The metaknowledge-based intelligent routing system (MIRS)

Ran Giladi a, Chanan Glezer b,*, Nir Melamoud a, Phillip Ein-Dor c, Opher Etzion d

a Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
b Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
c Faculty of Management, Tel-Aviv University, Tel-Aviv 69978, Israel
d IBM Research Laboratory in Haifa, Matam-Haifa 31905, Israel

Received 25 February 1999; received in revised form 11 November 1999; accepted 24 January 2000

Abstract

This paper addresses the issue of locating relevant information in a network of heterogeneous, unfederated information bases of various types, including structured databases, text, audio, picture and video files. The problem is to determine where the required information resides in a network, in locations unknown to the user. The objective is to construct a user-friendly, intelligent, search and routing mechanism in order to find the most relevant information bases in the network. We introduce a mechanism for presenting queries, routing queries, updating knowledge, and learning in a metaknowledge base (MKB). This has been named the metaknowledge-based intelligent routing system (MIRS). MIRS finds the location of the desired information by its ability to “understand” the user’s query and to access information by content, rather than by address. MIRS behaves like a distributed search engine, working with a distributed metaknowledge index-file. There is no need for periodic web-crawling, web-robots, or agents of any sort. The network itself encapsulates the knowledge and routing algorithms that provide the user access-by-content to the relevant information. Contrary to web servers, the specific MIRS servers are not linked by hypertext links, but rather by knowledge links, randomly acquired or expertly built. The system also differs from the usual search engines in that it is capable of handling different types of media (e.g., text, database, multimedia) and applies natural language parsing techniques to understand the intention of the user, as well as potentially use a user-profile to enhance the original query before distributing it over the network. The “metadata” describing the information bases are spread across a network of routing and information servers and are modified as a result of search operations and introduction of new information bases into the system. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Information Retrieval; Network; Internet; Routing; Metaknowledge

1. Introduction

Information is usually retrieved through its physical address. Dictionaries, directories, hypertext links and, lately, web search engines are used to obtain the physical address of the required information. This is usually accomplished using keywords extracted from the user queries. Only
after obtaining the physical address of the information is the information retrieval actually carried out. Information addresses are thus specific, either known from the beginning of the query or obtained through some kind of index assistance.

There have been many attempts to provide support for the information retrieval task. Techniques and approaches include, among others, query languages [11], routing protocols [12], hypertext navigation systems [13], semantic networks [25], network management protocols [8], software-agents [4,9], brokers [4,7,9], metadata management [16,17,22], file search systems [10,20], ontologies [5,18,21], federated and heterogeneous databases [5,6,23], directory systems [26,28] and mediator technology [1,2,12,19,27].

The main network retrieval methodologies nowadays are the World Wide Web (HTTP) [3] with its various search engines, along with other subsystems such as WHOIS [26], Archie [10] and Ingrid [13]. Regarding network intelligence used for retrieval, the most prominent system is Harvest [7], which is a distributed information retrieval system. There are, however, other less known systems oriented towards the same goal such as Glimpse [20], which is keyword-based distributed file search system.

Several other researchers approached the network retrieval problem by reconciling the semantic differences between information sources. They achieved interoperability among ontologies by identifying semantic relationships among terms in different ontologies (e.g., OBSERVER [18,21]), or used a mediating (brokering) layer in order to integrate results from various information sources into common structures (e.g., [2,27]). CAFE [9] is an example of a large document information system that uses brokers to integrate metadata from back-end resource agents. Medoc [4] is an information brokering system (IBS) developed specifically for handling bibliographic and full-text databases. The IBS has an agent-based, distributed and layered architecture. It consists of a user, user interfacing, resource selection, provider interfacing, and provider layers.

With the increasing use of the Internet, two main challenges face the solutions and approaches mentioned above. First, to ensure maximum coverage of the huge number of heterogeneous information bases available over the net; and second, to increase the precision of retrieval. These challenges emphasize the limitations of current search engines and network retrieval systems because any attempt to collect and register addresses of all available information sources over the net using a central mediating broker or indexing mechanism is not a practical solution. Direct, address-based access is not feasible in such a heterogeneous environment and a more decentralized, flexible and scalable approach is needed.

It is evident that a large portion of previous work in the area of network information retrieval emphasized hierarchical and centralized activities such as brokering (mediation) and ontology integration (to achieve semantic interoperability), under an assumption that metadata governing these activities are located in specific known places in the network. Nevertheless, with the rapid expansion in the number of resources available over the Internet, this assumption is not valid anymore and the issue of sharing knowledge about the location of information resources in the network becomes crucial.

In an attempt to overcome the challenges mentioned above, this paper adopts the ideas of mediation and semantic interoperability and proposes a comprehensive architecture of a dynamic, peer-based system for accessing a network of heterogeneous information bases. The network itself maintains and dynamically propagates knowledge about locations of possible information bases using caching and learning mechanisms. Under this approach, specific metaknowledge-based
intelligent routing system (MIRS) servers, much like web servers, route queries to repositories that potentially hold the requested information.

Contrary to web servers, the specific MIRS servers are not linked by direct hypertext links, but rather by knowledge links, randomly acquired or expertly built. “Metadata” describing information bases are dynamically spread across a network of routing and information servers and are continuously modified as a result of search operations and introduction of new information bases into the system. The system uses routing tables that enable it to pass a query from server to server until it reaches its target server, where the most relevant information base is located. The system also differs from the usual search engines in that it is capable of handling different types of information (e.g., text, database, video, image) from a natural language front end. In addition, MIRS provides semantic capabilities to address semantic problems such as detection of synonyms. MIRS uses semantic networks that enable it to standardize and maintain consistent terminology both in the user queries (filtering) and in the routing tables that hold metadata about the information bases.

The main application of MIRS is to serve as an information retrieval tool in a heterogeneous distributed and global environment such as the Internet. However, the system can also be deployed on a smaller scale in order to provide assistance in information retrieval for Intranet users within various organizations. An important requirement for a successful use of MIRS is scalability. The system can be easily deployed in new nodes and must be able to connect them to the existing MIRS network using knowledge links. Another important feature is fully distributed operation in order that a failed node will not affect the systems operability.

The goal of this paper is to review the conceptual model of MIRS [14,15,24] and expand this model by proposing several new learning mechanisms (Section 2). We present state diagrams that serve as a foundation for implementing the proposed model (Section 3) and illustrate the feasibility of the model by means of a prototype (Section 4). The natural language interface for the system is presented in an example and will be discussed in a separate paper in addition to an empirical evaluation of several learning mechanisms.

