Handling Semantic Inconsistencies in Distributed Knowledge Systems Using Ontologies

Zbigniew W. Raś[1,2] and Agnieszka Dardzińska[3]

[1] University of North Carolina, Department of Computer Science
Charlotte, N.C. 28223, USA
ras@uncc.edu

[2] Polish Academy of Sciences, Institute of Computer Science
Ordona 21, 01-237 Warsaw, Poland

[3] Bialystok University of Technology, Department of Mathematics
15-351 Bialystok, Poland
agnadar@wp.pl

Abstract. Traditional query processing provides exact answers to queries. It usually requires that users fully understand the database structure and content to issue a query. Due to the complexity of the database applications, the so called global queries can be posed which traditional query answering systems can not handle. In this paper a query answering system based on distributed data mining is presented to rectify these problems. Task ontologies are used as a tool to handle semantic inconsistencies between sites.

1 Introduction

In many fields, such as medical, banking and educational, similar databases are kept at many sites. An attribute may be missing in one database, while it occurs in many others. Missing attributes lead to problems. A user may issue a query to a local database S_1 in search for objects in S_1 that match a desired description, only to realize that one component a_1 of that description is missing in S_1 so that the query cannot be answered. The definition of a_1 may be extracted from databases at other sites and used to identify objects in S_1 having property a_1. The simplicity of this approach is no longer in place when the semantics of terms used to describe objects in a client and remote sites differ. Sometime, such a difference in semantics can be repaired quite easily. For instance if "Temperature in Celsius" is used at one site and "Temperature in Fahrenheit" at the other, a simple mapping will fix the problem. If databases are complete and two attributes have the same name and differ only in their granularity level, a new hierarchical attribute can be formed to fix the problem. If databases are incomplete, the problem is more complex because of the number of options available to interpret incomplete values (including null vales). The problem is especially difficult when rule-based chase techniques are used to replace null values by values which are less incomplete.

M.-S. Hacid et al. (Eds.): ISMIS 2002, LNAI 2366, pp. 66–74, 2002.

The notion of an intermediate model, proposed by [Maluf and Wiederhold] [1], is very useful to deal with heterogeneity problem, because it describes the database content at a relatively high abstract level, sufficient to guarantee homogeneous representation of all databases. Knowledge bases built jointly with task ontologies proposed in this paper, can be used for a similar purpose. Knowledge bases contain rules extracted from databases at remote sites.

In this paper, the heterogeneity problem is introduced from the query answering point of view. Query answering system linked with a client site transforms, so called, global queries using definitions extracted at remote sites. These definitions may have so many different interpretations as the number of remote sites used to extract them. Task ontologies are used to find new interpretations representing consensus of all these sites.

2 Distributed Information Systems

In this section, we recall the notion of a distributed information system and a knowledge base for a client site formed from rules extracted at remote sites. We introduce the notion of local queries and give their standard semantics.

By an *information system* we mean $S = (X, A, V)$, where X is a finite set of objects, A is a finite set of attributes, and $V = \bigcup\{V_a : a \in A\}$ is a set of their values. We assume that:

- V_a, V_b are disjoint for any $a, b \in A$ such that $a \neq b$,
- $a : X \longrightarrow 2^{V_a} - \{\emptyset\}$ is a function for every $a \in A$.

Instead of a, we may write $a_{[S]}$ to denote that a is an attribute in S.

By *distributed information system* we mean $DS = (\{S_i\}_{i \in I}, L)$ where:

- I is a set of sites.
- $S_i = (X_i, A_i, V_i)$ is an information system for any $i \in I$,
- L is a symmetric, binary relation on the set I.

A distributed information system $DS = (\{S_i\}_{i \in I}, L)$ is consistent if the following condition holds:

$$(\forall i)(\forall j)(\forall x \in X_i \cap X_j)(\forall a \in A_i \cap A_j)$$
$$[(a_{[S_i]}(x) \subseteq a_{[S_j]}(x)) \text{ or } (a_{[S_j]}(x) \subseteq a_{[S_i]}(x))].$$

Let $S_j = (X_j, A_j, V_j)$ for any $j \in I$. In the remainder of this paper we assume that $V_j = \bigcup\{V_{ja} : a \in A_j\}$.

From now on, in this section, we use A to denote the set of all attributes in DS, $A = \bigcup\{A_j : j \in I\}$. Also, by V we mean $\bigcup\{V_j : j \in I\}$.

Before introducing the notion of a knowledge base, we begin with a definition of $s(i)$-terms and their standard interpretation M_i in $DS = (\{S_j\}_{j \in I}, L)$, where $S_j = (X_j, A_j, V_j)$ and $V_j = \bigcup\{V_{ja} : a \in A_j\}$, for any $j \in I$.

By a set of $s(i)$-terms (also called a set of local queries for site i) we mean a least set T_i such that: