that appear to represent the hot topics in a community of interest (indicated, for example, by frequency of usage). Finally, the tools help users define their own categories, and sort a set of documents according to those categories, by mapping users' structuring of the information onto the explicit and implicit characterizations that have been extracted from the document set.

These tools are intended to address three critical tasks that arise in the very earliest stage of forming a collaborative team. First, they help individuals responsible for forming the team to understand what resources are available to them. Second, the tools help team members structure and categorize on the information available to them in a manner specifically suited to the task at hand. Third, and most crucial, the tools help team members understand the mappings between their personal organization of the information and those used by others in cyberspace who represent the alternative sources available for information and services.

The sections following elaborate upon these themes. Section 2 describes our prototype implementation of DASHER's Information Space Analysis Tools. We use as examples the issues that might arise in trying to form a design and development tiger team concerned with creating a product containing embedded GPS (Global Positioning Systems) technology.

We then discuss the underlining techniques in more detail. Section 3 describes approaches for categorizing a document set. We discuss using simple natural language techniques in combination with statistical analysis to get implicit categories for a document set. We also discuss how can one map a document set to predefined categories from web directory services such as Yahoo and LookSmart [LookSmart]. Section 4 talks about how categories can be expanded by using the existing members as a training sample and applying an induction learning algorithm to add more documents to the categories. We also discuss a technique that can search for web sites similar to an example web site and therefore can be used to expand our existing categories. Section 5 discusses plans for collaboration tools using Multivalent documents [Phelps and Wilensky, 1996a and 1996b] and Habanero [Chabert et al, 1998]. In section 6, we discuss GeoWorlds, a more complex application. Section 7 discusses the results and describes future plans.

2. Support Environment

The current suite of DASHER Information Space Analysis Tools consists of two major parts: DASHER client and DASHER server.

The *DASHER client* contains user interface components and the MVD annotation tool to provide collaborative working environment for organizing user-defined categories, displaying various views of the resulting categories and documents, partitioning the working set by filtering, and organizing the categories by grouping and ordering. All the users in a DASHER session can collaborate remotely while organizing their information space by sharing the user interfaces through the Habanero server. (See Section 5 for details about MVD and Habanero).

The *DASHER server* provides persistence services and various information retrieval and analysis services to the DASHER clients. Authentication server, persistence server, and clipbook server are the components in the DASHER server which provide the persistence services, and provide functionalities of authenticating users and managing the DASHER sessions, storing/retrieving information spaces to/from the persistence storage, and clipbook facilities to exchange data between different DASHER sessions. Various information retrieval and analysis services can be plugged-in to the DASHER server. The current DASHER server contains four major plug-in services for analyzing categories to extend a working set, extracting noun phrases from a set of documents, generating clusters of documents for a working set of documents, and extracting categories and documents from Web search engines. The DASHER server has a flexible architecture which can extend its services of information space analysis by including more plug-in services. Section 3 will explain detail about current plug-in services in the DASHER server and section 6 will mention integrating more services with the DASHER tools.

All the components in the DASHER client are implemented using Java and JFC (Java Foundation Classes), and are portable to any platform. The DASHER client interacts with the DASHER server using Java RMI (Remote Method Invocation) interfaces and sockets. Figure 1 shows the current organization of components in the prototype of DASHER Information Space Analysis Tools.

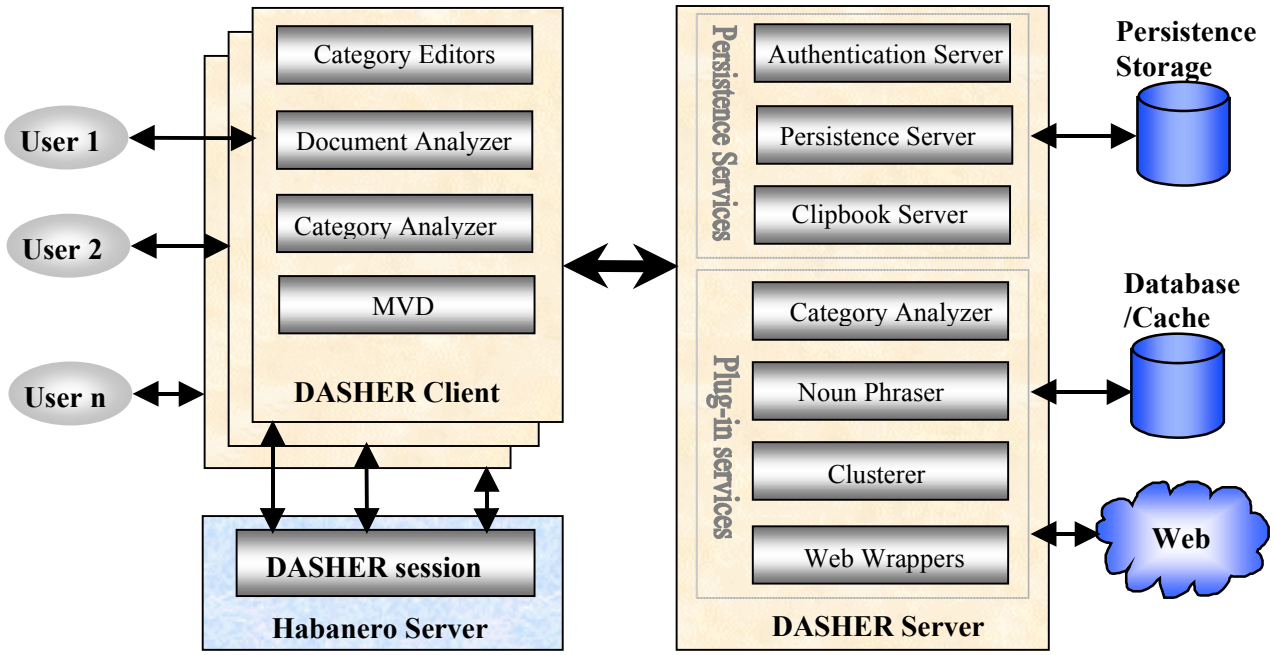

Figure 1. Current Organization of DASHER's Information Space Analysis Tools

The sections following illustrate the tools through an example: a case in which a tiger team is needed to rapidly build a product with embedded GPS (Global Positioning System) capabilities, requiring the team initiator to identify sources of GPS technology and services.

