Discovering Internet Resources to Enrich a Structured Personal Information Space

Michèle Ouellet

Stelvio Inc.,
430 Ste-Hélène, #604, Montréal,
H2Y 2K7 Québec, Canada,
mouellet@stelvio.com

Jan Gecsei, Jian-Yun Nie

Dépt d'I.R.O., Université de Montréal
CP. 6128, succ. Centre-ville, Montréal
H3C 3J7 Québec, Canada
{gecsei, nie}@iro.umontreal.ca

Abstract

The Internet is a tremendous resource where one can find documents to enrich a personal information space. The question is: how can one find relevant documents and how can these be organized into an information space? In this paper, we describe a prototype which aims to provide the user with assistance in these two tasks. Our approach assumes the existence of an initial concept structure set up by the user. This structure may contain only rudimentary descriptions for each concept. The system's task is to find relevant documents from the Internet and to insert them in the appropriate places in the concept structure.

1. Information Management for Internet Users

The amount of information available through the Internet is overwhelming; as a result, most of this information goes unnoticed or gets lost again soon after having been noticed. The problem is not new, it is just being exacerbated by two factors: a sudden growth in the number of information consumers accompanied by acceleration of information production. *Information management* has thus become a pressing problem: under this heading come several computing disciplines and activities, most notably authoring of information resources, information access and manipulation, as well as information collection, selection and display. The work described here is concerned with the last three activities.

A user searching for information on the Internet must describe his information need through a query. Then, upon finding a relevant resource, the user may create a bookmark or even a local copy of this resource. But the collection constituted by all these nuggets of information soon grows into a wasteland - for lack of organization - instead of a useful, personal information space. An important property of such a space would be *structure*. This structure could emerge through some automatic processing on the documents in the information space or it could be a manual construct that mirrors the user's mental model of a given topic. The work described here adopts the latter approach and implements an information space built around a course curriculum. Therefore, we assume that the user has already constructed a concept structure, with rudimentary descriptions of the concepts (e.g. a name) and only a few reference documents attached to them. The task of the system is to seek out potentially relevant documents for each concept and to classify them appropriately.

To better explain our approach, let us first analyze the current state of search on the Internet. Figure 1 depicts three users grappling with large amounts of information. The first user spends much time and effort in formulating queries to an information retrieval (IR) service and then in examining the results to determine how relevant they actually are. In the end, he has amassed an amorphous collection of documents. The second user has an assistant that helps with the formulation of queries; he can specify his information need at a more conceptual level; he can also indicate which sites are of particular interest, which sites to avoid, which services should be queried and during what period of time this activity should occur, thus improving network traffic. This user will find it easier to obtain relevant resources; but his information space has no structure and, as a result, the information he has collected is still hard to use. These two cases correspond to the current state of information search on the Internet.

We are working on an environment for the third user in figure 1. This user has formalized his conceptual model for a given topic. A query agent can exploit this model directly in order to find relevant documents. There is another new element in this environment: a classification agent receives the documents ferreted by the query agent and integrates them into the conceptual model or discards them as irrelevant. The reason for this second agent is that a document found by search engines for a given concept is not necessarily relevant to that concept. It may be more relevant to another concept or even totally irrelevant.

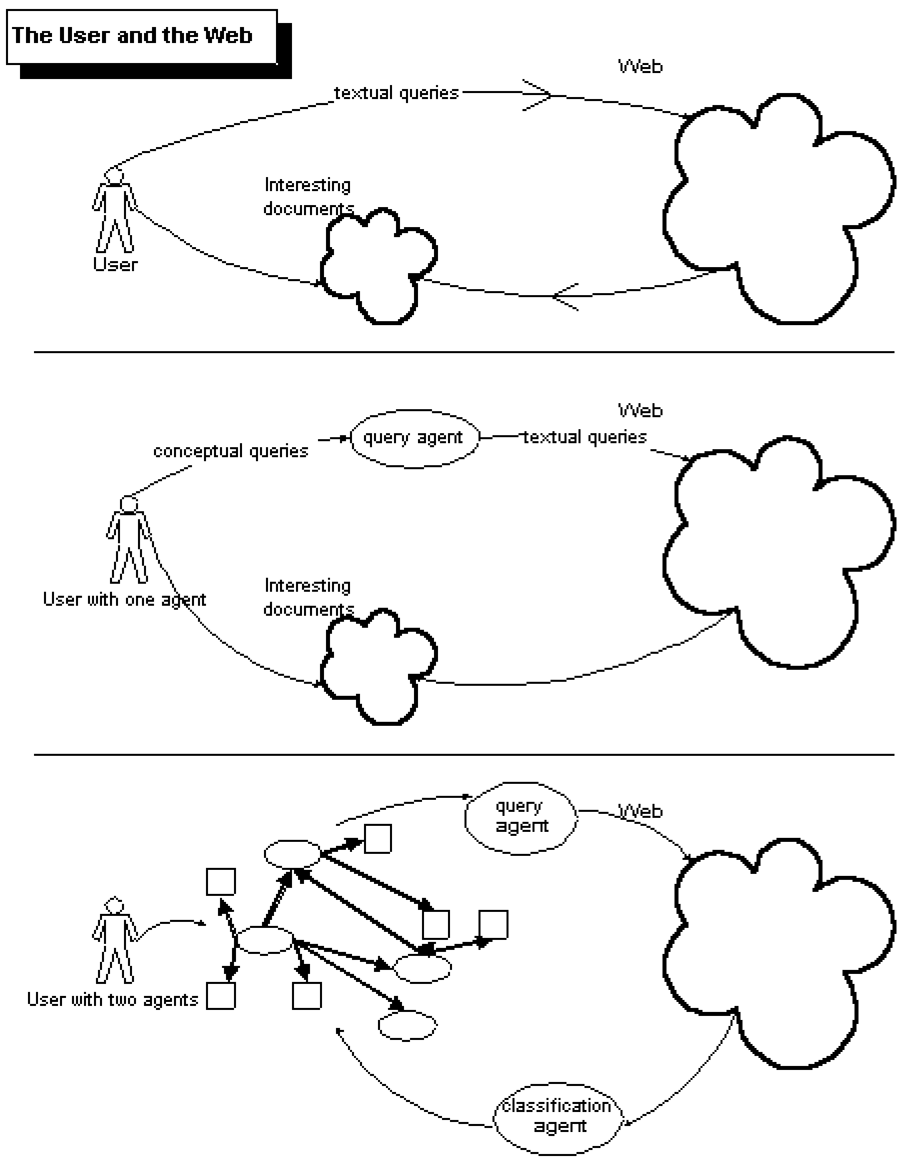

Figure 1: Information Management Support

The type of conceptual model we are using in this project is the Concept Map (Gaines & Shaw 1995): it consists of entities representing concepts and of links between these entities. We did not assume that this model could be built by a general user; rather, we targeted an environment where such models are routinely built and used. Our intended users are intellectual workers such as professors who might build a course plan for the courses they are teaching. We are assuming that this resource, which we call a curriculum, reflects the underlying concept structure for the topic being taught. We are also assuming that this curriculum contains a rudimentary bibliography in the form of documents already tagged as being relevant for such and such a concept.

The remainder of this paper is organized as follows. In section II, we present some relevant work in information filtering; we describe our approach in section III; section IV introduces a Java prototype; we report on evaluation of the prototype in section V; finally, section VI discusses some natural extensions of this prototype.

2. Related Work in Information Filtering

The query problem and the classification problem introduced above have been extensively studied by the information retrieval and filtering community. An overview of the work in textual information filtering systems can be found in (Oard & Marchinioni, 1996); the authors describe information filtering as one of the techniques for the mediation of automatic access to information present on a network. "Text filtering is an information seeking process in which documents are selected from a dynamic text stream to satisfy a relatively stable and specific information need." (Oard & Marchinioni, 1996). This information filtering task can be broken down into three main sub-tasks: collecting potential information sources, selecting relevant ones and finally presenting them to the user. Syskill & Webert (Pazzani, Muramatsu & Billsus, 1996) is a system that handles the first two sub-tasks. The Spring Embedder (Huang, Eades & Wang, 1998) provides an interesting technique for the presentation of large networks of information. We give a brief review of these systems.

2.1 Collection and Selection of Information - Syskill & Webert

Syskill & Webert (Pazzani, Muramatsu & Billsus, 1996) is a tool which plugs into a Netscape browser to facilitate the information navigation process; it supports a user in his navigation tasks by learning a profile of his interests and then exploiting this profile. Syskill can automatically formulate queries to Lycos on topics of interest and scout links emanating from the current navigation page; it presents results to the user in the form of a new HTML page with ratings for the various out-going links. The core operation in Syskill is the construction of a profile. For each user and for each topic of interest, a list of approximately 100 links is presented; the user is asked to rate each link as to its interest, then the system compiles these ratings into a set of features which are fed to various learning algorithms.

Syskill & Webert has some of the functionality we need, in the way of automatic query formulation. However, links get rated rather than classified; furthermore, the level of concept granularity in Syskill is a whole topic. This is not fine-grained enough for classifying documents into a structure of related concepts. Moreover, Syskill has a limited palette of information presentation techniques: it displays documents in a linear fashion. We now report on work dealing with the presentation of networks of information.

2.2 Graphs for Information Presentation

The problems of navigation through a large document universe are well-known (Conklin, 1987) and visual metaphors hold part of the solution. When attempting to visualize a collection of documents, we may want to emphasize the link structure - implicit or explicit - between these documents: these links have the potential to induce a useful navigational mechanism in document space. Whereas some approaches such as GSA (Chen, 1998) solve the problem of automatically creating a link structure, other approaches are more concerned with automatic layout: these techniques include Spring Embedder algorithms and can be used to visualize a large network. The goal of a graph layout algorithm is to assign a position to each node and to route each link. Since this can be seen as an optimization problem, some of the solutions have come from physics, in the form of analogies. Currently, the most important family of algorithms assimilates the nodes to steel rings and the links to springs between those rings; Hooke’s Law is used to minimize a global energy function of the graph, now viewed as a large system of springs. This model was developed first by Eades and then by Kamada and Kawai (Huang, Eades & Wang, 1998). An illustration of such a presentation may be found in Figure 3 (section 4).

While existing information management tools provided us with the Spring Embedder as a graph layout technique for document space display, we could not find a library or system to handle all the querying, classification and presentation tasks we need for curriculum resource discovery and integration. We found it necessary to build our own tools.

