An Architecture of an Intelligent System for Routing User Requests in a Network of Heterogeneous Databases

RAN GILADI RAN@BGMAIL.BGU.AC.IL
PERETZ SHOVAL SHOVAL@BENGUS.BGU.AC.IL
Information Systems Program, Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

Abstract. We present a general purpose model for routing user requests, e.g. queries, in a network of autonomous heterogeneous databases. The database schemas and other information on the database nodes are used to construct a multi-level knowledge-base (MKB) that resides in various nodes. Access to the databases is not done by creating direct connections between the user and the nodes where the data are presumably located. Rather, the user approaches the network by contents via an intelligent system that utilizes the MKB in order to identify the nodes and databases where the most relevant information resides, and establishes access routes to those nodes.

Keywords: computer communication networks, database management, distributed systems, information storage and retrieval, heterogeneous databases, federated databases, semantic networks, knowledge bases, database schema and subschema, query processing

1. Introduction

We assume a multi user and multi database networked environment. Certain nodes in the network may include one or more, possibly heterogeneous databases that do not necessarily utilize the same data model and DBMS. The entire database system is autonomous or "loosely federated", i.e., the databases are nonintegrated, and there is no global or federated schema (Sheth and Larson, 1990). Hence, users must interface with multiple autonomous databases. In large enterprises it is not uncommon to find autonomous database systems which are impossible or impractical to integrate. One common example of such an environment is information retrieval databases: in the presence of a large number of retrieval databases, the users may have trouble choosing which database to use, let alone formulating precise queries (Litwin, Mark and Roussopoulos, 1990). Another example is when different organizations merge into a single one, but maintain the existing information systems and databases.

In this environment, users need information that may reside in different locations and databases, but do not necessarily know where to find it. In current technology the user must know where the relevant database is located (the physical address) in order to trigger a regular communication process to that node. This is usually done by issuing a "call establish" phase for determining the route between the source and the target nodes, followed by a "data transfer" phase which carries out the dialogue between the user and the database. The "call establish" phase in communication networks offers means for
establishing a connection between one node and another which is being searched for by name (the question being: who are we seeking). We extend this mechanism, so that instead of seeking a node by name, we seek it by contents (so that the question: what are we seeking will be answered). Such a mechanism, which is a sort of pseudo-associative routing, permits users to "skip" the "call establish" phase and establishes connections to databases without knowing their locations. This mechanism resembles adaptive routing, where the objective is to build adaptive routing tables to optimize the response to both individual queries and global utilization. In the proposed mechanism, adaptive routing is carried out by optimizing the places where information resides (by building appropriate routing tables). Before we elaborate on our solution, we present current strategies that have attempted to overcome this problem.

One strategy, described by (Sheth and Larson, 1990), is to create a "tight" federated system, with a global database schema and a unified user interface, including mappings to the various databases and their query languages. This solution is costly and impractical, as it requires the resolution of all kinds of conflicts in the different databases. In addition to non homogeneity, which causes integration problems, we also have to know which nodes to access.

Another strategy is the multidatabase approach, presented by (Litwin, Mark and Rousopoulos, 1990). According to this approach, users are provided with a multidatabase language that manipulates a collection of autonomous databases. The authors admit, however, that it is unreasonable to assume that such languages are universally available. At any rate, even with such a language, it is still up to the users to specify which databases to access. Users have to be aware of the information in each database, and to send their requests to the relevant nodes. Since they cannot be sure which nodes are relevant, they either miss some of the data (in the case of a restriction to “safe” locations only) or they overload the network, incur extra expenses, and perhaps get irrelevant information (in the case of a less restrictive approach).

(Ouksel and Naiman, 1992) discuss the problem of semantic reconciliation in heterogeneous databases, and advocate the design of semantic communication protocols to handle meaningful exchange of information. Such protocols are communication independent (which is analogous to database architecture which are physically and logically independent), transparent (not requiring human intervention), and have an incremental discovery process (using available semantic knowledge in order to determine the relevant information and refine the context for further dialogue). They argue that semantic discovery is an iterative process that can be supported by some general semantic tools, such as a thesaurus or a domain specific knowledge base.

Research in semantic reconciliation includes techniques that attempt to provide an architecture, or an overall framework to support interoperability as well as the entire reconciliation process. One such framework is provided by (Weishel and Kerschberg, 1991), 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. The domain models are represented as Knowledge Sources (KS) in a “blackboard architecture”. The KSs work together in order to provide users with simultaneous, multiple viewpoints of the system at varying levels