2. MIRS architecture

MIRS is not a directory system in which metadata items are collected and maintained in some central or distributed server nodes. It is more of a cache system, in which clues to where information bases (e.g., text files, databases, video clips, image files) might be found are dynamically distributed among MIRS servers, which are updated on the basis of experience and are not necessarily consistent.

MIRS is composed of a provider agent or gatherer, called an information node (containing metaknowledge about local information bases – LKB), a broker or routing node (containing essential metaknowledge about information bases – EKB), and a user node which interacts with the end-user. A schema of the MIRS architecture is depicted in Fig. 1.

The LKB is in charge of registering the information bases in the system. It consists of three parts: a keyword/information base relationship table, an administrative information base parameters table, and a semantic network for maintenance of the keywords in the keyword/information base table. Following is a description of these tables:
Keyword/information base relationship table: This table holds keywords that describe the content of the information base. The information base catalogued in an LKB can contain different types of information such as text files, databases, and multimedia (video, audio, image). For small-scale systems the keywords can be extracted manually; however, as the system grows, there is a need to use an automatic registration mechanism.

For text files, the text is scanned and extraneous, non-key terms are ignored. The relevant keywords are registered in the LKB as representing the information base. It is assumed that the more frequently a keyword appears in a text file the better it represents the content of that file.

For databases, the schema information depends on the specific data model. As an example, in the case of a relational model, it is based on the names of the relations and attributes (fields) [11,15]. In the special case of a bibliographic database, with thousands of books and journal titles, the information base can be defined as an individual title and the keywords are extracted from the abstract or standard content descriptors. This is expected to make the retrieval of relevant information bases more effective.

For multimedia information bases, the title attached to a resource is parsed and used for extracting the most relevant keywords that describe the image, video or audio file. In many cases, there is no meaningful information associated with multimedia files. In such cases, metadata would have to be registered manually. We refer to the essential information extracted from all the above types of information bases as “essential elements”.

The elements that describe the database schema in detail are termed “detail elements”. In addition to the schema information described above, there is a list of scope descriptors (i.e., terms) for each database. These descriptors are extracted from the narrative scope and contents of the database.

The keyword/information base relationship table is represented in the form of a dictionary table where next to each keyword is a list of information bases in which it appears, represented by their uniform resource locators (URLs). A detailed schema for the keyword and information base table of the LKB is presented in Table 1.

The field Query_Frequency (QF) in Table 1 stores the number of times a query with keyword KW accessed the information base located at the address URL. The field Semantic_Value (SV) stores the level of importance keyword KW has in information base URL. The field Query_Word
indicates what type of words or phrases can be used to retrieve information base URL using the natural language processor tables (e.g., “show me...” implies video or image, “let me hear...” implies audio). These words or phrases are matched against the query keywords.

Administrative information base parameters table: The second part of the LKB holds administrative parameters about information bases, such as cost of access to the information node (LKB) and a specific information base, size of the information base, access restrictions (for read or write), access time, availability time, reliability, etc. This part of the LKB is kept in a separate table, so that the specific parameters are listed for each information base. A detailed schema for the administrative table of the LKB is presented in Table 2.

Keyword maintenance mechanism: The third part of the LKB includes a software mechanism and a semantic network used to scan the keywords in the table and to reconcile semantic differences by identifying synonyms or splitting keywords that have more than one meaning into several tuples [18,25]. As an example, when this mechanism detects two keywords that mean the same thing they are replaced by their equivalent keyword and the URLs are merged. This mechanism is used both for new keywords added to the table and for maintenance of existing keywords.

The EKB represents the essence of the system at a higher level of abstraction and granularity. As the EKB represents many information bases and information nodes, it must be restricted to include

<table>
<thead>
<tr>
<th>Field name</th>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Primary</td>
<td>String</td>
<td>Physical address of information base</td>
</tr>
<tr>
<td>MEDIUM</td>
<td>Primary</td>
<td>String</td>
<td>Type of information base (text, database, image, audio, video)</td>
</tr>
<tr>
<td>DESCR</td>
<td>String</td>
<td></td>
<td>Description of the information base content</td>
</tr>
<tr>
<td>ACCESS</td>
<td>String</td>
<td></td>
<td>Authorization clearance</td>
</tr>
<tr>
<td>AUTHOR</td>
<td>String</td>
<td></td>
<td>Creator of information base</td>
</tr>
<tr>
<td>OWNER</td>
<td>String</td>
<td></td>
<td>Owner of information base</td>
</tr>
<tr>
<td>D_CREATE</td>
<td>Date</td>
<td></td>
<td>Creation date of information base</td>
</tr>
<tr>
<td>D_REGIST</td>
<td>Date</td>
<td></td>
<td>Registration date of information base</td>
</tr>
<tr>
<td>D_UPDATE</td>
<td>Date</td>
<td></td>
<td>Most recent update date of information base</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>Number</td>
<td></td>
<td>Reliability grade for information base</td>
</tr>
<tr>
<td>COST</td>
<td>Number</td>
<td></td>
<td>Price of accessing information base</td>
</tr>
<tr>
<td>COMMENT</td>
<td>String</td>
<td></td>
<td>Free text for administrating the information base</td>
</tr>
</tbody>
</table>
only the most important information on these information bases. For text files, only the most frequent and essential keywords are used to represent them. Therefore, a certain threshold needs to be set for inclusion in the EKB table. For example, include only the most frequent keywords.

For databases, keywords are extracted mainly from the database schema definitions. For example, in the case of a relational database, the essential information is extracted from the names of the relations; in the case of a semantic model, e.g., ER, it is extracted from the names of the entities and relationships (without including attributes as in the LKB), and in the case of the object-oriented model, it is extracted from the names of the object classes (without methods and attributes) [15].

In addition to the essential elements, we assume that for each information base there exists a narrative description of its scope and contents. This description may be part of the system documentation. We refer to the scope descriptors as “terms” [15]. The names of the essential elements and terms for each information base represent it in the EKB. The EKB itself is constructed as a dictionary table, where for each keyword a list of information or routing nodes, which deal with this keyword is attached. A detailed schema for the EKB is depicted in Table 3.

The field QF stores the number of times a query including keyword KW was routed to an information node or accessed an information base located at URL. The field SV stores the level of importance a given keyword has in an information base/node located at URL.

