8-16-1996

# A Meta Knowledge Base and A Search Mechanism for Distributed, Heterogeneous Databases

Peretz Shoval
*Department of Industrial Engineering & Management, Ben-Gurion University*

Philip Ein-Dor
*Faculty of Management, Tel-Aviv University*

Ran Giladi
*Department of Industrial Engineering & Management, Ben-Gurion University*

Israel Spiegler
*Falculty of Management, Tel-Aviv University*

Israel Spiegler
*Falculty of Management, Tel-Aviv University*

***See next page for additional authors***

# A Meta Knowledge Base and A Search Mechanism for Distributed, Heterogeneous Databases

Peretz Shoval*, Philip Ein-Dor**, Ran Giladi*, and Israel Spiegler**
* Dept. of Industrial Engineering & Management, Ben-Gurion University, Beer-Sheva 84105 Israel {shoval/ran}@bgumail.bgu.ac.il
** Faculty of Management, Tel-Aviv University, Tel-Aviv, Israel. {eindor/n63}@vm.tau.ac.il.

## Abstract

This paper deals with the issue of accessing relevant information in a network that consists of heterogeneous databases of various types, e.g. structured databases as well as text, audio, and picture files, which reside in locations not necessarily known to the users. The objective is to construct a search mechanism to find the most relevant databases in the network with the help of a meta knowledge base (MKB). This is an ongoing research project, and the current paper provides only a general overview of the problem and the architecture of the proposed solution.

## 1. Introduction - The Problem

Assume a very large and widely distributed organization located in a number of different places (possibly in different countries). The organization maintains many heterogeneous databases of various types, including structured databases, document and other text files, pictures, graphs, voice and video files, etc. The databases are autonomous (i.e. not federated or otherwise integrated), and maintained by local organizations. Users, e.g., managers and knowledge workers, may want to find information of all types that resides in different locations and databases throughout the organization. They do not necessarily know what information is available or where it is located.

One possible solution to this problem is to create a "tight" federated system, with a global, integrated database schema and a unified user interface, as proposed in [1]. This solution is not practical, however, as it requires the resolution of many conflicts that may exist in the different databases, let alone the fact that we are dealing with different kinds of data, not just record structured databases.

In [2], a multiple database approach is presented, according to which users are provided with a multiple database language that permits manipulation of a collection of autonomous databases. But, as the authors admit, it is unreasonable to assume that such languages are universally available. At any rate, even with such languages, it is still up to the users to specify which databases to access; they need to know what information is in each database, and to send their requests to those nodes which they believe are relevant. Since users cannot be sure about the relevant nodes, they are likely either to miss some of the relevant data (in the case of a restriction to "safe" locations only) or to overload the network, incur extra expenses, and perhaps get irrelevant information (in the case of a less restrictive approach).

Research in semantic reconciliation includes techniques that attempt to provide an overall framework to support interoperability. One such framework is provided by Weishel and Kerschberg [3], who propose artificial intelligence techniques to construct domain models, i.e., data and knowledge representations of the constituent databases, and an overall domain model of the semantic interactions among the databases. A "global thesaurus" addresses semantic heterogeneity issues, playing the role of a data dictionary. The architecture of the solution we propose is in this vein.

## 2. The Proposed Solution

We propose a solution which is based on an intelligent search mechanism aided by a meta knowledge-base (MKB) that includes semantic and administrative information on the contents of the various nodes and databases. The general architecture of the meta knowledge base and the search mechanism are described in [4]. The users of the databases pose queries in natural language. A query is translated into a federated query language QNF [5]. QNF is a model for transforming queries in any language, or in natural language, into a standard format for accessing relational databases. The translated query consists of key words taken from the original user query, plus synonymous terms generated by the QNF thesaurus. The system meshes this information with information in the MKB to identify relevant database. The decision on which of the relevant database nodes to access is based both on semantic information and managerial considerations (e.g., cost, efficiency, security, etc.). Once decided, the system establishes efficient routes to the nodes in question and the user's specific request is submitted to the databases.

# 3. Data Sets and Meta Knowledge

The network consists of three types of linked nodes:

a. **data nodes** that may include various types of data, as exemplified before. We term all these types "data sets". A data node maintains "local" knowledge about all data sets in that location; we term these "Local Knowledge Bases (LKB).

b. **knowledge nodes** that maintain meta knowledge about databases in associated database nodes. They facilitate identification of the most relevant databases for a given query. We term these Regional Knowledge Bases (RKB).

c. **user nodes** that serve users; they accept users' requests for information in natural language and submit them to neighboring knowledge nodes. In return, each user gets a list of the most relevant databases and their locations.

We next elaborate on the contents of data and knowledge nodes.

