

Katsir: A Framework for Harvesting Digital Libraries on the Web

Uri Hanani

Information Science Department, Bar-Ilan University, Ramat-Gan 52900, ISRAEL, uri2@netvision.net.il

Ariel J. Frank

Mathematics & Computer Sciences Department, Bar-Ilan University, Ramat-Gan 52900, ISRAEL, ariel@cs.biu.ac.il

Abstract-The information era has brought with it the well-known problem of 'Information Explosion'. There are many and varied search engines on the Internet but it is still hard to locate and concentrate only on materials relevant to a specific task. Digital libraries, on the other hand, provide better services for focused discovery of relevant Web resources. However, digital libraries have been much less researched and implemented than search engines. The 'Katsir/Harvest' project laid the ground for our understanding that a new paradigm should be developed - the Harvested Digital Library (HDL). The contribution of this article is in presenting a new framework and harvesting model for constructing HDLs. The open harvesting architecture proposed here uses advanced information retrieval tools and provides a set of integrated DL services to its users. This model and architecture are discussed throughout the article, including description of the implemented Katsir system and discussion of future research directions. The future DLs will be knowledge rich in the sense that each DL contains relevant meta-information on its domain and employs advanced knowledge management techniques.

Keywords: Internet, Web, Information Retrieval, Search Engines, Digital Libraries, *Knowledge Management*, *Web Farming*, *Harvested Digital Library*

1. INTRODUCTION

The information era has brought with it the well-known problem of 'Information Explosion'. There are many and varied Search Engines (SEs) on the Internet but it is still hard to locate and concentrate only on materials relevant to a specific task. Also, it is not easy to get unique services usually provided by a regular library, for example: advice in locating materials, a guided tour of an existing data repository, or extraction of metadata, such as title, authors, category, keywords and summary.

Digital Libraries (DLs) could better provide such services on the Web. However, digital libraries have been much less researched and implemented than search engines. In any case, there is a real need to formulate a methodology for efficient construction of both of these types of Web data repositories, and especially of digital libraries.

The 'Harvest/Katsir' project [1] has laid the ground for our understanding that a new paradigm should to be developed - the Harvested Digital Library (HDL). The contribution of this article is in presenting such a framework for constructing HDLs. The open HDL harvesting architecture proposed uses advanced Information Retrieval

(IR) tools [2, 3] and provides a set of integrated DL services to its users [4, 5]. The future HDL architectures will be knowledge rich in the sense that each DL contains relevant meta-information on its domain and employs advanced knowledge management techniques.

This article is structured as follows. The next section concentrates on data repositories and search on the Web by way of contrasting the use and development of search engines and digital libraries. The section following introduces the harvested digital library framework, architecture and system. The before to last section discusses intelligent approaches on the Web and their implications for the next generations of DLs. We conclude the article with future directions.

2. DATA REPOSITORIES AND SEARCH IN THE WEB

Following the introduction, this section aims to provide a general overview of various aspects and issues related to data repositories and information search on the Internet. After contrasting shortly search engines [6] to digital libraries [7, 8, 9], we review their parallel historical and functional evolution.

A. Search Engines vs. Digital Libraries

The SE paradigm and the DL one are really located at the extremes of a spectrum of data repositories and types of Web search. They are two sides to each of these coins: the data repository construction (server) side and the user (client) information search side. We now discuss and contrast these aspects.

As regards to the construction of a SE, this is a complex undertaking. It is clearly a long-term effort that is (eventually) supported by commercial companies. The SE aims to build a quantitative global repository that represents as much information available on the Internet as possible or at least a large amount of it. The SE maintains various data structures to represent its repository, like indices, directories and catalogs. It also provides basic and advanced user interfaces for search purposes. The SE continuously employs various types of robots to search out and index or summarize pages on the Internet and to dynamically update its provided repository. Advanced SEs use a variety of sophisticated artificial intelligence techniques and natural language processing algorithms to gather and organize their contents.