The user submits queries at a user-node, and receives a list of possible locations for the required data. The list can be a final one, containing all data required for issuing a valid query to the location best suited, or a list provided for manual selection by the user.

There are three complementary processes handled by MIRS:

Querying: Receiving user queries, analyzing them and routing them across the network;
Registration: Extracting and gathering information (as metadata) from the information bases (including multimedia sources), and propagating metadata in the MIRS components; and
Learning: Caching the replies from the information bases and updating the metaknowledge base (MKBs) residing in the MIRS components.

2.1. Querying

The functional diagram of the querying process of MIRS is depicted in Fig. 2. The query initiator is a module that receives a query string from the user (written in natural language), and extracts a list of keywords. The query initiator is implemented using different artificial intelligence techniques including a natural language processor (NLP) and a semantic network (SN). The idea

<table>
<thead>
<tr>
<th>Field name</th>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KW</td>
<td>Primary</td>
<td>String</td>
<td>Keyword describing information base</td>
</tr>
<tr>
<td>URL</td>
<td>Primary</td>
<td>String</td>
<td>Physical address of information base/node</td>
</tr>
<tr>
<td>MEDIUM</td>
<td>Primary</td>
<td>String</td>
<td>Type of information base/node (text, database, image, audio, video)</td>
</tr>
<tr>
<td>QF</td>
<td>Primary</td>
<td>Number</td>
<td>Frequency of keyword in user queries</td>
</tr>
<tr>
<td>SV</td>
<td></td>
<td>String</td>
<td>Importance of keyword in this information base/node</td>
</tr>
<tr>
<td>QUERY_WORD</td>
<td></td>
<td>String</td>
<td>Type of medium expected for keyword</td>
</tr>
</tbody>
</table>
of the NLP technique is to locate potential keywords, media designators, and negatives within a query while deleting all possible extraneous (non-key) terms. The idea of the SN is to standardize the query keywords by feeding the meaningful keywords in the query into a semantic network where relationships between nodes represent semantic relationships between keywords. If two or more nodes match (invoke the same node), the newly discovered node is assumed to possess the meaning of its ancestors and it replaces them. The consultation process of matching keywords (and possibly backtracking) continues until we locate a set of controlled keywords that best represent the user's query. An expert user should be able to control the use and modify the content of the semantic network. For a detailed discussion and an example of applying the semantic network approach to an information retrieval problem please refer to [15,24].

The process of query initiation using any of the methods described above (or their combination), results in a list of keywords that addresses three criteria:

1. content – subjects, names, etc.,
2. media description – textual, multimedia, etc., and
3. constraints – quantity, description, location, administrative parameters, user-profile indicators, etc.

Next, the keyword list is sent from the user node to an attached routing node. At the receiving node, the query is handled by two modules: infobase locator and query distributor. First, the list of keywords in the request is matched against the EKB dictionary table by the infobase locator. For every routing or information node that has a column in the EKB table, the infobase locator counts the number of query keywords that are listed as appearing in that column. The result of this match is a ranked list of relevant information and/or routing nodes. The list is ordered by the number of hits and weighted by semantic values, which are both expected to represent relevance,
so that at the top of the list we find the column of the routing or information node that has the highest relevance to the query keywords.

After the ranked list of nodes is extracted, the next stages are to contact the relevant information nodes, in order to examine their local knowledge bases (LKBs), and, if applicable, also to contact subsequent relevant routing nodes in order to propagate the query further and search their EKBs. This latter stage is handled by the query distributor, which propagates the query across the network. The criteria for propagating the query are to locate the routing or information nodes that have the highest number of relevant query keywords appearing in their column within the current EKB table. The propagation process continues independently from the transmission of replies (information base addresses) to the user. This process is bounded by a time-to-live counter, which is set by the user and limits the number of nodes the query can pass through. Every time the query is routed to a new node this counter is reduced until it reaches zero and the query dies. A detailed example of the routing process using the fields in the EKB table is presented in Tables 4 and 5.

Assume the following query:

“Show me all paintings in the Louvre museum” (keywords identified by NLP are underlined).

We see that the QF and SV are different for the same keyword in each EKB or LKB column. SV measures the importance of a keyword in an LKB or another EKB.

Table 4
Illustration of the routing process (EKB1 routing table – before)

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Routing or information nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LKB1</td>
</tr>
<tr>
<td>“Louvre”</td>
<td>SV = 1; QF = 3</td>
</tr>
<tr>
<td>“Painting”</td>
<td>SV = 1; QF = 2</td>
</tr>
<tr>
<td>“Museum”</td>
<td>SV = 1; QF = 5</td>
</tr>
<tr>
<td>“Paris”</td>
<td>SV = 9; QF = 1</td>
</tr>
<tr>
<td>“France”</td>
<td></td>
</tr>
<tr>
<td>“Belgium”</td>
<td></td>
</tr>
<tr>
<td>“Europe”</td>
<td></td>
</tr>
</tbody>
</table>

Table 5
Illustration of the routing process (EKB1 routing table – after)

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Routing or information nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LKB1</td>
</tr>
<tr>
<td>“Louvre”</td>
<td>SV = 1; QF = 3</td>
</tr>
<tr>
<td>“Painting”</td>
<td>SV = 1; QF = 2</td>
</tr>
<tr>
<td>“Museum”</td>
<td>SV = 1; QF = 5</td>
</tr>
<tr>
<td>“Paris”</td>
<td>SV = 9; QF = 1</td>
</tr>
<tr>
<td>“France”</td>
<td></td>
</tr>
<tr>
<td>“Belgium”</td>
<td></td>
</tr>
<tr>
<td>“Europe”</td>
<td></td>
</tr>
</tbody>
</table>

QF is a counter that records the number of times a query with a given keyword was routed to a given LKB or EKB. The more frequently queries with a given keyword are routed to a certain EKB or LKB hints that this keyword is strongly related to that EKB or LKB.

Assuming that SV and QF are equally weighted and their value is zero whenever not listed in the table, then MIRS calculates the Routing Function (RF) for every column in the routing table:

\[
\begin{align*}
RF(LKB1) &= 0.5 \times (1 + 3) + 0.5 \times (1 + 2) = 3.5 \\
RF(LKB2) &= 0.5 \times (2 + 9) + 0.5 \times (1 + 3) = 7.5 \\
RF(EKB2) &= 0.5 \times (10 + 8) + 0.5 \times (10 + 9) = 18.5
\end{align*}
\]

Assuming that routing is done only to the node with the maximum RF value, then the query is routed to EKB2 and the following values are updated (bold and underlined) in Table 5: QF (Painting, EKB2) = 9; QF (Museum, EKB2) = 10; QF (Louvre, EKB2) = 1.