### 3.1 Data nodes

A data node may contain one or more data sets of the following types:

- A structured database consists of data files that may be implemented with any DBMS and data model. The structure of the database is defined by the database schema, using the DBMS DDL. In addition, we assume the existence of descriptions of the scope of the database (what it is about, etc.).
- A text file consists of documents containing text and various content descriptors, such as title, authors, abstract, journal/book, a list of key-words, etc.
- Picture, sound, and movie files contain appropriately coded representations of the relevant media.
- A software file contains a program.

Note that picture, sound, movie, and software files contain little self-description as in text files, nor are there schema which describe their contents.

Each data set is represented both in its LKB and in an associated RKB. The main difference between "local" and "regional" knowledge is in level of detail: The LKB has more details about each local node data set; the RKB has more general knowledge about all data sets associated with that region. Whenever a new data set is added to a data node, it is registered in its LKB and RKB. The registration process creates the necessary representation of the data set in each of the knowledge bases, as described herein.

### 3.2 Knowledge nodes

Each user node and each data node is associated with one RKB. That is, every data set in every data node is registered and represented in its RKB; similarly every user node which processes a user query is serviced by its RKB. The knowledge in an RKB is, as said, more general or essential than that in the LKB. It is represented in the form of a lookup-table, or index. Every data set is represented in its RKB by a set of records. The attributes are:

- Data set ID: this includes a unique ID plus location (node address).
- Data type: indicated the type of data set (e.g., structured database, documents, pictures).
- Key words: one of the following:
    1. for a structured database: a list of the file/ relation/class names.
    2. for a documents file: a list of key words that represent the contents of the document.
    3. for "media" files (e.g. pictures): a list of key words based on the file name and other descriptive attributes relating to content.

Until some standard is established for indexing multimedia data, the content information would have to be provided by the local data administration. Such information might also be accumulated as the result of learning from previous successful searches.

In addition to knowledge about its "regional" databases, the RKB may include similar information (records) on "foreign" data sets. This knowledge may also be added after successful searches in other RKBs.

# 4. The Search Mechanism

### 4.1 Search engines

The system comprises several search and evaluation mechanisms, that are located in the various nodes, as follows:

1. In *knowledge nodes* there is a **regional matching & routing mechanism (RMR)** that finds the most relevant databases in the region, based on similarity (correlation) of the query terms and the index keywords.
2. In *database nodes* there is a **local matching & scoring mechanism (LMS)** that finds the most relevant databases in that node, also based on similarity of the query terms and the LKB terms.
3. In each *user node* there is an **evaluation mechanism (EV)** that evaluates the list of data sets and locations proposed by the system after the search (in either type of higher level node). This is the basis for the user to make a decision if, and how many of the relevant databases to access for actual retrieval of data.

### 4.2 Search process

The search process is as follows:

1. First, the user submits a request for information, expressed in natural language. (We do not assume the existence of a federated query language). The request is routed to the neighboring knowledge node, triggering the QNF translator, which generates a QNF query that consists of keywords based on the original user query plus synonymous terms.
2. In the next step, the MRM (i.e., the regional matching & routing mechanism) conducts a search in the RKB, matching query terms with index terms. The result is a rank ordered list of relevant data set nodes.
3. From this point there are three possibilities which need to be investigated:
    1. to return with the list to the user, who will decide if and to how many of the suggested data sets and nodes to access, in order to search in their LKBs to obtain a better ranking.

2. to go "automatically" to those nodes (or the "x" top data sets) and do the same search, or
3. to expand the search to other RKBs, on the basis of additional knowledge found in the current RKB. "Additional knowledge" means meta knowledge on data sets that belong to other knowledge nodes. It includes only successful results ("hits") of searches conducted by users who are attached to the current knowledge node.

The search in data nodes is performed by the LMS (local matching & scoring mechanism). The result now is a weighted list of relevant data sets, together with administrative information (e.g., access rights, cost, efficiency, quality of service, etc.). This list is routed back from each of the relevant data nodes to the user node, where the EV (evaluation mechanism) determines the final list of data sets and nodes where the user should send his or her specific request.

## References


[1] Sheth, A. and Larson, J., "Federated database systems for managing distributed, heterogeneous, and autonomous databases", ACM Computing Surveys, Vol. 22 (3), September 1990, pp. 183-236.

[2] Litwin, W., Mark, L. and Roussopoulos, N., "Interoperability of multiple autonomous databases", ACM Computing Surveys, Vol. 22 (3), September 1990, pp. 267-293.

[3] Weishal, D. and Kerschberg, L., "Data/knowledge packets as a means of supporting semantic heterogeneity in multidatabase systems", ACM SIGMOD Record, Vol. 20 (4), December 1991, pp. 69-73.

[4] Giladi, R. and Shoval, P., "An architecture of an intelligent system for routing user requests in a network of heterogeneous databases", Journal of Intelligent Information Systems, Vol. 3, 1994, pp. 205-221.

[5] Ein-Dor, P. and Spiegler, I, "Natural language access to multiple databases: a model and a prototype", Journal of Management Information Systems, Vol. 12, 1995, pp. 171-197.