3. Curriculum Resource Discovery

The premise of our work is the intuition that it should be possible to instrument networks of concepts in such a way as to discover and integrate information resources from the Internet. There are various ways of construing this discovery and integration of relevant resources. At one end of the spectrum, there are deep knowledge systems; one such system - Cyc (Guha, 1990) - has been estimated to one person-century of development. Then, there are less sophisticated systems and frameworks, like Parka and Shoe (Luke & Hendler, 1997), which rely on markup of resources and exploitation of domain ontologies. Finally, there is room for tools that rely solely on lexical processing and statistics in order to match concepts and documents; most information retrieval and filtering systems belong to this group. We want to explore the limits of this third type of tool.

3.1 Problem Description

The tool we are targeting can tell that a document is about such or such a concept, as opposed to "understanding" what a document says about its theme. We wish for an assistant tool, that could help a knowledge worker find more resources on a topic which he has already mapped out; the map would also serve to organize documents of interest, so that they do not get lost amidst unrelated documents. In order to get a better grasp of what this assistant could be like, we focussed on the context of tutorial authoring environments. In this context, we consider the networks of concepts embodied in the topic structure declared by a curriculum; we use the term *curriculum* to denote a hierarchical course outline, with documents already attached to some of the topics.

Therefore, the task is the following: given a course outline assumed to contain online, bibliographical references for some or most of the topics, build a semi-autonomous tool that will discover new, relevant resources for this course and integrate these resources into the existing concept structure. To get an appreciation for the kind of processing involved, we went through some manual experimentation using an online tutorial on semiotics (Chandler, 1995). We formulated queries to the Profusion search engine (http://www.profusion.com) and obtained, on average, 60 documents per query. We sifted through these documents in an attempt to weed out the irrelevant ones and sort the remainder in the bins provided by the topic structure. Some of the results can be seen in figure 2. The tedium and complexity of the task clamor for automation.

3.2 The Data

We first need to understand what data we will are dealing with. This exploration of the data should be twofold: it should address the data contents of a curriculum (semantics) and the format for that data (syntax). However, although a standardization effort is underway for tutorial architectures, most notably with the Learning Technology Systems Architecture (LTSA, 1998), there is currently no widespread standard and no public data format for a curriculum resource. So a curriculum is simply a textual object with topics arranged in a tree structure and references attached to some of these topics. Whereas online ontologies, glossaries and thesauri can be found in some fields, we cannot count on their existence in general. Therefore our topics, as given, will be devoid of any deep structure or meaning.

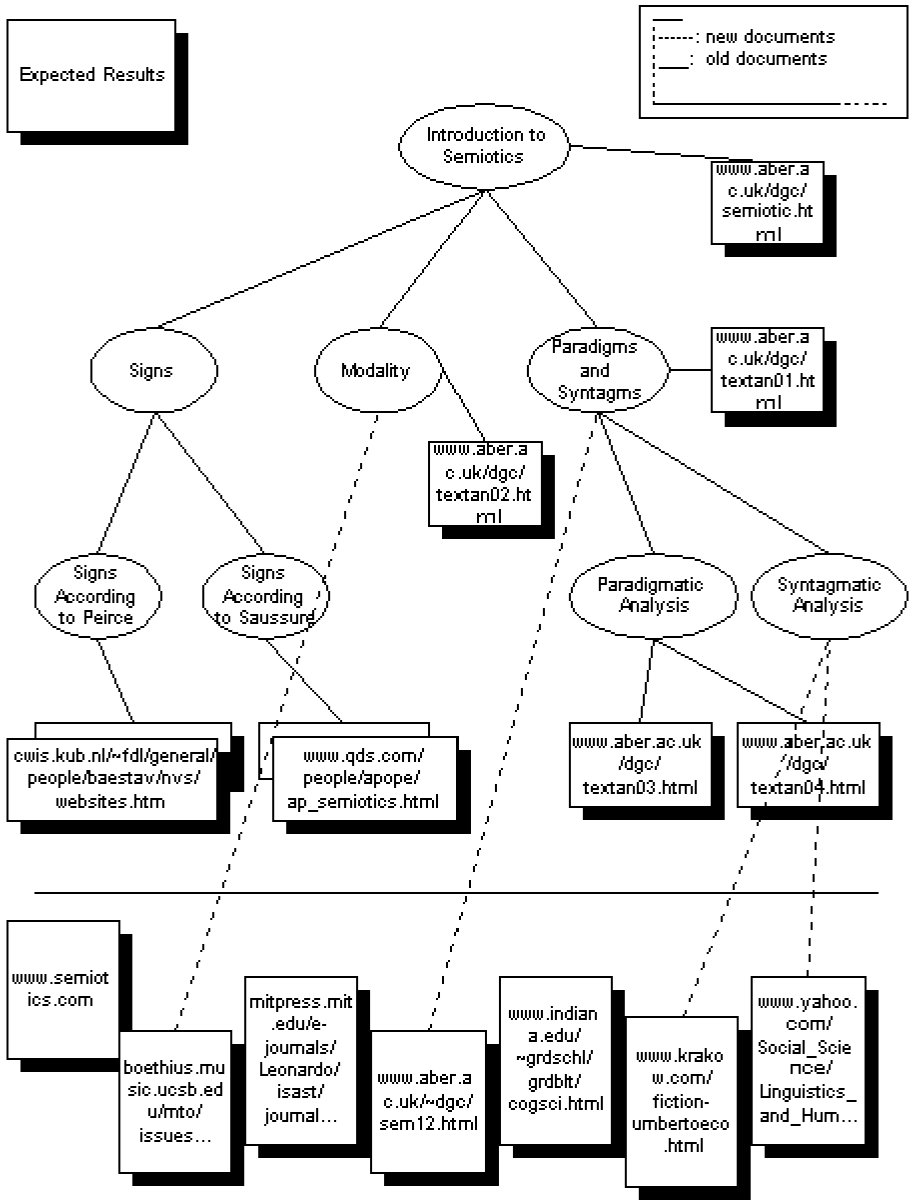

Figure 2 - The Task to be Automated.