2.1. Developing customized categories

Users start the information analysis task by finding candidates using a Web browser and COTS (commercial off the shelf) Web search engines. While performing the search, users need to record and classify useful Web sites found by the search engines and form their own customized categories.

We call the tool that supports this function a *category editor*. Many of its functionalities resemble the bookmark facilities found in a Web browser, in that it is able to communicate with the browser in real time, bookmark the current URL into categories, copy and paste, etc. However, it is also able to perform additional operations on the user-defined categories like importing all the external links in a Web page at once under a category, merging different categories together, classifying a set of documents or a sub-category under multiple categories by creating symbolic links, etc. The user can open and maintain multiple category editors in a single DASHER session to organize multiple user-defined category structures for several different topics. Also the user can export a part of a category structure to another category structure using the internal clipboard system.

Figure 2 shows the category editor, in which the user opened two category editors, 'Main' and 'GPS' and created a category structure for GPS in the 'GPS' category editor. The user imported all the external links in a GPS reference page under the 'GPS Links' category.

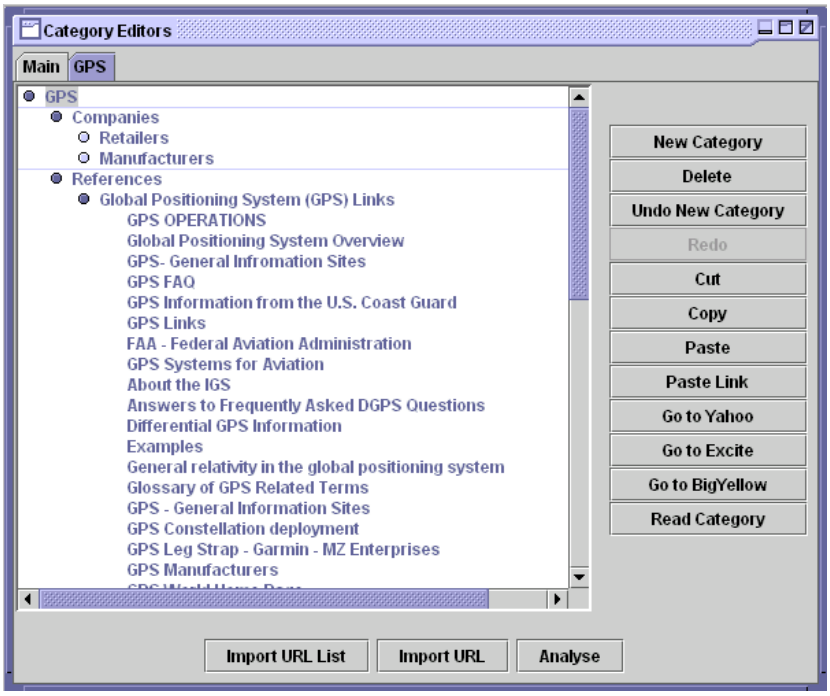

Figure 2. Category Editor

2.2. Internal analysis of the working set

While organizing the user-defined categories, users may need to analyze a set of documents in a category structure to extract useful noun-phrases, e.g., most frequently cited terms, place names and company names. They may also want help generating different classifications of the documents by clustering similar documents into several groups.

The *document analyzer* component in the DASHER client provides user interfaces for analyzing a selected set of documents and displaying the results in a structured and graphical manner. This component can be instantiated from other components in the client by selecting a part of categories or documents. Users can selectively initiate information analysis services in the DASHER server, and each service's result will be displayed in a tabbed panel. The document analyzer component also helps users to understand the clusters returned from the clusterer service by displaying the cluster maps and document lists. It helps users compare their own categories with the new classification by drawing 3-D bar graphs which shows the distribution of the documents in intersections of the two different classification schemes.

Figure 3 shows some of the windows available in the document analyzer interface. These display the most frequently cited noun-phrases, a new classification generated by the clusterer service, a cluster map, and a 3-D bar chart to compare the user's current classification of the documents with the alternative categories found by the clusterer.