As soon as an information node locates an information base relevant to the query keywords, the address of the information base is returned to the user via the reply integrator. This module merges results, checks semantic value scores and presents the final suggestion to the user. The user then has an option to initiate a separate query to a selected information base.

The routing process of a query can be executed in two modes: automatic routing or user-assisted routing. Under the automatic routing option, MIRS independently decides which information or routing nodes to contact. Under the user-assisted routing option, MIRS returns to the user with a ranked table of hits after every routing stage and the user is given a chance to select which information or routing nodes will be contacted in the next stage of the process.

2.2. Registration

Another process handled by MIRS is the extraction and registration of descriptive information (as metadata) from the information bases, and the distribution of these metadata among MIRS components.

The functional diagram of the gathering and updating process of the MIRS MKB is depicted in Fig. 3. The modules and relations of the proposed architecture are as follows:

![Infobase registration process](image)
The infobase-keyword extractor “reads” the information base, or, when it is a database, obtains the data and their schema. It then extracts a keyword list representing the above information base including its media description and constraints. The detection of new information bases in such a distributed environment is under the sole responsibility of the LKB. This process can be done manually by end-users or by some automatic mechanism that detects new information bases in an area of the network known by the LKB and then extracts their keywords and records them in the table. Frequent changes in the information bases may cause the content of an LKB to be obsolete. In this kind of system, this is a realistic limitation, which can be reduced by the automatic registration and maintenance mechanisms that will refresh the content of the LKB tables on a periodic basis.

The keyword list is then sent to the LKB-keyword registrar module at the information node. This module updates the LKB table at the information node with the new keyword/infobase links, and passes a subset of the keyword list further to its ‘parent’ routing node. The LKB-keyword registrar also passes the selected keyword list to several other routing nodes that have issued queries to this LKB and are therefore registered in its table. The keyword list is received by the EKB-keyword registrar, which updates the EKB table of the routing node.

The LKB-keyword registrar selects only the most important keywords to be stored in the EKB table to represent the information base. This is done using subjective criteria such as highest frequency in text file, names of relations (without attributes) in a relational schema, and title of a multimedia file.

As the number of keywords in the LKB and EKB tables grow, it might be necessary to construct overlaying indexes and split these tables into several extensions in order to maintain a reasonable level of performance. A standard database management system (DBMS) can provide this functionality and was used in the second version of our prototype (see Section 4).

In addition to the above active registration process, it is possible that update of an EKB table will take place as a by-product of a querying process, when answers to queries are transiting through that EKB. This means that MIRS has learning and caching capability and can gradually expand the knowledge links between the nodes based on replies from previous queries. In the following section we describe several alternative learning mechanisms used to operationalize the third process supported by MIRS.

2.3. Learning

The learning capability of MIRS is tightly coupled with the routing strategy adopted because routing relies on knowledge accumulated in the routing tables through the learning process, which in turn is a function of previous routing decisions. The main added value of the learning capability is that it makes the routing process more focused and enables MIRS to provide better service to its users. By constantly analyzing queries and the way they are handled by the system, MIRS identifies and records areas in the network that are more likely to hold information about certain keywords. As a result, future queries with these keywords are routed directly to the most promising areas of the network.

The following is a description of three alternative learning strategies in MIRS:

Keyword (content) learning. When MIRS is first initialized, the routing table of each EKB contains a pointer to a default routing or information node. Every query received at an EKB is
initially routed to that node. This pointer is implemented by storing a tuple with a “default” keyword and a default routing target in the routing table.

As queries are issued through an EKB and answers are collected, the EKB records direct pointers (in the form of uniform resource locators (URLs)) to the information nodes relevant to the keywords mentioned in each query. As a result, the search becomes more efficient because future queries including those keywords are routed directly to relevant LKBs based on the modified routing tables.

The keyword learning strategy can be classified using the following two dimensions: “keyword extraction method” and “keyword reference method.” The extraction method refers to the amount of information MIRS extracts about each information base when a response is cached. Under the first option, only the keywords from the query for each information base retrieved by MIRS are recorded in the EKB. Under the second option, MIRS records all keywords attached to a located information base. For example, under the first option whenever MIRS locates an information base consisting of an NYC map in response to a query with the keyword “NYC,” it records in the EKB the URL address of the map with the keyword “NYC.” Under the second option, the map’s URL address will be recorded as an information base with all keywords attached to it (e.g., NYC, NY-State, USA, State, East-Coast, JFK, Newark, La-Guardia, NYSE).

The learning process is accelerated by the second extraction method because whenever MIRS identifies a new information base it uses all the keywords attached to the resource and not only those mentioned in the query. However, this option can quickly overload the EKB routing tables and negatively affect performance.

The second dimension (“keyword reference method”) can be either positive or negative. Positive reference requires MIRS to record the keywords that do appear in a certain information base when a reply is cached. Under positive reference, however, MIRS tends to become fixed on certain paths and might miss new information bases. The exploratory learning strategy that is explained below attempts to overcome this limit by randomly routing queries to additional nodes other than those registered in the routing tables.

Negative reference records the information nodes where a keyword is not found. Under negative learning, MIRS has to erase the routing tables on a rolling basis in order to avoid overlooking new information bases that are introduced into the system. The main disadvantage of negative learning is that it might overload the network because if a keyword does not appear in an EKB then a query with that keyword is routed to all the neighbors of that EKB.

Fig. 4 demonstrates a positive keyword learning strategy for a case of 4 EKBS \(A, B, C, E\) and 2 LKBs \(D, F\). At the beginning of the first query, all routing tables are empty and point to some default EKB or LKB. All queries received by a given EKB are routed to the default EKB. The first query is routed along the following paths: \(A \rightarrow B \rightarrow C \rightarrow D\) and \(A \rightarrow B \rightarrow C \rightarrow E \rightarrow F\). After all answers are sent and received by \(A\), the learning process occurs. First, \(C\) records that \(D\) has information about the keywords in the query, \(C\) updates \(B\), and finally \(A\) is updated by \(B\). On the second path, \(E\) records the existence of the keywords in \(F\), \(E\) updates \(C\), \(C\) updates \(B\) and finally \(B\) updates \(A\). The update between \(C\) and \(B\) takes place only after \(C\) receives both updates from \(D\) and \(E\).