3.3 The Approach

If all we had was a concept structure on the one hand and a document universe on the other, the solution would require knowledge acquisition for each new course topic and natural language processing, at the very least. However, with a few attached documents to bridge the gap, we can use information retrieval techniques to find potentially relevant documents for each concept. The approach works as follows:

1. For each concept in the structure, we assign a descriptor based on its name (using a few keywords). We also build an initial representation for the concept, through automatic indexing of already attached documents.

2. A request, formulated using the concept descriptor, is emitted to some search engines.

3. For each new document obtained through the request, we evaluate the similarity between the new document and each concept. The most similar concepts are suggested to the user as possible classifications for that document.

As already mentioned in section 1, a new document is not automatically attached to the concept node we start out with. It may be more relevant to another concept, because the concepts in the structure are strongly related. It may also be relevant to none of the concepts. Therefore, the system has to compute the similarity of the document with respect to all the concepts in the structure. The final decision is made by the user. Once a document is added to a concept, the concept representation changes accordingly.

3.4 Conceptual Model

Before going on to the design and construction of a prototype, we produced a detailed conceptual model. The objective of this step was to assess the complexity and difficulty of the task as we built an inventory of the various mechanisms required in terms of data and processing. The main mechanisms are the initial processing of the curriculum, the generation and handling of queries and finally the rejection or integration of documents in the concept structure. We outline these mechanisms in the paragraphs that follow.

Processing the Curriculum The input concept structure is an HTML file in which the hierarchy of concepts is indicated by HTML tags such as H1, H2. The system first parses the HTML file to build an internal tree representation of the structure. The main example used to test our prototype is a curriculum for an introductory course to the Internet, prepared by professor Jan Gecsei at University of Montreal. A standard indexing process creates a vector for each concept. This constitutes our initial representation.

Generating and Handling Queries In order to find new documents, and since there are already so many search and meta-search engines, we will just interact with those instead of building a new one. So our problem is to generate appropriate queries to feed the search engines. This includes the following steps:

- choose terms that are representative of the concept
- organize these terms into a query
- choose a search engine and formulate the query for this particular engine
- launch the query and receive the results
- filter out links that are strictly publicity or that have been judged to be irrelevant
- present the results to the user

Of all these steps, the most difficult is the selection of representative terms for the query. We cannot directly use the representation of a concept as a query because the search engines have their own query language that is incompatible with our vector representation. Therefore, we have to select a set of keywords to form a query. We implemented only the most basic of selection strategies, namely we extracted terms from the "name" of the concept (e.g. "web site design and management"). Some more sophisticated methods include: using a thesaurus to fetch descriptive terms (this strategy is already used by some systems and search engines); using the most frequent terms in the documents already attached to a concept (if any); using the terms that are involved in the title or headers, or that have some typographic emphasis or that are declared as keywords. These latter means are complementary to what we use now: we will explore their integration in future research.

Classifying Documents For each query issued above, the system normally obtains a certain number of documents. How can these documents be integrated into the curriculum? Viewed at a very abstract level, this is a matter of building a model of a document and doing some inference on this model and on the concept structure to determine where it best fits into the structure. However, as already indicated above, we adopt a simple indexing approach to this problem. When we process the original curriculum, we build a model of each concept using the documents that are already attached to this node. When we receive a document for classification, we treat it as a query. It is first indexed into a vector, and compared with all the concepts using the cosine similarity measure (Salton & McGill, 1983). The closest concept is returned. Note again that the comparison of the document with all the concepts is important: even though we launched a query with concept X in mind, there is no guarantee that the search engine will return a document relevant to concept X. Furthermore, once a document is integrated into the structure, its term contents will impact future classification: to avoid perpetuating errors, we only proceed to this integration under user control.

The scheme we just outlined has an important flaw: it assumes that each concept has online documents attached to it already, so that this concept gets represented in the vector space resulting of document indexing. Judging that this was too much of a limitation to impose on a draft curriculum, thus reducing potential usability, we decided to supplement this first, automatic indexing scheme with another scheme involving manually assigned keywords for each topic. Thus a concept which is originally without documents will still be able to collect and attach documents with manual keywords. This left us with the problem of combining two ranks assigned by the system, one based on automatic indexing and the other on manual keywords. We will come back to this issue when we describe our experimentation.

These three mechanisms, parsing the curriculum, processing queries and classifying documents, provide the most basic expression for the functionality we set out to build, namely a tool for discovering and integrating Internet resources. We now present a prototype that implements these mechanisms.