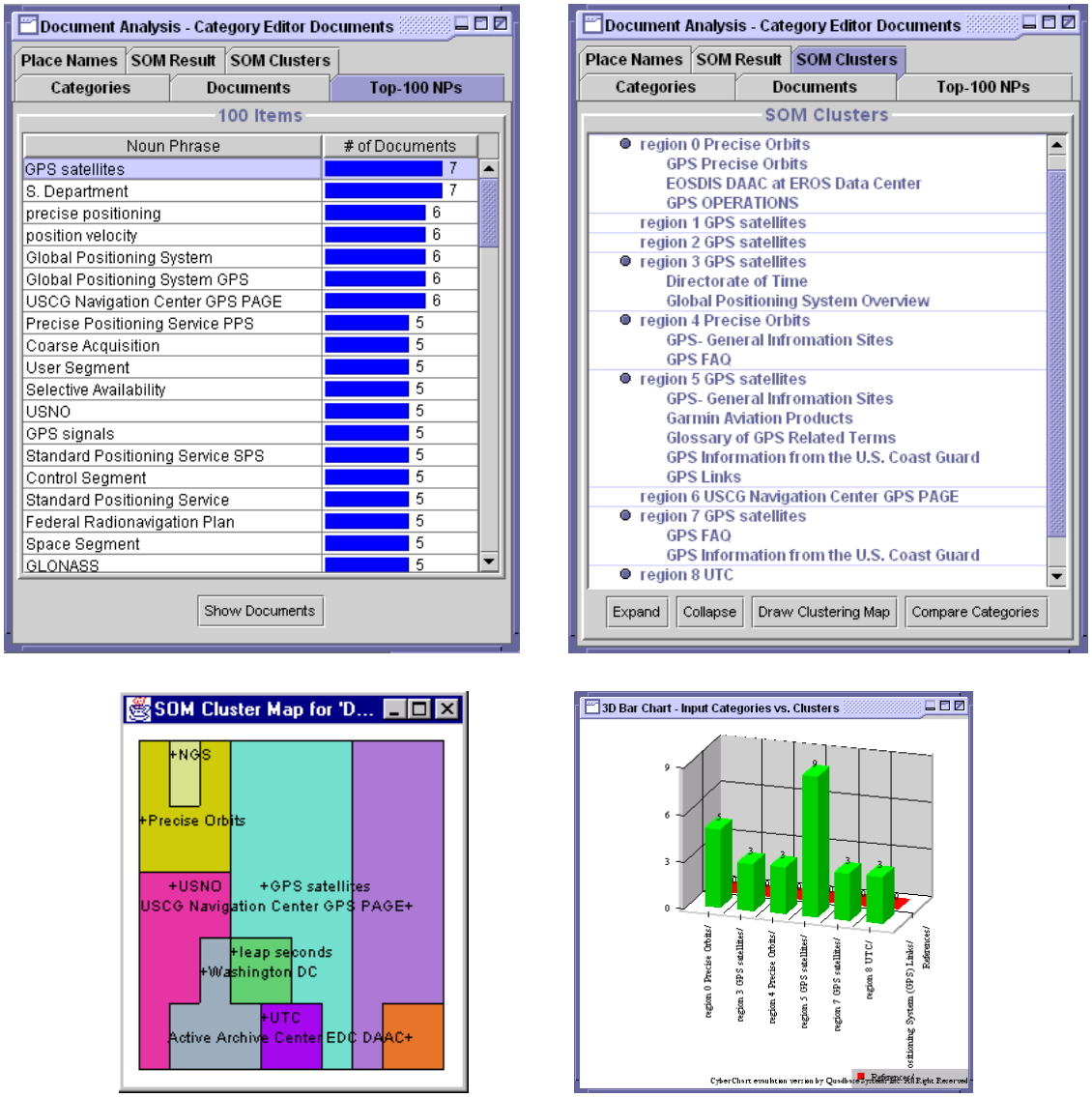

Figure 3. Document Analyzer Interfaces

2.3. Leveraging external analyses of the Web

Users can leverage others' work in analyzing Web sites by looking at large sets of pre-categorized Web documents, available from COTS search engines such as Yahoo and LookSmart. By doing this, users can identify pre-defined categories which are relevant to their documents collected by the category editor. Our *Web wrapper* in the DASHER server is a software component that wraps COTS search engines and extracts the pre-defined categories and all documents found under those categories. The *category analyzer* component in the client displays hierarchical and tabular views of the categories and documents extracted by the Web wrapper. By using this component, users can partition the working set by pruning a sub-tree in the category tree or by filtering category nodes using a node name.

Figure 4 shows the category analyzer displaying hierarchical and tabular views of categories and documents extracted from the Yahoo search engine. The node name 'Companies' has been used as a filter, so it shows only information related to GPS companies.

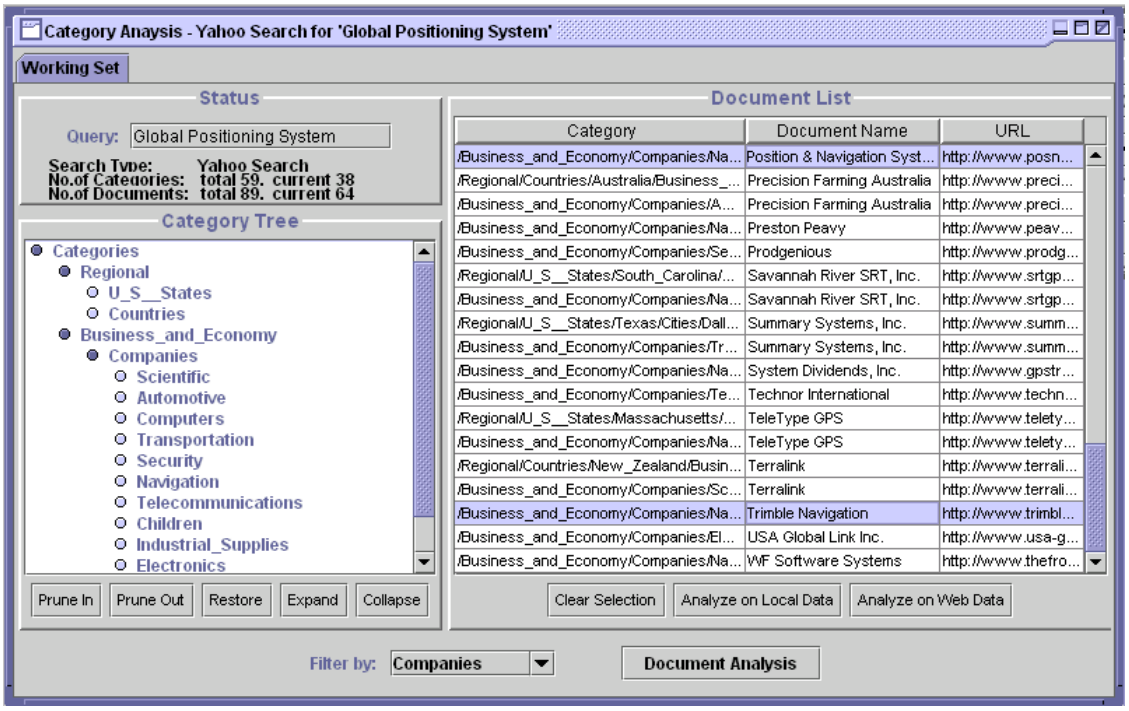

Figure 4. Category Analyzer

2.4. Extending categories / re-organizing sets of documents

When users want to develop more categories and documents related to current working set, they can request the category analyzer service in the DASHER server to do an extended search using fan-out (see section 4.2). The category analyzer is then able to display the resulting extended set of documents graphically. Using the category analyzer, users can re-organize sets of extended documents to meet their own needs by defining subsets and combining them via set operations. This allows users to define categories that make sense for their specific tasks, and to fit whole sets of documents into the new categories.