As an outcome of this process, a subsequent query with the same keywords is routed directly to the sources of the information \((D, F)\). MIRS creates a shortcut and learns that \(D\) and \(F\) contain relevant information on specific keywords.
Exploratory ("flooding") learning. Every EKB records the chain of EKBs that were visited en-route for every query it receives. This information is passed back to an origin routing node when a response is sent to the user. Such a mechanism allows MIRS to expand the EKB routing table and to learn about the existence of other areas of the network that were not known before. The user issuing the query can control if he or she prefers to use the newly acquired information or simply use the conventional keyword routing strategy. Exploratory learning is used to overcome the disadvantage of the positive reference method that tends to lock on to certain routing paths for a given keyword. This strategy can always be used in conjunction with other strategies in order to explore new potential information routes.

Fig. 5 demonstrates the use of exploratory learning. Nodes A, B, C and D are EKBs and nodes 1 and 2 are LKBs. Before the first query is issued, C knows only about the existence of LKB1. When a query is routed from A to LKB1, C learns about the existence of B. A second query is now routed from D to LKB1 and C immediately records the existence of D. A third query is then routed from A to B and C and from there to D and LKB1. In such case the query finds new information in LKB2 which was not retrieved in the first query. The exploratory learning strategy enables C to learn about the existence of D after several queries without having to create an explicit link from C to D. This mechanism is similar to some of the routing protocols used by routers in the Internet environment. It enables the MIRS network to expand autonomously and dynamically reveal peer servers.

Expert certification learning. Certifying expert nodes on specific topics is another type of learning. In order to enhance efficiency, MIRS evaluates queries and answers passing through the network and identifies routing nodes that are most likely to hold information about certain topics. The underlying assumption is that if queries with a certain keyword are frequently directed to a certain routing node then that node is likely to be considered knowledgeable (expert) about that
keyword. After certifying a routing node as an expert on a topic, all queries related to that topic are routed directly to that node. There are two main methods for identifying expert servers:

- **Keyword-path analysis** – Each user node maintains a log of all queries it issues and the paths those queries pass through the network (based on the responses arriving from the network). By analyzing the correlation between queries and their paths, a user node is able to identify routing nodes that are more likely to be experts on specific keywords. Future queries on these keywords are routed directly to those routing nodes.

- **Keyword-frequency analysis** – Each routing node includes a frequency analysis module that analyses the keywords passing through the node on a periodic basis. A routing node is more likely to be an expert on topics related to keywords that are frequently routed via that node.

The expert certification process (in any of the methods) is preceded by a keyword expansion phase, which takes place at the user node. The goal of this phase is to standardize the terminology of the query and increase the chances of hits. This phase is executed using a combined knowledge base that consists of a NLP and a SN [25]. The user has control over this phase and can affect the keywords that are finally sent to the routing nodes.

To demonstrate the terminology presented above, Fig. 6 depicts a network of MIRS servers including two user nodes connected to a group of routing and information servers. User 1 submits Kw1 = “DELTA” and user 2 submits Kw2 = “LaGuardia” to the respective attached user-node. At the user nodes, the user has an option to replace the keywords from the query with more general terms (keyword expansion). In our example the user1 and user2 keyword expansion modules maintain the following knowledge in the form of a semantic network:

LaGuardia ~ (Airport, Airline, NYC, NY State, USA);
Delta ~ (Airline, Atlanta, Airport, USA, Greek letter),
Atlanta ~ (City, USA, Georgia, Olympics).

The processing of the two keywords supplied by the users produces:
Delta ~ (Airline, City, USA, Georgia, Olympics, Airport, USA, Greek letter),
LaGuardia ~ (Airport, Airline, NYC, NY State, USA).

Next, the expanded queries are routed to EKB3 on two parallel paths. EKB3 only knows about the existence of the information nodes LKB1 and LKB2. EKB3 examines its routing table and decides to route the query both LKB1 and LKB2. LKB1 examines its keyword/infobase table and
finds a relevant infobase holding an airport file. LKB2 performs the same task and finds an airline file. Finally, the locations of these infobases are sent to user1 and user2 directly from the LKBs via their respective user nodes (see reply arrows).

In the keyword-path analysis strategy, users 1 and 2 work individually to identify a correlation between the keywords “airline”, “airport” and EKB3 (assuming that several queries including these keywords were routed to that server over time). Each user node has its own statistical and knowledge-based tools to analyse the correlation between his or her queries and their routes. The users are not connected and cannot directly share knowledge in such a distributed system.

In the keyword-frequency analysis strategy, the frequency of keywords in the queries is analysed and compared to some predetermined threshold stored at EKB3. The most frequent keywords are: USA with 3 occurrences; Airport with 2 occurrences; and Airline with 2 occurrences. As a conclusion, the expert certification module certifies EKB3 as an expert on USA airlines and
airports. The certification is propagated to the neighbors of EKB3 who pass it to their neighbors. Another alternative is to send a notification about the certification directly and selectively to certain routing nodes based on a log of queries that were processed by EKB3. As a result, future processing of queries related to USA airlines and airports will become more efficient because queries containing these keywords will be routed directly to EKB3, which gradually becomes prominent across the network as an expert on this subject.

3. Implementation of a MIRS

In this section we provide a general design specification that served as a basis for implementing the MIRS conceptual model as a prototype. The prototype is based on the following independent modules that handle a query. Modules 1–3 cooperate in order to provide the functionality of the user-node as presented in the conceptual model.

1. **Query initiator (QI):** An entry point to MIRS. There can be a number of such entry points. QI filters and analyses the keywords provided by the client using the semantic network and the natural language processor.

2. **Reply integrator (RI):** Collects all replies from LKBs (per query) and presents them to the end-user.

3. **Client:** An applet running inside a browser. It connects the end user with the QI using a standard browser and depicts the final results delivered by the RI.

4. **EKB:** The routing node of MIRS.

5. **LKB:** The information node of MIRS.

Fig. 7 depicts the relationship between the QI and client modules. In step 1, a user submits a request for an entrance page to the web server. In step 2, the page is sent to the user who enters a query. In step 3, the query is pre-processed (using the natural language processor and the semantic

![Fig. 7. QI-client relationship.](image-url)
network) and the search is activated with the filtered keywords. The query is then routed across
the network and the results are depicted by RI as a set of relevant URLs. In steps $X_1, \ldots, X_n$, the
URLs are sent to the client. In step $X_{n+1}$, the QI notifies the user that the search process has
terminated and all updates and learning across the query’s path have completed.