Let us look now at the user side of SEs. Assume that a user needs some information on a certain topic that is currently of interest. So the user summons on a whim his favorite SE to search for any relevant information. The SE is invoked with an ad-hoc query, composed of a supposedly appropriate combination of keywords. The SE will certainly return a lot of noisy information (with low precision and recall) that is bound to overload the user. The user will then have to tediously sift through it all and manually filter the supplied references. The relevant information found will then be immediately consumed or temporarily kept in a cache for a short-term period.

Consider now the process of constructing a DL. A user, e.g., an information scientist, realizes a well-thought out need to build a qualitative data repository on an important focused topic. The information scientist decides to invest by constructing and maintaining a long-term DL, described by a set of specific categories. So he/she interacts with a special interface to carefully define his/her DL request. The DL is then gathered and made available to its users. It supports various data structures to enable efficient keywords search, touring a DL via a topics-tree, and DB/SQL oriented meta views of the DL contents. The contents of the DL are continuously kept current and can be annotated and enhanced with additional relevant material.

Let us check now on the use of DLs. A serious user will tend to often need information on a topic of interest. There is a good chance then that the user already has access to a relevant DL, previously constructed. So he/she invokes the high-level DL interface and chooses an appropriate way to search this DL. The DL will return a reasonable amount of information (with high precision and recall) that the user can readily digest. The returned results could be made available at three levels of detail: first, a high-level summary; then, if requested, an additional abstract; and finally, if relevant, the referenced resource itself will be fetched and presented. Not much sifting will be necessary in any case. The relevant information can be further annotated by the user and later rediscovered whenever needed.

So, to summarize, SEs necessitate a huge organizational effort, provide the user with too much noisy information, but are useful for one-time shots for quickly needed information. DLs, on the other hand, require a modest support effort, excel in quality and ease of use, provide the user with focused information, but have to be made available beforehand. It is important to note that these two paradigms are neither conflicting nor exclusive, but are complementary in nature. In fact, both SEs and DLs have a lot of similarities in their evolution patterns, as described in the following subsection.

B. Parallel Evolution of SEs and DLs

To get a broader perspective (see table 1), it is worthwhile to study how the SEs and DLs tools evolved in parallel through three generations. The first generation of SEs,

referred to here as basic-SEs, were composed, in general, of the following three components:

- 1) Various robots (also called crawlers, ants, worms, spiders, etc.) that roamed the net in search of web resources worthy of reference by the SE.

- 2) Various databases containing metadata [10, 11] on all the referenced resources. These metadata databases could be full-text indices, keyword indices, directories (also called guides, catalogs, weblogs, etc.), or other similar metadata structures.

- 3) A SE interface/tool that enables the SE clients to launch a basic search on the SE's database and get back a list of web pages of (supposedly) relevant resource URLs, or to recursively descend the branches of the SE's topic-tree in search of sites of interest.

Representative first generation SEs [6, 12] include, for example, Alta Vista, Infoseek, Lycos, Open Directory, HotBot, Excite, WebCrawler, Northern Light, LookSmart and Yahoo.

The second generation of SEs, referred to as meta-SEs, put the emphasis on easier methods for the locating of web resources, on procedures for reducing the accumulated results, and on better ranking. Meta-SEs employ a score of basic-SEs that provide the raw search results that are to be merged and ranked by the meta-SE so as to present a unified list of search results to the inquiring user. Representative second generation meta-SEs include, for example, MetaCrawler, SavvySearch, Search.Com and DogPile [12].

The third generation of SEs, referred to here as popularity-SEs, put the emphasis on supporting various basic SE structures, and various advanced techniques and services, such as enrichment of the SE databases by user initiatives and feedback, and on higher quality and faster search results.

For Example, Google and IBM's Clever [13] apply link popularity measures, and DirectHit applies usage and time popularity measures, to determine the relevancy and ranking of Web pages. As another example, FAST provides parallel speedy search services.

This evolutionary process (presented in table 1) paved the way to the appearance of the Portal (or as it is called these days, the Enterprise). One can look on the Portals as an ensemble of different SEs types in one meeting place in order to provide rich and enhanced services to their frequent users.