4. A Prototype in Java

In this section, we move to a more physical model of our tool. Using AWT and JDK 1.1 we developed a prototype capable of tackling the query problem and the classification problem. The user can ask the system to enrich a concept or to classify a particular document; he can modify or accept the classification computed by the system; he can also peruse the document and edit the keywords for a concept.

4.1 User Interface

Our goal with the GUI was twofold:

- provide a graphical way of entering commands and getting results;
- offer a visualization depicting the links between the various topics, along with the documents attached to each topic or sub-topic.

This goal was reached through the menu structure and a partitioning of the main window into distinct zones (see figure 3). The left part of the figure shows the curriculum tree. The part on the right shows the results of document collection. The topmost list presents documents retrieved and considered relevant but not yet integrated; the second list offers a view of the concepts and the documents attached to each; finally there is a text area at the bottom for feedback or error messages. The panel on the right is where the user visualizes his current concept structure; the concepts are represented as ovals, the documents as tiny, anonymous rectangles and the links as straight lines. This graph is

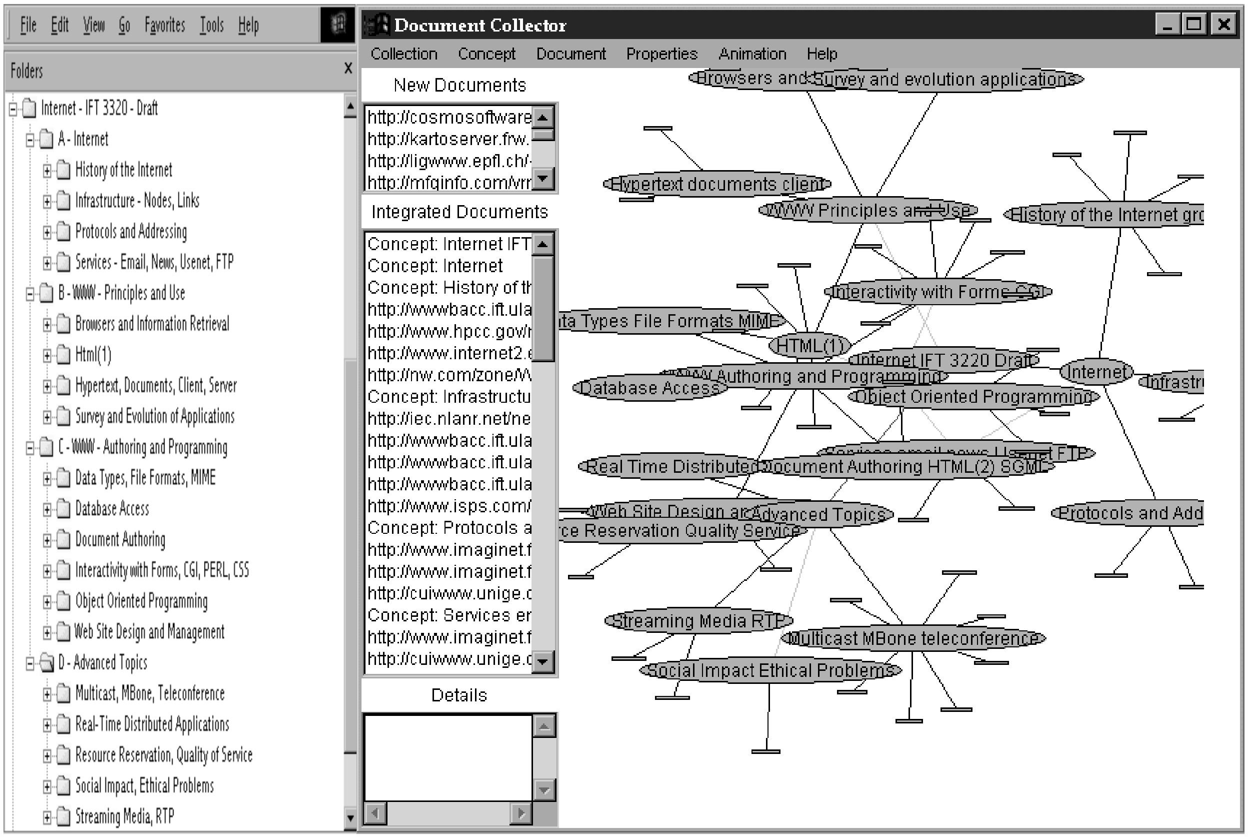

Figure 3: Graphical User Interface

automatically laid out using an animated spring algorithm adapted from the one in class GraphLayout of the Java distribution (Frick, Sander & Wang, 1999).

Although the interface lacks polish, it does give easy, intuitive access to the underlying functionality.

5. Prototype Evaluation

Are the results satisfactory? How could they be improved in the future? How well would this approach carry over to another curriculum, or even to a loosely structured topic map? To answer the first two questions, we have done a manual evaluation on the two sub-tasks, namely on the effectiveness of finding relevant documents using existing search engines (in our case, AltaVista, Inference, Lycos and Profusion) and the effectiveness of classifying a document into the structure. To answer the third question, we put together a pseudo-curriculum, this time on personal financial planning.

5.1 Evaluation of Queries

Given the Internet context, it is difficult to evaluate recall. So we evaluated only precision with a given number of results, as an indicator of quality. We proceeded as follows:

We ran a command-line version of our tool, requesting twenty documents, this for each of the four search services used in these experiments and for each topic or sub-topic in the concept structure. In order to evaluate the queries independently from the classification, we sorted all the documents manually, producing a rating of: relevant for target concept, relevant for another concept in the structure, irrelevant. We then considered, for each query, and for each search engine responding to the query, the proportion of documents falling into each category.

Concept Id	Query (French)	Target Concept	Related Concept	Unrelated
1	Internet	Profusion: 0.1	Profusion: 0.7	Profusion: 0.2
2	Historique Internet	Profusion: 0.0	Profusion: 0.2	Profusion: 0.8
4	Protocoles adressage	Inference: 0.5 Profusion: 0.3	Inference: 0.0 Profusion: 0.3	Inference: 0.5 Profusion: 0.4
5	Services courrier	Altavista: 0.0 Inference: 0.7	Altavista: 0.0 Inference: 0.0	Altavista: 1.0 Inference: 0.3
7	Hypertexte documents	Altavista: 0.3 Inference: 0.2 Profusion: 0.8	Altavista: 0.3 Inference: 0.6 Profusion: 0.2	Altavista: 0.4 Inference: 0.2 Profusion: 0.0
10	Survol évolution	Altavista: 0.0 Inference: 0.0 Lycos: 0.0	Altavista: 0.0 Inference: 0.0 Lycos: 0.0	Altavista: 1.0 Inference: 1.0 Lycos: 1.0
13	Types données	Inference: 0.4 Profusion: 0.1	Inference: 0.2 Profusion: 0.3	Inference: 0.4 Profusion: 0.6
16	Programmation objet	Altavista: 0.6 Profusion: 0.5	Altavista: 0.1 Profusion: 0.1	Altavista: 0.3 Profusion: 0.4
18	Sujets avancés	Altavista: 0.0 Profusion: 0.0	Altavista: 0.6 Profusion: 0.4	Altavista: 0.4 Profusion: 0.6
23	Impact social	Altavista: 0.1 Inference: 0.0 Lycos: 0.6	Altavista: 0.0 Inference: 0.0 Lycos: 0.2	Altavista: 0.9 Inference: 1.0 Lycos: 0.2

Table 1 – Sample from Query Evaluation Results

Excerpts from the results are given in table 1, where the first column holds the Id of the target concept in the concept array, the second column holds the query formulated by the system for the given concept, column three holds, for each contacted engine, the proportion of documents about the target concept; column four holds the proportion of documents relating to another concept in the structure and the last column holds the proportion of unrelated documents.

We averaged these individual proportions to get a precision indicator of 23% for documents relevant to intended concept. While this may seem low, it is understandable in view of the fact that Internet search engines have a precision of 23 to 38% (Hawking, et al., 1999). Moreover, some documents, although irrelevant for the target concept, may still be useful for some other concepts in the structure.

Note that the simple strategy of formulating queries by extracting terms from the concept descriptor is generally effective; there are notable exceptions, however. For instance, the query for topic 10, on the Evolution of Internet Applications, resulted in the retrieval of documents concerned exclusively with the theory of evolution. On the other hand, the query formulated as Advanced Topics surprisingly retrieved a good proportion of documents relating to Advanced Internet Topics, which was the intended topic. This may be due to the nature of the topic, the Internet being naturally Internet-centric in its contents : to shed light on such limitations, we will describe another conceptual universe and its queries in section 5.3.

5.2 Evaluation of Classification

We evaluated classification by asking the system to find the appropriate concept for a set of documents. When running in the mode of pure, automatic indexing - using the set of documents already attached as representative of the concept - the system classified 60% of documents correctly. In order to explore the possibility of classifying documents using manual keywords, we defined a set of keywords for each concept, and used them for comparison with the document to be classified. This second approach gave us an accuracy of 36%. Obviously, this second mode of classification is severely hampered by the difficulty of selecting appropriate keywords.

For the second method, we also tried to allow the user to assign a weight to those keywords or have the system learn appropriate weights through relevance feedback. The tests were not conclusive; nevertheless, this alternate mode of determining where a document belongs is important: it can cover cases where no document is attached. But it should be used sparingly.

Consequently, we combine the two classification methods based on rank rather than similarity value; higher importance is assigned to the automatic classification than to the classification with manual keywords. The strategy used to select the most likely attachment point is as follows. Let A1, A2 be the highest ranking concepts for automatic classification, M1, M2 for manual keywords, then:

- If A1 is not in { M1,M2 } and A2 is in { M1, M2 }, choose A2;
- Otherwise, choose A1.

Examples of classification results using this latter, combined approach are given in table 2. Column 2 holds the Descriptor of the correct attachment point for the given URL. Column 3 holds the Descriptor of the attachment point computed by the system.