The relationship between the QI and EKB modules is depicted in Fig. 8. In step 1, an EKB is
activated using a port and IP address. In step 2, a MIRSMessage object is duplicated from the QI
to the EKB. This object is used for transmitting knowledge among MIRS servers (e.g., keyword–
URL hits). The EKB then handles the query (see later) and in step $X_{n+1}$ notifies QI (which notifies
the client) that the search and update processes have terminated and that all EKBS have submitted
their findings and updated their routing tables (the learning process).

The relationship between an EKB that receives a query and other EKBS or LKBS across the
MIRS network is depicted in Fig. 9. In step 1, the most promising neighboring EKBS and/or
LKBS are selected according to the routing tables of the initial EKB. These nodes are activated via
a web server. In step 2, after a direct link is established, the MIRSMessage object is duplicated to
the receiving EKB and/or LKB. In step $X$, an answer is sent to RI from an LKB, or alternatively,
the EKB passes the query further to other EKBS/LKBS (goto step 1). In step $X + 1$, the LKB
passes the updates of keywords and signals that its search has terminated. In step $Y$, the EKB
sends back its updates of keywords (after its search is terminated) and signals that event to its peer
EKB. In step $Z$ (where $Z > X + 1$ and $Z > Y$) the origin EKB which initially received the query
sends a message to QI, or its previous node, with updates of keywords and a signal that all its
descendants have terminated their search.

Fig. 10 depicts the final part of the search process in which the user receives a list of possible
locations found by MIRS (step 1). The user selects a preferred location and uses standard
browsing operations to retrieve the located information (step 2).

4. Illustration of the MIRS model

The feasibility of the conceptual model was demonstrated using a prototype that evolved in two
stages. In the first stage, an initial prototype was developed using Java running over TCP/IP in a
Fig. 9. EKB–EKB and EKB–LKB relationship.

Fig. 10. Final information base retrieval.
very large telecommunications lab with many networks, web-servers and workstations. An alternative approach for constructing the prototype was to use the Lightweight Directory Access Protocol (LDAP) [28], which is based on the X.500 directory system. Nevertheless, the Java approach was preferred because of its ability to support network hardware independence, flexibility in implementing the learning mechanisms, and widespread use.

The first prototype focused on demonstrating the feasibility and effectiveness of the routing and search mechanisms and not on the actual performance efficiency in handling large-volume information bases or a sophisticated user interface. The extraction of keywords from the information bases was therefore done manually. We installed approximately 25 EKB and LKB servers in the network representing several different organizations. Altogether, we had over 50 keywords as descriptors and over 100 information bases catalogued in its LKB tables (including picture, video and audio files). The routing (EKB) and information (LKB) nodes were implemented using flat tables with local IP addresses for each relevant keyword.

To demonstrate the capabilities of MIRS, queries with keywords were issued from a console to an attached EKB. At the beginning of the session, the EKB submitted the query to some default EKB. The system exhibited keyword type learning because as the query was further passed across the network and responses were accumulated, the attached EKB recorded (cached) the keywords and the EKBs where they were located. The keyword type learning was implemented by using the positive reference method.

The next time similar keywords were submitted, MIRS was able to improve its performance by submitting the query directly to the most relevant EKBs and shortening the path of the query. As a result new information catalogued during the search session on one LKB became gradually available to the whole MIRS network. These results provide evidence of significant saving in network resources (mainly routing processing) as the size of the MIRS network grows.

During the second prototyping stage, another prototype was developed for information retrieval over the Internet in a format similar to standard search engines. This prototype complemented the first prototype because it emphasized aspects of performance efficiency in handling large-volume information bases and a standard, user-friendly, web-based interface. The prototype was implemented using negative reference learning. It incorporated several technological enhancements such as: a natural language processor; use of a database management system to improve efficiency in capturing the keyword/information base links; an automatic registration module capable of reading html and text files and extracting descriptor keywords; and an ability to present the relevance of located information bases by ranking them according to the number of keyword hits.

To illustrate the interface of the second prototype, Fig. 11 depicts the query screen of this version. The user enters the query at the top of the screen. Feedback-report is used to notify the user about the status of the search. Search-time is a parameter field that specifies a time limit for the query after which all replies received by the system are sent to the user. Previous-keys store a log of previous keywords submitted during the current session. Type-of-search specifies whether the EKB that receives the query examines the updated learning tables in the cache memory or simply submits the query to all its neighbors without any selection criteria.

Fig. 12 depicts the registration facility screen. In field 1, the user specifies the location of the MS-Access database in which the keyword information base links are stored. Field 2 specifies the
address of an information base or a root directory of information bases that need to be analysed and registered in the system. Field 3 is used in case a local information base needs to be loaded (not via the Internet). In field 4, the user can add some special additional keywords to be associated with each information base analysed in addition to those extracted automatically. Finally, default parameters are used to tune the search by specifying whether or not the registration program should read all lines in the information base (for text and html files), retain suffixes, and delete numbers and stop words. The user has the ability to open a separate window in order to modify the DAT files which maintain the natural language processor knowledge (e.g., stop-words, verbs, numbers, negatives, media identification words).

To demonstrate how the second prototype works, an identical query was submitted twice to a network with one EKB and four LKBs. Appendix A depicts the processing of a query, the learning process and how it affects future routing decisions. The log is divided into four sections called search streams. There are two EKB search streams (for the first and second run of the identical query), and two matching LKB search streams. The EKB search streams depict the query initialization, parsing (extracting multimedia, negative and keyword terms), as well as
the routing and keyword learning effects at the EKB level. The LKB search streams depict the retrieval of locations for relevant information bases.

In the first EKB search stream the query is routed to all four LKBs registered in the prototype because the routing tables contain no previous information about the query keywords extracted (paleontology, sociology, recanati, lahav).

In the second EKB search stream we can see the effect of the learning process. A second identical query is directed only to the LKB called GSBA because the first query did not find hits for any of the four query keywords in all other LKBs. For a more detailed explanation of the prototype behavior please refer to the comments embedded in the log.

In another test, the second prototype was used directly by a group of end users. We loaded into MIRS about 100 html pages including text and graphics that were downloaded from commercial web sites on the Internet. The LKBs included several thousands of keywords and hundreds of images.