The DLs field has experienced a similar development pattern over three generations. The first generation of DLs is represented by the Stand-alone DL (SDL). A SDL is a self-contained DL that has all its digital material physically located at the SDL site. Most SDLs represent a single or several local classical libraries whose material was mostly digitized/scanned. Its domain is usually focused and it usually serves its local constituency. SDLs are static in nature, and are not frequently updated. Representative first

generation DLs include, for example, the Alexandria project, American Library of Congress and its National DL (NDL), Berkeley DL SunSITE, ACM DL and the Internet Archive [9, 12].

The emphasis of the second generation of DLs is on Federated DLs (FDLs). A FDL is a collection of several autonomous SDLs that represent heterogeneous repositories connected by a network. It forms a virtual networked library by using network protocols to overcome interoperability problems. A FDL provides its users with a transparent user interface for ease of access to all involved libraries. Representative second generation DLs include, for example, NCSTRL (Networked CS TR Library) and OCLC (Online Computer Library Center) [9, 12].

The third generation of DLs is characterized by Harvested DLs (HDLs). A HDL is a DL that contains only summaries that refer to the distributed data objects. It is usually domain focused, has fine granularity, and provides various metadata structures and advanced library services. HDLs provide a rich environment for applying varied knowledge management techniques. Representative HDLs include, for example, IPL (Internet Public Library) and WWW Virtual Library [9, 12].

TABLE 1
Parallel Evolution of Search Engines and Digital Libraries

Search Engines Generations	Digital Libraries Generations
1st Generation – Basic Search Engine Robots, Crawlers, Indices, Directories, has basic/advanced user interfaces	1st Generation – Stand-Alone DL (SDL) single or several local, classical, self-contained, focused material, digitized or scanned
2nd Generation – Meta Search Engine uses several basic-SEs simultaneously, ranks gathered pages by relevancy	2nd Generation – Federated DL (FDL) several autonomous SDLs representing heterogeneous networked repositories
3rd Generation – Popularity Search Engine uses link analysis and popularity measures to filter and rank the Web pages	3rd Generation – Harvested DL (HDL) contains only summaries and metadata structures; domain focused, of fine granularity

Following the above review, and since SE technologies and tools are well known, we concentrate here on the less researched DL front, and especially on HDLs.

3. THE DIGITAL LIBRARY HARVESTING FRAMEWORK

The 'Harvest/Katsir' project laid the ground for our understanding that a new paradigm should be developed - the Harvested Digital Library (HDL). The following subsections present the researched Katsir system and the resulting logical harvesting model with emphasis on the functional components of the harvesting architecture and their interactions.

A. The Harvest/Katsir System

As our initial basis for the implementation of the Katsir system we chose to use the Harvest system. Harvest (<http://www.tardis.ed.ac.uk/harvest>) [14, 15], developed mainly at University of Colorado, USA, is an integrated set of tools to gather, extract, organize, search, cache, and replicate relevant information across the Internet. With modest effort users can tailor Harvest to digest information in many different formats, and offer custom search services on the Internet. Moreover, Harvest makes very efficient use of network traffic, remote servers, and disk space.

The Katsir system (<http://bicsir.cs.biu.ac.il:8088/katsir>) [1] is based on the Harvest system. The system includes various tools designed to gather materials and references, while locating resources, extracting metadata on the documents harvested, and indexing them. In order to enrich the access methods available to the DL user, we developed a topics-tree mechanism. The full implementation of the Katsir system [16] uses the Perl and JavaScript languages. Katsir can be directed to build a focused DL, based on both local and networked harvested materials. And through a user-friendly interface, the user can retrieve information by keywords or conduct a guided tour by browsing a topics-tree that enables hypertext access to relevant materials.

The Katsir system aims to provide an open software architecture for building harvested digital libraries and for intelligent information retrieval. The initial Katsir system was developed and implemented in an educational environment as a response to the unique requirements of the Israeli educational system. This project was part of a drive to enhance and assimilate information and telecommunication technologies in Israel, based on the public Internet.