Figure 5 shows the extended categories for the two documents selected in Figure 4. Fan-out found that the documents belong to two categories: GPS and GPS/retail. In turn, these two categories are related to 12 extended categories.

In this example, the tools are helping to handle two different issues in parallel. Both cases involve merging multiple topics in the working set into a single, user-defined category by defining and attaching titled color tags. In one case, different categories containing COTS suppliers of GPS components are rolled together under the title of "retailers." At the same time that topics are being examined and appropriate ones merged into that category, the user is also creating another category that merges different topics concerning use of GPS in agriculture. These two cases illustrate a very common situation: the team initiator and potential team participants have not categorized themselves along the same lines. While the team initiator, for example, is thinking in terms of GPS *technologies*, potential sources instead classified themselves in terms of their intended *markets*.

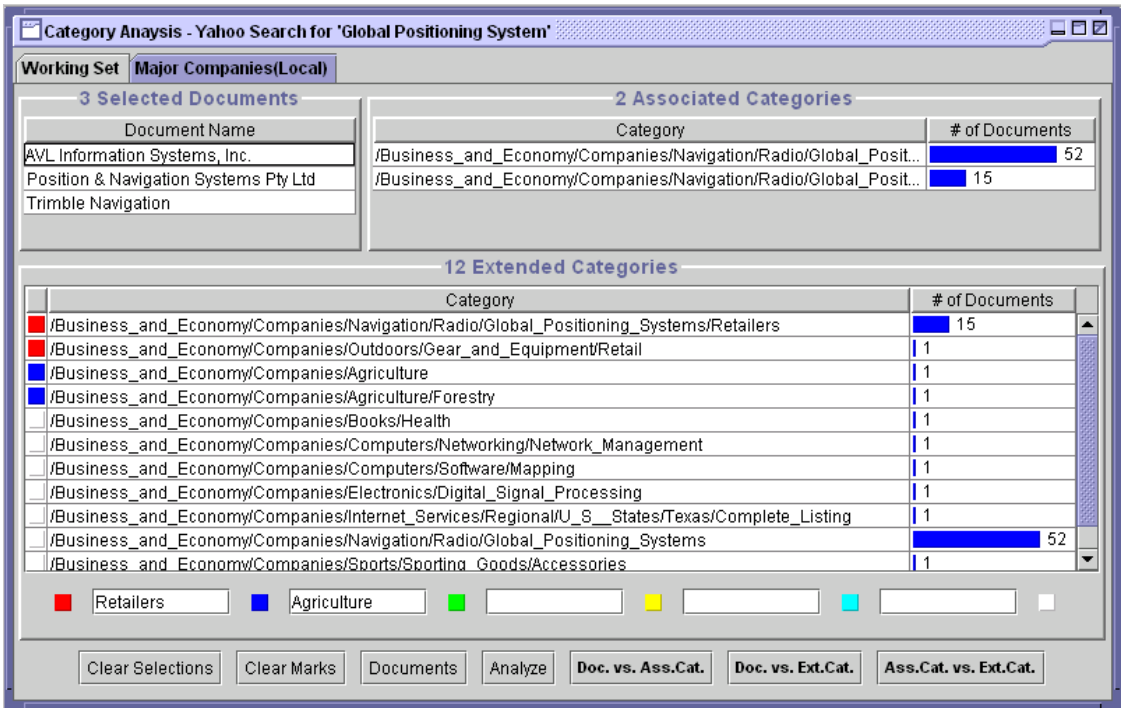

Figure 5. Extended Categories

2.5. Saving and exchanging information spaces

As users proceed through an information analysis, they develop an *information space,* which is composed of several views of the working set of documents and results of analyses. Users can save their information spaces to the persistent store at the server side, and giving the information space a name and a description when they do so. The saved information space can be accessed by any user joining the same DASHER session and can be updated synchronously.

DASHER provides clipbooks to exchange information space data between client components and between different DASHER sessions. For example, a user can import a part of the clustering results into the 'GPS' category editor if the clusters introduce new categories for the document set. Alternatively, the user could export a set of documents classified under the 'agriculture' category to another DASHER session in which other users are gathering information about current technologies related to agriculture and forestry.

A *clipbook* can contain multiple clipboards and sub-clipbooks; users can organize a hierarchical structure of clipbooks. There are two types of clipbooks: local and global. *Local clipbooks* are only visible within a DASHER session, while *global clipbooks* are accessible by all DASHER sessions. Data exported to the local clipbooks will be lost after finishing the DASHER session. However, all global clipbooks are saved in the persistent store in the DASHER server and managed by the clipbook server.

3. Component Services for Defining Categories

The capabilities described above rely on a number of component services. This section describes some of those component services in more detail. These components, respectively, help users understand implicit categories in their information space, identify useful explicit categories, and use those categories to further structure their information space.

3.1. Extraction of implicit categories

The World Wide Web is an enormous repository, filled with such a rich variety of information, but it is often difficult for users to digest. For example, in the case being used in this article, one part of the product development tiger team that is being formed might be tasked to focus upon design or supply of battery components for the device. Searching the Web with the keyword 'battery' yields thousands of pages, far more than can be easily examined. We need to know more. Where are

these pages from? How many types of batteries are currently popular? Who manufactures them? Who sells them? Where are they sold and manufactured?

In this section, we describe methods that can extract product and service categories implicit in the data in order to answer the kinds of questions posed above. This can be accomplished without relying on *a priori* knowledge or regularity assumptions on the structure of Web pages. The methods take advantage of the amount of keyword search data returned, making use of statistical correlation analysis, textual analysis, and artificial intelligence techniques.