Document Descriptor - URL or Title	**Correct Concept**	**System Choice**
A Brief History of the Internet	Internet History	Internet History
History of the Internet and the WWW	Internet History	Hypertext
ISPs.com -- High Speed Modems	Infrastructure	Infrastructure
Survol historique -- client serveur	Infrastructure	Browsers
RealNetworks Home of Streaming ...	Streaming Media	Internet History
Reducing WWW Latency ...	Resource Reservation	Resource Reservation
The Rodeo Group	Resource Reservation	Streaming Media
Conferencing Software for the Web	Multicast, MBone	Multicast, MBone
Real-Time Multimedia Web	Multicast, MBone	Streaming Media
The VRML Repository	Multicast, MBone	OO Programming

Table 2 – Sample from Classification Evaluation Results

This strategy for combining evidence from already attached documents and manually assigned keywords provided a classification which was consistent with a human classification in 64% of cases. Because there is no special emphasis on document titles or headers, and because the size and breadth of coverage of attached documents varies widely, classification errors are not surprising . For instance, document RealNetworks, classified by the system under Internet History, actually belongs under Streaming Media. It was tentatively attached to the Internet History concept because the documents already attached to that concept had a vocabulary that seemed to be similar : a heading on history will, by its very nature, encompass a broad variety of topics and initially attract many documents at classification time. Overall, classification is not totally satisfactory, but since our tool is to be used as an assistant, suggesting rather than deciding, this way of merging judgements has been retained for the time being.

5.3 Application to Another Curriculum

The prototype proved useful in the case of a course dedicated to the Internet. But was it limited to academic subjects? Did it require an expert concept structure? Or could a layman just use it to learn about some other topic? To explore these questions, we built another example : we developed a pseudo-curriculum around the topic of Personal Financial Planning, without prior knowledge; we started with some basic knowledge acquisition from newspapers and from the Internet. We built a concept structure – first materialized as a directory structure – to house the initial URLs that would serve as the basis for the Curriculum object. This is shown in figure 4.

This concept structure is in a very different style from the one about the Internet. It is neither an expert structure nor a teaching structure, it has more of a naive learning flavor. Moreover, the general subject area is less conceptual and more skills-related, as can be seen from the large number of verbs in topic names. Once we had created this structure and attached some URLs, we constructed the actual curriculum using our simple convention of denoting hierarchy through standard HTML markup. We tested our prototype on the financial curriculum, ran queries on all topics and classified some of the newly retrieved documents. This new instrumented curriculum seemed to perform in a manner that was comparable to the previous one, the one about the Internet.

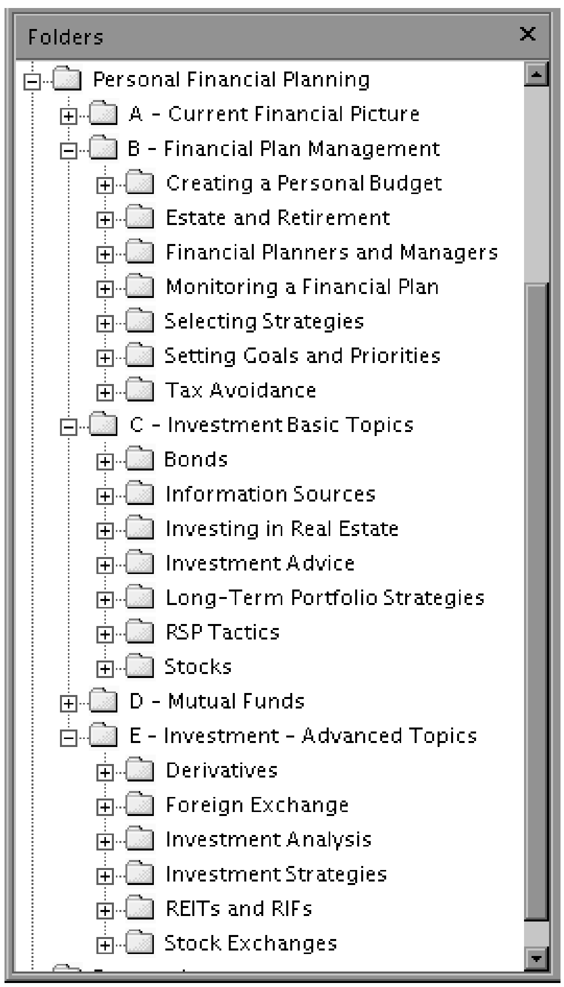

Figure 4 – Personal Financial Planning

We did not investigate performance systematically; rather, the goal of this experiment was to determine two things: whether or not the approach was applicable to other subjects or situations and how much work was involved in setting up the initial concept structure.

We found that this approach is definitely applicable to a wide range of subject areas, as long as there are readily available documents on a subject.

The hard work is in setting up a concept map for a given topic: although easier for an expert or a professor, we found a layman can do it in a few days. Once this is done, authoring the curriculum is straightforward.

6. Discussion and Future Work

There are three types of objects in this system: concepts, queries and documents. For a user, the important ones are concepts and documents whereas queries are just transient objects used to gather more documents for a given concept. However, for the system, the queries are the pivotal point. In this last section, we mention some of the shortcomings in each area and we indicate the direction of our research in that respect.

- **Saturation Problem**: a basic limitation of our approach is that, after a while, a curriculum thus instrumented will no longer be able to discover new documents because the search engines will only respond with documents already examined by the user. The solution might lie in sophisticated operations on queries.

- **Work on Queries**: we only achieved an accuracy of 23% with our automatically generated queries: our accuracy is bounded by the accuracy of the Web search engines to which our queries are sent. However, some steps could be taken to improve the situation, for instance, using a thesaurus and supporting advanced search options (Schatz, et al. 1996; Srinivasdan, 1992).

- **Handling of Concepts**: Our prototype does not support work on concepts. However, we see this prototype as a part of a larger environment, a kind of hyper-editor that would treat the concept world and the document world as two facets of the same knowledge universe. The fusion and fission of concepts and the creation of new links and even new types of links would be supported in this hyper-editor. The environment would be able to detect such opportunities through cluster analysis of the documents and it would suggest these modifications to the user.