B. Functional Components

We now present the functional components of the logical harvesting model, as shown in figure 1. The harvesting architecture is composed of the following seven components:

1. **Harvester** - prepares the harvesting request for the HDL. The harvester provides an interface to the information scientist to achieve this goal. The harvesting request consists initially of a DL profile and a list of URLs. The DL profile contains the DL categories, list of keywords and a set of expected stereotypes of the DL. The given URLs represent the most relevant sites that are known to the information scientist.
2. **Locator** - receives the initial harvesting request and automatically or semi-automatically expands the URLs list. The Locator can consult with various SEs and information repositories to expand the harvesting request. It can also enhance the DL profile by using knowledge management techniques.

3. **Gatherer** - contacts the Internet and Intranet providers in order to gather the prospective resources for the HDL. The gathering is done by recursive descent of all the URLs provided in the harvesting request. Gatherers can be composed in a hierarchical manner to enable efficient and less repetitive gathering.
4. **Filterer** - filters the irrelevant gathered documents and passes on only the documents that should be part of the HDL.
5. **Summarizer** - extracts summaries from all the relevant resources and streams them to the Broker.
6. **Broker** - organizes the set of HDL summaries and builds the various metadata structures, such as full index, a topics-tree and relational views of the DL. It can relieve network traffic and solve bandwidth bottlenecks in the Web by using a Harvest Caching Server.
7. **Retriever** - provides the user with an interface for querying, browsing and touring the HDL (see figure 3).

C. The Logical Harvesting Model

We now describe our fully developed logical model for harvesting DLs (see figure 2). The model includes processes (represented by circles), data repositories (represented by rectangles) and auxiliary repositories (represented by parallelograms).

An information scientist who is interested in constructing a DL on a specific domain of interest initiates a HDL harvesting. The initiating information scientist/librarian (represented as ISL in figure 2) invokes the **Harvester** with a harvesting query. The **Harvester** generates the initial harvesting request and passes it to the **Locator**. The **Locator** uses various network search techniques to enrich the initial collection of URLs to be harvested. The next component to be invoked is the **Gatherer**. It uses each top-level URL, in a recursive descent manner, to gather all referenced resources from the Providers (see figure 2), and passes them to the **Filterer**.

The **Filterer** is responsible for blocking the non-relevant documents from reaching the focused HDL. It uses various levels of filtering that all remaining documents have to pass to be considered relevant. A first level, for example, can use 'regular expression' to match query keywords with the URL string tokens. A second level can use statistical techniques on the document itself based on keyword counts and frequencies. A third level might use a Categorizer tool (see figure 2) to classify the document and check if it belongs to the gathered DL categories. More levels or any combination of levels can ensure a cleaner DL devoid of 'noises'.

All relevant documents are passed now to the **Summarizer**. It extracts a summary of the document, and passes a stream of summaries to the **Broker**. The **Broker** organizes the HDL, indexes the summaries, and builds a relevant topics-tree using advanced IR tools for clustering

and categorization. The **Retriever** finally provides the DL user with a user-friendly interface (see figure 3).

To keep the information up-to-date, the HDL is dynamically maintained (see the two dashed cycles in figure 1). The **Broker** is responsible for preserving the HDL contents by regularly invoking the internal refresh cycle. In addition, the information scientist can invoke, through the **Retriever**, the external update cycle to both refresh and enhance the HDL contents.