Four major mechanisms for extracting implicit category types are described below:

a) Building citation indices

By extracting phrases, such as general noun phrases [Chen et al., 1994], place names and company names, mentioned in Web pages retrieved through a keyword query, we can create a list of phrases referenced in those pages. Then, using statistical techniques, we can build up a ranked citation index that lists the most frequently mentioned phrases. The ranked citation index constructed using general noun phrases provides a good overview of the results of the query. The place names and company names citation indices provide places and companies strongly correlated with the query. For example, if the keyword query is the word "battery", then the top ranked entries in the company name citation index contains the most popular battery-related companies. We can also use artificial intelligence techniques to cluster those web pages according to the extracted phrases, as described below.

b) Discovering subcategories

This extraction suggests potential subcategories of a certain product. For example, given the document set returned from the keyword query battery, this method returns lead-acid, acid, and lithium as the most popular subcategories of battery. The subcategories are discovered by searching through the documents in the query result set for words that appear to modify the word battery. The top ranked word modifiers are returned as subcategories. This information extraction method is highly approximate. It may reject true descriptions and admit spurious ones. We rely upon the redundancy of the Web to provide enough true descriptions and rely upon statistical analysis methods to reject the false positives.

c) Building URL link list indices

The ability to cite other pages using URL links is an important feature of the Web. It provides a quick way of navigating through relevant information resources. By extracting URL links from a set of web pages, we can build up a URL link list index. Similar to the company citation index, this index would contain the most frequently cited URL links. The following two examples show how useful such indices can be:

- From a URL link list index obtained from pages retrieved based on a given topic, we can locate the most active web sites about that topic;
- From a URL link list index obtained from pages retrieved based on a company name, we can locate the most active web site about that company, which in most cases is the company's home web site.

d) Clustering search results

Using the SOM clustering algorithm and software from University of Arizona [Chen et al, 1996], we are able to group the result pages based on the noun phrases extracted. We can apply the SOM algorithm on company names, place names and general noun phrases, which will generate three different types of clusters for the result pages giving the user different perspectives of the information space. The generated clusters are viewed in a 2D display, which emphasizes the relation between adjacent clusters. Depending on the number of resulting pages from the keyword search, the SOM clustering algorithm can generate hierarchical clusters, which subdivides large clusters into smaller ones.

3.2. Extraction of explicit categories

Users can leverage others' work in analyzing Web documents either by retrieving pre-defined categories from COTS Web search engines or by accessing user-defined categories from bookmark files made accessible by other users. By looking at these explicit categories, and at documents classified in those categories, users can identify pre-defined categories used by others which might be useful in structuring their own space.

COTS Web search engines like Yahoo and LookSmart return related category names along with a list of Web sites as a result of keyword-based search. The result pages of those Web search engines have similar structure, and their contents can be transformed into relational data. A *tuple* in the relational data is composed of a URL, a link name, and a description, and can be mapped into category or Web site information.

We built a *Web Wrapper* to extract the tuples for category and Web site information from search-result pages of such COTS Web search engines. To improve the efficiency, we make use of the parallelism in the Web wrapper by using multiple threads in Java. Our multi-threaded Web wrapper can access several Web pages at the same time and improve the data retrieval speed. The Web wrapper is composed of four components: the query translator, token generator, tuple extractor, and data reformatter.

The *query translator* accepts a query and transforms it into a URL pointing to the result page of a Web search engine. After the information extractor extracts the result-size value, the query translator generates additional URLs, if necessary, to create multiple threads of token generators and information extractors that will access the result pages in parallel. The *token generator* maintains an HTTP connection to a search-result page and generates a sequence of tokens from the HTML page. A token is a word or a phrase or a token delimiter that is not a space character. The *tuple extractor* extracts the category and site tuples from a Web page. The *data reformatter* component provides methods to reformat the extracted category and site tuples into the data types that are accessible by the other services in the DASHER server and by the DASHER client.

To import others' bookmark files, we developed a bookmark file reader. The *bookmark file reader* can read a bookmark file generated by a user using a bookmark editor in a Web browser, convert it into the data format used by the DASHER tools, and import the converted data into the DASHER client. Usually, such bookmark files have a hierarchical structure and can be imported into a category editor and displayed as a structured category tree. After importing the hierarchical bookmark data, users can use the category editor to re-organize the bookmarks or copy and paste a portion into their own category structures.

4. Component Services for Populating a Category Structure

The techniques described above enable a user to import, create, and modify category structures. To assess the resources available to them fully, users need to be able to populate the structures they have created with data instances. DASHER information analysis tools offer two such techniques: category-based retrieval and fan-out.

4.1. Extended category-based retrieval: induction learning

This component service provides two capabilities. First, given a set of categories and a sample of categorized documents, it uses *category-based retrieval* over the Web to find similar documents under each category. Second, it uses the same category-based retrieval technique to partition previously unseen documents into related sets by assigning categories to the documents.

Much effort has been expended on building and populating category structures. For example, Yahoo employs a team of human editors to sift through and categorize Web pages. Our service leverages Yahoo's and other similar manual efforts, in order to automatically retrieve and categorize documents. Conversely, those efforts could potentially benefit from our service. Even with a highly active team of editors, a service like Yahoo is currently able to classify only a small fraction of the Web into its category structure.

Our approach is to characterize a set of categorized documents using word-based rules. Documents under a particular category tend to use the same set of words. The word-based rules attempt to differentiate among categories by finding words that tend to occur in one category, but not in others. These rules can then be used either to find similar documents or to categorize unseen documents. The word-based rules can be generated automatically using machine learning rule-induction algorithms, such as Ripper [Cohen, 1995] and Swap-1 [Weiss and Indurkhya, 1993].

All induction learning algorithms require a set of training examples. In this case, each example is a set of words labeled with the appropriate category. The extraction tools described in Section 3 can be used to obtain categories and documents. However, they have to be preprocessed to make them suitable for learning.

One type of preprocessing is on the categorical level. Documents from several categories may be merged to form one category. Merging is performed either when there are too few documents to make learning meaningful, or when detailed distinction among the categories are not needed. Figure 6 shows two sets of sample rules generated using training examples from Yahoo. In the first set, categories are created by collapsing entire subtrees of Yahoo categories. In the second set, categories are created by merging categories related to particular computer peripherals.

- subscrib ∧ issu ⇒ Magazine
- magazin ∧ public ∧ site ∧ comput ⇒ Magazine
- review ∧ target ∧ inch ⇒ Information and Documentation
- review ∧ featur ∧ america ∧ type ⇒ Information and Documentation
- mailto ∧ fax ∧ sale ⇒ Companies
- mailto ∧ custom ⇒ Companies

- captur ∧ tv ⇒ video card related
- modem ∧ review ⇒ modem related
- scanner ∧ ocr ⇒ scanner related
- monitor ∧ pitch ⇒ monitor related
- storag ⇒ storage related

Figure 6. Sample rules induced from Yahoo computer related categories.

Another type of preprocessing is on the document level. Lexical analysis is performed on the documents to extract the words. Commonly occurring words are removed using a *stoplist* of words, words so pervasive that they are worthless in differentiating one category from another. Then, stemming is performed to collapse morphological word variants to word roots. Stemming should allow the induction algorithms to generate a more compact set of rules, because words used in each rule have greater coverage. To perform these document-level preprocessing we are using a set of algorithms found in the book *Information Retrieval: Data structures & Algorithms* [Frakes and Baeza-Yates, 1992].

The rule-induction algorithms generate word-based rules of the form:

$$\text{If } word_1 \wedge word_2 \wedge \ldots \wedge word_n \Rightarrow category_i.$$

If a document contains $word_1$, through $word_n$, then the document belongs in $category_i$.

This rule representation of categories has two significant advantages over the traditional information retrieval representation of vector spaces [Salton, 1989] and some machine learning representations, such as neural networks. First, word-based rules are easier to understand. Our experience, so far, shows that the Ripper rule-induction algorithm generates about 6 or 7 rules per category with about 3 words per rule ($n = 3$). (See Figure 6.) The small number and size of the rules makes it feasible to have users visually inspect the rules in order to determine if the rules make sense. The typical vector space and neural network representations, which consist of hundreds or thousands of numerical weights, are not as easily deciphered.

A second advantage of the word-based approach is that the rules are readily convertible to Boolean expression queries by extracting the antecedents of the rules. These queries can then be used as inputs to Web search engines, such as AltaVista and Excite. This ability to query search engines offers a quick way of further populating categories with similar documents. This is much more difficult to accomplish with vector space and neural network representations. We are not aware of any Web search engine that allows vector-based queries, that is, queries with numerically weighted words. Neural networks also do not seem to be readily convertible to queries.

4.2. Fan-out techniques for finding similar web sources

When searching the Web, a typical problem facing an information seeker is: "I know one interesting Web site, how can I find 10 other sites offering similar information or services?" For example, the initiator of a product development team might be interested in companies developing navigation equipment components based on GPS technology. He has heard that Trimble Navigation, Inc. makes GPS related products, but he wants to know if there are other companies who are also involved in GPS technology. The component described in this section uses this initial company as a search seed, and fans out from this seed using hyperlinks to find similar companies.

The fan-out technique is based on the general observation that web documents for a specific topic area are often linked to each other through hypertext references – as if people are collaborating together to present information of common interest on the web. It is true that different people often have different views on the same subject, but one would expect that through browsing a large number of web documents, one would be able to get a picture of a "consensus" view on a subject. Recent work by [Brin and Page, 1998] of Stanford and [Weiss *et al.*, 1996] of MIT have shown that the category, or subject topic, of web pages is strongly influenced by the presence of a specific web site link (or links). If used in conjunction with keyword information, this vastly improves the effectiveness of users' search tasks. If more than one web site's links were present in a web document, especially links that point to different web sites, one would expect that these links are correlated in the sense that they involve similar categories. "Spurious" or false correlations should become insignificant if a large

number of web documents were scanned. While the objective of the web search system described in [Brin and Page, 1998] was to find the most relevant web site related to a user query, our approach here is to generate a list of additional web sites that are similar in nature to the example web site given (a "fan-out"). The general approach we use here to find similar or related web resources given an initial example web site is fairly similar to what's reported in [Zhu *et al., 1997*], except here we are dealing with URL links instead of trying to recognize named entities in natural language texts.

Step 1: Information retrieval. In this step, we rely on existing commercial search engines to gather an initial set of web documents that contain at least one instance of the URL link for the example web site. One could also add keyword information to narrow the search if, e.g., the example web site is a company that is broadly diversified but we are only interested in a particular part of its business (for example, we know Rockwell Corporation is in the GPS business, but it is also a conglomerate involved in many other activities).

Step 2: Information extraction. Using the initial set of web documents collected in the previous step, we look for URL links present in the web documents. Here, we are particularly interested in whether a URL link is referencing a web site other than the web site the web document comes from (web sites with the same domain names are not considered as "outside"). Other information of interest here are the "distance" of URL links to the URL link for the example web site, the total number of URL links, whether URL links are form into any distinct groups (an example is a web page that is someone's personal bookmark), etc.

Step 3: Information mining. Statistical correlation analysis is performed here for the information collected in the previous steps. In our current implementation, each URL link (has to be an outside link) is treated with equal weight regardless of the total number of URL links present in the web document. In the future, we might explore different weighting schemes by taking into account the total number of URL links in a page. The reason that one might be interested in doing this is that from web surfing experience, a really "useful" web page doesn't overwhelm the reader with a long list of URL links, but instead the page usually contains just a selected few. This issue has been explored recently by [Brin and Page, 1998].

The final search result produced from these steps is a list of web sites, ranked according to the frequencies (or weighted frequencies if one takes into account factors discussed in step 3 above). These appear together with the original example web site from which the fan-out was initiated. If needed, one could choose web sites from this ranked list to make them new example web sites and, therefore, generate a new expanded ranked list.

5. Collaboration using MVD and Habanero

Work is in progress to provide synchronous collaboration and joint information exploration capabilities by incorporating University of Illinois NCSA's Habanero and UC Berkeley's Multivalent Document (MVD) technologies as components. Habanero is a collaborative framework for remote users to share tasks and information live over the Internet. Using the API provided by the Habanero framework, remote DASHER sessions can exchange mouse events, keyboard events, and computational results. For example, if one user clicks on a category tree node to expand it to view its subnodes, then all the users get to view the expanded subnodes. Or, if one user creates new category tree node, then this new node is created for all the users. This capability not only allows remote DASHER users to share information, but also to pool together their expertise to collaboratively explore the information space.

UC Berkeley's multivalent document is made up of multiple layers of related data and a set of loadable behaviors. In particular, we plan to incorporate their annotation layer and its corresponding behaviors. This layer allows users to mark-up documents by highlighting word phrases, adding hyperlinks, and attaching notes. These capabilities allow users to emphasize key points in the document, provide supporting documents, and attach comments and critiques to the document. With the incorporation of Habanero, this annotation facility becomes even more powerful. With both components in place, remote users are able to share the mark-ups, to carry on live discussions, and save the content of the discussion as annotations.

6. GeoWorlds Application

As a parallel effort ISI is also performing the GeoWorlds project. GeoWorlds is integrating, testing, and evaluating a unique combination of DASHER technology with Digital Library and Geographical Information System technology in collaboration with multiple sources (including the University of Arizona, the University of Illinois at Urbana-Champaign, the University of California at Berkeley, and the University of California at Santa Barbara). Using Humanitarian Assistance / Disaster Relief as a testbed application, GeoWorlds is demonstrating the feasibility of relating geographical information to a corpus of "other" information in documents. The function of the system is to help a user understand facts and events in

relation to space and time. It allows users to take a set of documents, relate them to places and times relevant to their contents, and provide a visual environment for presenting and exploring those relationships.

The GeoWorlds system integrates a number of powerful capabilities:

1. **Customized Repository**: Users can work with the system to readily create a customized repository of documents restricted to those supporting a particular task at hand. In parallel, the system also has a standing geographic knowledge base that contains detailed geo-located information.

2. **Characterization/Partitioning**: Users can browse both sets of documents and, in the process, develop characterizations of them. In doing so, they can simultaneously specify a taxonomy of topics of interest to them and use that taxonomy to describe and organize documents.

3. **Extraction/Piping**: Users can extract (or add as annotations) information about places and times referenced in documents, and link them to geo-locations. This linking will often involve "piping," i.e., chaining together several services.

4. **Interactive Analysis and View Control**: Users can interact with documents and the extended information about them in order to explore relationships between them and lay out presentations that will help them understand connections (e.g., the user might animate the sequence in which reports from different locations came in). Additionally, they should be able to annotate the documents to record reactions and discussions.

5. **Collaboration, Visualization and Animation Substrate**: User interaction with documents and geographic information can be presented using highly graphical techniques to represent relationships. Document displays interplay with terrain maps and other geographical information presentations, so that the user can see which documents or document sets relate to which physical locations, and vice versa. This takes place in an environment which supports multi-user collaboration.

A Scenario

[The kinds of situation in which GeoWorlds applies are scenarios like the following:]

> *A hazardous chemical incident has occurred. An agent released near a harbor begins spreading inland, impacting a heavily populated region containing both military and civilian elements. Using GeoWorlds in a distributed collaborative fashion, a cadre of cooperating military and civilian authorities work to understand and respond to the crisis.*
>
> *Using the geographical information, and automatically pulling on other information, this package computes the "plume," i.e., the growing, moving toxic cloud that will drift across the vicinity over time. This is displayed for users in both static and animated presentations. Examining those presentations, users seek to understand the impact on the region. Using tools for conversion from geo-locations to text references, the geographic indices are converted to textual geographic references and an initial set of documents is retrieved using those references.*
>
> *The initial set of documents is refined, filtering out uninteresting information sources and adding in other additional sources, and is structured around ad hoc topics of concern suggested by different participants in the crisis management team. Using extraction and piping tools from multiple participants, geographic references in the documents and annotations are obtained. These are converted into geo-locations, and mapped in the geographic world. This allows the team to check the feasibility of their plans against constraints of geography and time sequencing, e.g., performing and viewing path planning of an evacuation over geographic mapping of the transportation routes to verify that the plan will keep the movements outside the path of the toxic plume.*
>
> *The toxic plume is projected to envelop a major hospital near the harbor. The medical group of the crisis management team activates an evacuation plan to transport patients and perishable medical supplies to a neighboring hospital. The medical group determines that it does not have enough refrigerated trucks to move all the perishable medical supplies. Using the University of Illinois NCSA*

Habanero tool, the medical group contacts the transportation group to acquire refrigerated trucks. The two groups jointly browse the transportation category structure, information prepared in advance in anticipation of emergency evacuations, but they discover that the need for refrigerated trucks was not anticipated. Using DASHER's Information Space Analysis tools, the groups create a refrigerated truck category under transportation category structure, and they collaboratively search the Internet to populate the category. The result of this search is copied to the clipbook to be imported into the medical category structure.

Using the transportation category structure, the transportation group finds a highway congestion conditions Web site maintained by the local government. The highway to the neighboring hospital is currently clear, but the highway passes by a sports arena. Again, using the category structure to find a local calendar of events Web site, the transportation group determines that there is an event that will clog traffic. The transportation group shares this information with the medical group, and using UC Berkeley's multivalent document, the transportation group marks-up an alternate route for the evacuation.

7. Conclusions and Future Work

In the context of supporting collaboration, the DASHER effort may be viewed at two levels: service and information.