- **Document Universe**: While HTML documents currently represent a substantial proportion of all documents accessible on the Web, it seems desirable to accommodate other types of files. Another way of extending our reach would be passage retrieval and link exploration. Passage retrieval (Salton, Allan & Buckley, 1993) provides sub-document access: it allows the user to zoom in on parts that are meaningful to his task. Link exploration, in turn, could be triggered automatically or based on characterization. These extensions do not ease the task of examining and mining this information, however. A true increase in power could be provided through incorporation of some Natural Language Processing, thus giving the user a summary, a characterization and structure extraction, as in MIDS (Helm, D’Amore & Yan, 1996).

- **Scalability and Usability**: We want to support visual interaction in the graph panel, such as customization, selection, description, zooming in and out, showing related concepts, accessing a document from its visual representation and so on. We would also like to improve HTML output, used to describe query or classification results: this output was conceived as a way of bridging the gap between the interactive and the batch mode of operation. Finally, and most significantly, we need access to more than one concept space: this is mandatory if we are to scale the approach and control complexity. Thus we can relate various spaces, travel between spaces, and enrich our queries (Schatz, et al., 1996).

7. Conclusion: Towards the Management of Knowledge

Information collection and filtering tools are already available but not always easy to use. End-user tools for processing information are becoming possible; these tools will support conceptual queries, extraction, mining and structuring. The programming community is making progress towards an InfoSpace (Card, 1996), in which the visualization and navigation of information are the normal way of building and manipulating models for the user's knowledge work. It might be, however, that information systems will become more and more transparent, providing just-in-time information and disappearing behind the task at hand. But, in all cases, information access systems will help users to discover, create, use, reuse and understand information (Hearst, 1997). The present work may be seen as a tiny step in this direction.

Bibliography

Card, S.K. (1996). Visualizing Retrieved Information: A Survey, *IEEE Computer Graphics and Applications*, 16(2), 63-67.

Chandler, D. (1995). *Semiotics for Beginners*, http://www.aber.ac.uk/dgc/semiotic.html

Chen, C. (1998). Generalized Similarity Analysis and Pathfinder Network Scaling, *Interacting with Computers*, 10(2), 107-128.

Conklin, J. (1987). Hypertext: An Introduction and Survey, *IEEE Computer*, 20(9), 17-41.

LTSA (1998). *Learning Technology System Architecture Specification version 4.00*, http://www.edutool.com/ltsa/04/index.html

Frick, A., Sander, G. and Wang, K. (1999). Simulating Graphs as Physical Systems, *Dr. Dobb's Journal*, 302, 58-64.

Gaines, B. and Shaw, M. (1995). *Concept Maps as HyperMedia Components*, http://ksi.cpsc.ucalgary.ca/articles/ConceptMaps

Guha, R.V. and Lenat, J. (1990). Cyc: A Mid-Term Report, *AI Magazine*, 11(3), 32-59.

Hawking, D., Craswell, N., Thistlewaite, P. and Harman, D. (1999). Results and challenges in Web search evaluation, in *Proceedings of the Eighth International World Wide Web Conference*(pp. 1321-1330), North Holland.

Hearst, M.A. (1997). Text Data Mining - Issues, Techniques and the Relationship to Information Access, in *UW/MS Workshop on Data Mining*, http://www.sims.berkeley.edu/~hearst/talks/dm-talk

Helm, D.J., D'Amore, R.J., Yan, P.-F. (1996). MITRE Information Discovery System, in *Proceedings of WebNet96*, http://aace.virginia.edu/aace/conf/WebNet/html/201.htm

Huang, M.L., Eades, P. and Wang, J. (1998). On-Line, Animated Vizualisation of Huge Graphs Using a Modified Spring Algorithm, *Journal of Visual Languages and Computing* 9, 623-645.

Luke, S. and Hendler, J. (1997). Web Agents That Work, *IEEE Multimedia*, 4(3), 76-80.

Oard, D.W. and Marchinioni, G. (1996). *A Conceptual Framework for Text Filtering*, University of Maryland CS-TR-3643, 32 p.

Pazzani, M.J., Muramatsu, J. and Billsus, J. (1996). Syskill and Webert: Identifying Interesting Web Sites, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference* (pp. 54-61), AAAI Press/MIT Press, Menlo Park, http://www.ics.uci.edu/~pazzani/RTF/AAAI.html

Salton, G. and McGill, M. (1983). *An Introduction to Modern Information Retrieval*, New York, McGraw-Hill.

Salton, G., Allan, J. and Buckley, C. (1993). Approaches to Passage Retrieval in Full Text Information Systems, in *Proceedings SIGIR-93* (pp. 49-58), Association for Computing Machinery, New York.

Schatz, B., Mischo, W.H., Cole, T.W., Hardin, J.B., Bishop, A.P. and Chen, H. (1996). Federating Diverse Collections of Scientific Literature, *IEEE Computer*, 29(5), 28-36.

Srinivasdan, P. (1992). Thesaurus Construction, in *Information Retrieval - Data Structures and Algorithms* (pp. 161-176), W.B. Frakes and R. Baeza-Yates (Eds.), Englewood Cliffs, Prentice Hall, 1992, 504p.