4. DISCUSSION

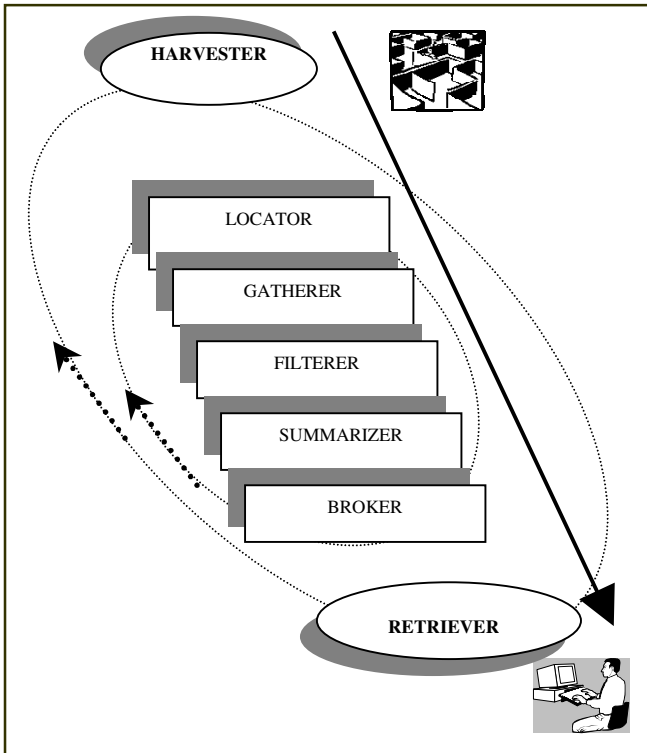
After reviewing the evolution and development of both SEs and DLs, and especially of HDLs, we now discuss the emerging intelligent approaches to construction of data repositories on the Web, and their implications for the next generation of DLs.

A. Intelligent Approaches on the Web

A few researchers have tried to formalize the various approaches used on the net in order to reach a comprehensive methodology for organizing data repositories on the Web. For example, the Web Farming concept [12] views the operations of data and knowledge retrieval, carried out by SEs, as part of an integrated process, thus endowing the knowledge seekers (and the organizations' Warehouses) with a variety of new approaches and tools. In Web farming, the search tasks are replaced with a sequence of actions, as follows:

- 1) Discovering relevant Web content.
- 2) Acquiring quality content that is validated within the domain.
- 3) Structuring the content into usable form compatible with a data warehouse.
- 4) Disseminating the content to the users.
- 5) Managing these tasks in a systematic manner as part of data center environment.

Fig. 1. Seven Components of the Harvesting Architecture



Thus, Web farming reflects the need to better understand the organization of data repositories, thus paving the way towards integrating SEs and DLs into one comprehensive framework. Such efforts can render us with a full picture of appropriate approaches and tools, showing us the way towards next generation intelligent Web data repositories.

As another example, the TetraFusion system [17] supports, what can be called, an Intelligent-SE, using knowledge discovery and data mining techniques on the Web. Another trend is the use of intelligent visualization techniques [12], such as 'The Brain' (<http://www.thebrain.com/>), 'MindMan', (<http://www.mindman.com/>), 'Inxight' (<http://www.inxight.com/>) and 'Semio Maps' (<http://www.semio.com/>).

B. The Next Generation DLs

After discussing the current Katsir framework and the vision of future intelligent DLs, we present here the features expected to appear in the next generation DLs (detailed here by HDL components):

Harvester-Locator:

- Better tools for the automatic construction of the harvesting request and of the DL profile.
- Support for a more semantic environment that enables the information scientist to define in a

higher-level way the desired data repository to be built.

Gatherer-Filterer:

- Dynamic validation and refresh of summaries to ensure HDL contents preservation.
- More semantic filtering of Web resources based also on the DL profile.

Summarizer-Broker:

- Intelligent information extraction from Web resources, thus giving more meaningful summaries.
- A semi-automatic construction of HDL metadata structures such as a topics-tree and a statistical thesaurus.
- Use of advanced knowledge management techniques to support the HDL and its rich integrated services.
- Support for push technology to enable SDI (Selective Dissemination of Information).

Retriever:

- Use of advanced visualization techniques to provide HDL usability and enrich the user experience [18, 19].
- Enhancement of user queries by use of intelligent metadata structures such as a thesaurus (like 'WordNet' - <http://www.cogsci.princeton.edu/~wn/>), concept maps and ontologies.
- Enhancing the user profile and sociological stereotypes based on users' behavior feedback [20, 21].
- Provision of knowledge rich library services, such as consultation, users collaboration, contents annotation [22], organizational workflow and routine ERP activities.

Fig. 2. Logical Harvesting Model

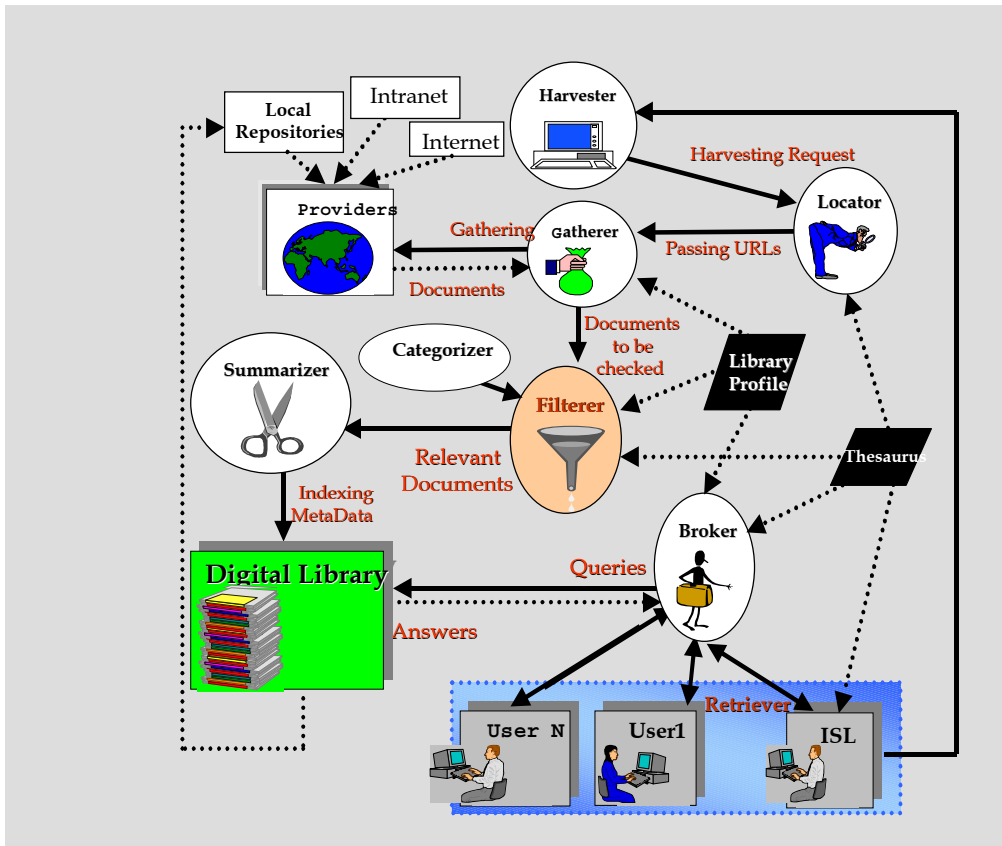


Fig. 3. Katsir Topics-Tree Interface

Harvest

Search

Activate [Navigation Window](#)

Move to [Tree Root](#)

Search all subjects Search this subject only

No Subcategories found

This category contains 7 documents .

- [Bar-Ilan Harvest Project Members](#)
- [The official site for Hughes Technologies products including Mini SQL \(mSQL\), Lite, and W3-mSQL](#)
- [Index to Multimedia Information Sources](#)
- [Taly's Broker - מגשק למאגר](#)
- [The New Zealand Digital Library MELody inDEX](#)
- [mSQL](#)
- [KATSIR - User Guide](#)

5. CONCLUSIONS AND FURTHER RESEARCH

Throughout this article, we have been contrasting SEs and DLs, which eventually led us to introduce the harvesting paradigm and its implications. Consequently, we defined the harvesting framework and its resulting DL harvesting model. This proposed harvesting model has been implemented in the current Katsir system. There is still a way to go though to reach a full implementation of the intelligent DL harvesting model.

Much research is still needed in how to best integrate the various techniques proposed for search engines [23], digital libraries, information retrieval, warehousing, artificial intelligence and knowledge management. For example, a clear challenge here is to better compose separate HDLs into a virtual HDL that also includes integrated metadata structures and provides a transparent visual user interface. However, we believe that the proposed harvesting framework and development methodology for HDLs provide us with a better understanding of how to organize data repositories on the Web, and have a high potential to lead us to the next generation DLs.