At the service level, DASHER provides a deeper way of combining and adding value to existing Web services compared to metasearch services like MetaCrawler and SavvySearch. For example, one component of the DASHER Information Space Analysis Tools is the fan-out search. Both metasearch and fan-out search are able to generate multiple queries based on an initial query. However, the multiple queries generated by metasearch are simply syntactic variants of the initial query. The variants are used to query many Web engines simultaneously. In contrast, each query generated by fan-out search is different. Each query is based on the results of previous queries. In fan-out search, the initial query is an instance of what the user wants to find. Based on this single instance, fan-out search retrieves documents similar to that instance. Similar documents are defined to be documents classified under the same category as the initial instance, or that appear together in the some resource listing. The similar documents can be used to recursively query for more documents. Fan-out search provides a good way of making sense of the information space around the initial query.

Other DASHER tools that add value to existing services include the category analyer and the category editor. The category analyzer adds value to category structures, such as Yahoo, by providing a uniform way of accessing such structures and a better way of graphically viewing them. The category editor is tightly integrated with the Web browser. With a category editor augmented browser the user can not only interactively store and edit Web documents found during browsing, but also analyze the contents of the Web documents to retrieve additional documents.

At the information level, DASHER provides ways of collaboratively structuring information available on the Web. The World Wide Web is a rich source of information, but often the information is too disorganized to be useful for the problem task at hand. One way of organizing information is by structuring it into hierarchical categories. Anticipating how the information consumers are going to utilize the information, information providers sometimes attempt to pre-structure the information. Yahoo and LookSmart are popular Web sites that provide such category structuring. If the users determine such categories are useful, the DASHER tools allow them to import these category structures as a basis for structuring their information.

Often, however, information providers cannot fully anticipate the information needs of the consumers. In such cases, the DASHER Information Space Analysis Tools offer ways to help the information consumers to interactively edit the imported hierarchies. Through the category analyzer, the users can prune and filter the categories. They can regroup categories and create new categories. In addition, the DASHER tools support automatically generating categories through implicit category extraction. The users are then able to organize companies found during browsing by the battery types they sell. Once a category structure is constructed, the DASHER tools offer two ways to quickly populate the structure. One is the previously mentioned fan-out search, and the other is category-based retrieval. Category-based retrieval is able to succinctly characterize a category with word-based rules. Then, these rules can be used to query Web search engines, such as AltaVista and Excite, to retrieve more documents.

There is an inherent interest in collaboration between information seekers and information providers: each has a strong desire to make contact with the other. DASHER operates in today's world, where the collaboration takes place with little direct communication between the two sides. The providers structure the information by guessing at the needs of the consumers. Information seekers restructure the information without informing the providers. The current suite of

DASHER Information Space Analysis Tools focuses upon bridging the gap between the two groups by providing means for information seekers to better understand and structure the offerings of providers. The tools are, however, largely oriented toward supporting the seeker side of the interaction.

In future work, still focusing on team formation, we plan to move to supporting closer, more evenhanded collaboration between information providers and information seekers. To this end, there are three new directions we intend to explore. First are mechanisms by which the frames of reference developed by team initiators can be fed back to inform potential providers. Second are mechanisms giving information providers greater flexibility in presenting their information to accommodate the seekers. Third are tools that support negotiation over the exact nature of services to be provided by various team members, through structured discussion over the documents placed in a team's joint information space.

8. References

[Brin and Page, 1998] Sergey Brin and Lawrence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", to appear in Proceedings of the 7th International WWW Conference, April 14 – 18, Brisbane, Australia.

[Chabert et al., 1998] Annie Chabert , Ed Grossman , Larry S.Jackson , Stephen R.Pietrowiz , and Chris Seguin, "Java object-sharing in Habanero," Commun. ACM 41, 6 (Jun. 1998), Pages 69 - 76.

[Chen et al., 1994] HsinChun Chen, Bruce Schatz, Tak Yim, and David Fye, "Automatic Thesaurus Generation for an Electronic Community System," Journal of the American Society for Information Science, 1994.

[Chen et al., 1996] Hsinchun Chen, Chris Schuffels, and Rich Orwig, "Internet Categorization and Search: A Machine Learning Approach," Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries, Volume 7, Number 1, Pages 88-102, 1996.

[Cohen, 1995] William W. Cohen, "Fast effective rule induction," *Machine Learning: Proceedings of the Twelfth International Conference*, Lake Taho, CA, 1995.

[Frakes and Baeza-Yates, 1992] William B. Frakes and Ricardo Baeza-Yates, *Information Retrieval: Data Structure & Algorithms,* Prentice Hall, New Jersey, 1992.

[LookSmart] LookSmart [http://www.looksmart.com/]

[Phelps and Wilensky, 1996a] Thomas A. Phelps and Robert Wilensky, "Toward Active, Extensible, Networked Documents: Multivalent Architecture and Applications", Proceedings of Digital Libraries '96.

[Phelps and Wilensky, 1996b] Thomas A. Phelps and Robert Wilensky, "Multivalent Documents: Inducing Structure and Behaviors in Online Digital Documents", Proceedings of Hawaii International Conference on System Sciences '96 (Best Paper Award, Digital Documents Track).

[Salton, 1989] Gerard Salton, *Automatic Text Processing*, Addison-Wesley, 1989.

[Weiss et al., 1996] Ron Weiss, David Gifford *et al.*, "HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering", Proceedings of the Seventh ACM Conference on Hypertext, March 1996, Washington, DC.

[Weiss and Indurkhya, 1993] S. Weiss and N. Indurkhya, "Optimized rule induction," *IEEE Expert* 8, 6, 61-69.

[Yahoo] Yahoo [http://www.yahoo.com/]

[Zhu *et al.*, 1997] Quan Zhu *et al.*, "Searching for Parts and Services on the Web", Proceedings of International Symposium on Research, Development, and Practice in Digital Libraries, Nov. 18 – 21, 1997, Tsukuba, Japan.