The MS-Access database management system, the EKB/LKB nodes and the infobase registration modules worked properly in cooperation and were able to absorb and retrieve a vast number of information bases in a matter of seconds over a standard TCP/IP network. The
successful demonstration provided evidence that learning is not only effective and feasible but also efficient in an environment closer to the scale of the real business world in which MIRS should operate.

5. Future research

Future research on this topic needs to focus on the following two issues:

(1) Evaluation of the system’s performance as a function of the knowledge architecture, routing and learning strategies (e.g., keyword versus exploratory learning). As an example, a user should be able to change different parameters (e.g. time to live, semantic value, routing function) and evaluate their effect on the performance of the system.

One of our plans is to create an inter-university network of MIRS servers where each university will have several LKB and EKB servers for administration and colleges. The system will gradually learn the expertise of every university and will route queries to the gate server of the appropriate university (e.g., medical related queries will be directed only to universities that have a medical school, engineering queries to technological institutes, etc.)

(2) Development of several advanced features to strengthen the model, such as:

- **Interface module to external search engines (XKB):** Enables MIRS to decide when to invoke an external search using standard engines (e.g., Yahoo, Alta Vista), format a query according to their interfaces and integrate the external hits with the information stored inside MIRS.

- **User profile:** Supplement the keywords provided by the user to improve hit ratio. Each user will have a template with his or her interests in different domains. The template will accompany query keywords during the routing process and will make the information base retrieved more relevant to the user.

- **Filtering and refinement module:** Provide the end user with capabilities to reduce the number of irrelevant hits by ranking the relevance of the results. MIRS will update its routing tables and as a result the responses will become more and more accurate.

6. Conclusions

We have presented a conceptual model for MIRS with new learning capabilities. Contrary to web servers, the specific MIRS servers are not linked by hypertext links, but rather by knowledge links, randomly acquired or built by experts. “Metadata” are spread across a network of routing and information nodes and are continuously modified as a result of standard search operations and introduction of new information into the system. The system is flexible, scalable and adaptive and therefore suitable for deployment over the Internet.

The effectiveness and performance efficiency of the model were demonstrated using a prototype that evolved in two stages. During the first stage, an initial prototype was developed in a large telecommunications lab using Java running over TCP/IP. This prototype focused on demonstrating the effectiveness of the routing and searching mechanisms. The prototype includes a network of routing (EKB) and information (LKB) nodes with flat routing tables and a set of information resources catalogued by keyword descriptors.
During the second prototyping stage, a prototype was developed for information retrieval over the Internet in a format similar to standard search engines. This prototype emphasized aspects of performance efficiency in handling large-volume information bases and a standard, user-friendly, web-based interface. It incorporated several technological enhancements such as: a natural language processor, use of a database management system to improve efficiency in capturing the keyword/information base links, automatic registration module capable of reading html and text files and extracting descriptor keywords, and ability to present the relevance of located resources by ranking them according to the number of keyword hits.

The findings obtained from developing and using both prototypes in several query sessions provide evidence supporting the feasibility and performance efficiency of the MIRS conceptual model even in an environment consisting of infobases that hold thousands of keywords.

Appendix A. Demonstration of MIRS prototype

A.1. EKB search streams

FIRST RUN

**Initialization:** Establish a connection to all LKBs

Requesting Connection NEWS database
Requesting Connection GSBA database
Requesting Connection JAVATUT database
Requesting Connection JDCNEWS database

**Query reception:** Natural Language Processor at user node extracts keywords and attributes

CLIENT NO 1 recanati234-1.tau.ac.il/132.66.166.138 1891 request received
SEARCHTYPE mirs only
SEARCH TIME 5
QUERY RECEIVED: please give me all the information you have on paleontology and sociology at recanati and lahav. thanks
MULTIMEDIA TERMS =
KEYWORD TERMS = paleontology sociology recanati lahav
NEGATIVE TERMS =
FINAL KEYS AFTER PARSING paleontology sociology recanati lahav

**EKB table lookup:** All LKBs need to be searched because there is no LKB for which all the keywords in the query do not exist
Learning Base Polled
Key Term sociology not found in Database(s) jdcnews javatut
Key Term lahav not found in Database(s) jdcnews javatut

**LKB table lookup:** Query is routed to all 4 LKBs

JAVATUT polled for paleontology sociology recanati lahav with mediatype nulltype at //132.66.166.138/mirsmain
NEWS polled for paleontology sociology recanati lahav with mediatype nulltype at //132.66.166.138/mirsmain
JDCNEWS polled for paleontology sociology recanati lahav with mediatype nulltype at //132.66.166.138/mirsmain
GSBA polled for paleontology sociology recanati lahav with mediatype nulltype at //132.66.166.138/mirsmain

!-Wait for reply from LKB servers-!

**Keyword learning:** EKB table records the keywords that were not found in LKB news

Learning Base Updated on Return from news
UPDATING learning base: term PALEONTOLOGY not found in NEWS
UPDATING learning base: term SOCIOLOGY not found in NEWS
UPDATING learning base: term RECANATI not found in NEWS
UPDATING learning base: term LAHAV not found in NEWS

**Keyword learning:** EKB table records the keywords that were not found in LKB jdcnews

Learning Base Updated on Return from jdcnews
UPDATING learning base: term PALEONTOLOGY not found in JDCNEWS
UPDATING learning base: term SOCIOLOGY not found in JDCNEWS
UPDATING learning base: term RECANATI not found in JDCNEWS
UPDATING learning base: term LAHAV not found in JDCNEWS

**Keyword learning:** EKB table records the keywords that were not found in LKB Javatut

Learning Base Updated on Return from javatut
UPDATING learning base: term PALEONTOLOGY not found in JAVATUT
UPDATING learning base: term SOCIOLOGY not found in JAVATUT
UPDATING learning base: term RECANATI not found in JAVATUT
UPDATING learning base: term LAHAV not found in JAVATUT

**Keyword learning**: EKB table records the keywords that were not found in LKB GSBA

Learning Base Updated on Return from gsba
UPDATING learning base: term PALEONTOLOGY not found in GSBA

**Keyword learning terminated successfully!**

Now the same query is submitted again to MIRS and we can see the effect of the learning process through the change in the behavior of the system