ACKNOWLEDGMENT

The authors are indebted to all dedicated Katsir project members: A. Abraham, I. Abramov, U. Hasson, S. Katz, I. Miller, A. Mizrhachi, T. Sharon, M. Shtern and H. Weinberger, and to all other involved project students. We also thank Dr. S. Gal for his insightful observations.

REFERENCES

- [1] U. Hanani and A. Frank, "Intelligent Information Harvesting Architecture: an Application to a High School Environment, *Online Information 96*, London: December 1996, pp. 211-220.
- [2] W. B. Frakes, and R. Baeza-Yates (eds), *Information Retrieval*, Englewood Cliffs: Prentice Hall, 1992.
- [3] G. Kowalski, *Information Retrieval Systems - Theory and Implementation*, Boston: Kluwer, 1997.
- [4] M. Lesk, *Practical Digital Libraries*, San Francisco: Morgan Kaufmann, 1997.
- [5] J. Kessler, *Internet Digital Libraries*, Boston: Artech House, 1996.
- [6] C. Schwartz, "Web Search Engines", *Journal of the American Society for Information Science*, vol. 49, no. 11, pp. 973-982, September 1998.
- [7] Special Issue: Digital Library, *Communications of the ACM*, vol. 38, no. 4, April 1995.
- [8] Special Issue: Digital Library, *IEEE Computer*, vol. 29, no. 5, May 1996.
- [9] H. Chen and A. L. Houston, "Digital Libraries: Social Issues and Technological Advances", *Advances in Computers*, Academic Press, vol. 48, pp. 257-314.
- [10] O. Lassila, "Web Metadata: A Matter of Semantics", *IEEE Internet Computing*, vol. 2, no. 4, pp. 30-37, July/August 1998.
- [11] G. Rust, "Metadata: the Right Approach", *D-Lib Magazine*, July/August 1998.
- [12] R. D. Hackathorn, *Web Farming for the Data Warehouse*, San Francisco: Morgan Kaufmann, 1999.
- [13] D. Clark, "Natural Language, Relevancy Ranking and Common Sense", *IEEE Intelligent Systems*, vol. 14, no. 4, pp. 17-19, July/August 1999.
- [14] C. M. Bowman, et al., "Scalable Internet Resource Discovery: Research Problems and Approaches", *Communications of the ACM*, vol. 37, no. 8, pp. 98-107, August 1994.
- [15] C. M. Bowman, et al., "Harvest: a Scalable, Customizable Discovery and Access System", *Dept. Computer Science, Univ. Colorado, TR CU-CS-732-94*, March 1995.
- [16] I. Miller, "Katsir 1.0 User Guide and Documentation", *Department of Mathematics and Computer Sciences, Bar-Ilan University, Israel*, 1996, <http://bicsir.cs.biu.ac.il:8088/katsir>.
- [17] F. Crimmins, et al., "TetraFusion: Information Discovery on the Internet", *IEEE Intelligent Systems*, vol. 14, no. 4, pp. 55-62, July/August 1999.
- [18] D. E. O'Leary, "Enterprise Knowledge Management", *IEEE Computer*, vol. 31, no. 3, pp. 54-61, March 1998.
- [19] L. Terveen, W. Hill, and B. Amento, "Construction, Organizing, and Visualization Collections of Typically Related Web Resources", *ACM TOCHI*, vol. 6, no. 1, pp. 67-94, March 1999.
- [20] B. Shapira, U. Hanani, A. Raveh, and P. Soval, "Information Filtering: A New Two-Phase Model Using Stereotypic Profiling", *Journal of Intelligent Information Systems*, vol. 8, pp. 155-165, 1997.
- [21] B. Shapira, P. Shoval, and U. Hanani, "Stereotypes in Information Filtering Systems", *Information Processing & Management*, vol. 33, no. 3, pp. 273-287, 1997.
- [22] A. P. Fraenkel and S. T. Klein, "Information Retrieval from Annotated Texts", *Journal of the American Society for Information Science*, vol. 50, no. 10, pp. 845-854, 1999.
- [23] S. Lawrence and C. L., "Accessibility of Information on the Web", *Nature*, vol. 400, pp. 107-109, July 1999.