SECOND RUN

**Query reception**: Natural Language Processor at user node extracts keywords and attributes

CLIENT NO 1 recanati234-1.tau.ac.il/132.66.166.138 1891 request received
SEARCHTYPE mirs only
SEARCH TIME 5
QUERY RECEIVED: please give me all the information you have on paleontology and sociology at recanati and lahav. thanks
MULTIMEDIA TERMS =
KEYWORD TERMS = paleontology sociology recanati lahav
NEGATIVE TERMS =
FINAL KEYS AFTER PARSING paleontology sociology recanati lahav

**EKB table lookup**: Only the LKB named GSBA needs to be searched because the learning from the previous query recorded that LKBs News, Jdcnews and Javatut do not hold the keywords

Learning Base Polled
Key Term paleontology not found in Database(s) news jdcnews javatut gsba
Key Term sociology not found in Database(s) news jdcnews javatut
Key Term recanati not found in Database(s) news jdcnews javatut
Key Term lahav not found in Database(s) news jdcnews javatut
Database news not searched
Database jdcnews not searched
Database javatut not searched

**LKB table lookup:** Query is routed this time **only** to LKB GSBA because this is the only LKB that contains at least one of the keywords: paleontology, sociology, recanati or lahav

GSBA polled for paleontology sociology recanati lahav with mediatype nulltype at //132.66.166.138/mirsmain

!-Wait for reply from LKB servers-!

**Keyword learning:** EKB table records the keywords that were not found in LKB GSBA. The learning is identical to the first query so there is no need for any change in the EKB table

Learning Base Updated on Return from gsba
Learning Base not Updated due to duplication
Report sent back srch2.htm to recanati234-1.tau.ac.il/132.66.166.138

A.2. **LKB search streams**

**FIRST RUN**

**LKB table lookup:** Keywords are searched in the LKB table of News

Connected to jdbc:odbc:news
searching paleontology sociology recanati lahav ... stand by
nofind paleontology sociology recanati lahav
sentback to sendbackresults
inside threaddeath

**LKB table lookup:** Keywords are searched in the LKB table of Javatut

Connected to jdbc:odbc:javatut
push connection opened with server
searching paleontology sociology recanati lahav ... stand by
nofind paleontology sociology recanati lahav
sentback to sendbackresults
LKB table lookup: Keywords are searched in the LKB table of Jdcnews

Connected to jdbc:odbc:jdcnews
searching paleontology sociology recanati lahav ... stand by
nofind paleontology sociology recanati lahav
sentback to sendbackresults
inside threaddeath

LKB table lookup: Keywords are searched in the KB table of GSBA

Connected to jdbc:odbc:gsba
searching paleontology sociology recanati lahav ... stand by
1 paleontology
nofind paleontology
sentback to sendbackresults

SECOND RUN

LKB table lookup: Only the GSBA LKB is accessed contrary to all 4 LKBs in the first query

Connected to jdbc:odbc:gsba
searching paleontology sociology recanati lahav ... stand by
1 paleontology
nofind paleontology
sentback to sendbackresults

References


Ran Giladi received a B.A. degree in physics and a M.Sc. degree in biomedical engineering from the Technion, Israel Institute of Technology, Haifa, Israel, in 1980 and 1983, respectively, and a Ph.D. degree in information systems from Tel-Aviv University, Israel, in 1992. He is the Head of the Department of Communication Systems Engineering at Ben-Gurion University of the Negev, Beer-Sheva, Israel. He co-founded a multinational data communication firm and is involved in the Israel data and telecommunication industry. He also founded the Israeli Consortia for research on network management systems, and served as Chairman of the consortia board of directors. His research interests include computer and communications systems performance, data networks and communications, and network management systems.

Chanan Glezer is currently a lecturer of information systems in the department of Industrial Engineering and Management at Ben-Gurion University of the Negev. Dr. Glezer holds a Ph.D. in MIS from Texas Tech University, and a M.B.A. with a specialization in information systems and B.Sc. in General Science both from Tel-Aviv University. He has published in the Communications of the ACM, Data and Knowledge Engineering, the Journal of Medical Systems, and the Journal of Organizational Computing and Electronic Commerce. His main research interest is in the development and evaluation of architectures to address organizational coordination tasks such as meeting-scheduling. He is also working on topics such as interoperable electronic product catalogues, data integrity, integrated hospital information systems and methods for achieving management control over the IS function. He has over eight years of MIS-related work experience with the Israeli Defense Force, serving as an application programmer, systems designer and MIS officer.

Nir Melamoud holds a B.Sc. in Computer Science from Ben-Gurion University of the Negev. He worked as a research assistant at the Department of Communication Systems Engineering and assisted in designing, implementing and testing the MIRS prototype. He is employed as a software engineer in the Internet software industry.

Phillip Ein-Dor obtained his Ph.D. from Carnegie-Mellon University in 1971 and from 1973 has been member of the IS faculty in the Faculty of Management, Tel-Aviv University. Since 1992 he has been Director there of the Marcel and Annie Adams Institute for Business Management Information Systems. His research Interests are: Theory of Information Systems and Information System Research, Natural Language Processing, Artificial Intelligence, Information Technology and Infrastructure and the Internet. Dr. Ein-Dor has been a Council Member of AIS and a member of the Executive Committees of ICIS and ECIS. He is Co-Chair of ICIS 2003 to be held in Jerusalem, Israel. In 1998 he has appointed founding editor of the Journal of the Association for Information Systems which began publication this year. He has been an Associate Editor of MIS Quarterly, a member of the Editorial Board of the Information Resource Management Journal and is currently a member of the Advisory Board of Information Technology and Management and of the Editorial Boards of Information & Management, Internet Research, and Management Science and Regional Development. He has published four books and some 50 papers on various aspects of information systems and their management. He has held visiting appointments at New York University, Claremont Graduate University, the Naval Postgraduate School, and the Czechoslovak Management Center (now CMC Graduate School of Business).
Opher Etzion is a research staff member and chair of the research promotion activities at the IBM Research Laboratory in Haifa; he also serves a senior teaching associate at the Technion. Previously he had been a faculty member at the Technion, and the founding head of the information systems engineering area and graduate program. His research interests includes active technology, temporal databases, middleware systems and rule-base systems. He is a member of the editorial board of the IIE Transactions Journal, was a guest editor in the Journal of Intelligent Information Systems in 1994, served as a coordinating organizer of the Dagstuhl seminar on Temporal databases in 1997, and the program chair of CoopIS2000. He is the coeditor of the book "Temporal Databases – Research and Practice" published by Springer-Verlag. He also served in many conference program committees as well as national committees and has been program and general chair in the NGITS workshop series.