Ontology Based Distributed Autonomous Knowledge Systems

Zbigniew W. Raś[*,°], Agnieszka Dardzińska[+]

[*] *University of North Carolina, Department of Computer Science, Charlotte, N.C. 28223, USA*
[°] *Polish Academy of Sciences, Institute of Computer Science, Ordona 21, 01-237 Warsaw, Poland*
[+] *Bialystok University of Technology, Department of Mathematics, Wiejska 45 A, 15-351 Bialystok, Poland*

Abstract

Traditional query processing usually requires that users fully understand the database structure and content to issue a query. Due to the complexity of the database applications and the variety of user needs, the so called global queries are introduced which traditional query answering systems can not handle. Query posed to a database D is global if minimum one of its attributes is missing in D while it occurs in other databases. Definitions of a missing attribute in D can be extracted from other databases and shared with D. To handle semantics inconsistencies between the same attributes used at different sites, task ontologies are used as a communication bridge between them. These inconsistencies can be caused either by different granularity levels or by different interpretations of the same attribute. As the final outcome of this research, a rough query answering system based on distributed data mining is presented.

Key words: distributed autonomous knowledge systems, query processing, knowledge mining, ontologies, semantics inconsistencies

Email addresses: ras@uncc.edu (Zbigniew W. Raś[*,°],), adardzin@uncc.edu (Agnieszka Dardzińska[+]).

URLs: http://www.cs.uncc.edu/ ras (Zbigniew W. Raś[*,°],), http://www.coe.uncc.edu/ adardzin (Agnieszka Dardzińska[+]).

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. Databases are often incomplete which means that queries can be either answered approximatively (for instance in a rough sets framework) or to retrieved objects some weights are assigned. Each query can be answered in many different ways, depending on the interpretation of incomplete values. In the most general scenario, an answer to a query submitted to an information system (see [Ras and Dardzinska][13], [Pawlak][8]) is a set of objects certainly satisfying the query, and a set of objects possibly satisfying the query. Different interpretations may assign different values to possible objects showing the confidence of the query answering system in each object returned as an answer to the query. These values can be calculated either using information on frequency of appearance of some attribute values in the information system or using local and global KDD methods which can predict what attribute values should replace the missing values of incomplete attributes and what weights should be assigned to them.

Missing or incomplete attributes lead to problems when answering queries. For example, a user may issue a query to a local database S in search for its objects that match a desired description, only to realize that attribute a_1 used in that description is either missing or vast majority of its values in S is incomplete so that the query cannot be answered. But definitions of a_1 may be extracted from databases at remote sites and used to identify objects in S having properties related to a_1. The simplicity of this approach is no longer in place when the semantics of terms used to describe objects at 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 in a distributed framework when rule-based chase techniques, driven by rules extracted at remote sites, are used by a client site to replace null values by values which are less incomplete.

The notion of an intermediate model, proposed by [Maluf and Wiederhold][3], 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. Knowledgebases built jointly with task ontologies proposed in our paper, can be used for a similar purpose. They

contain rules extracted from databases at remote sites. These rules are seen as definitions of their decision values. Sometime these definitions are inconsistent so some form of a consensus has to be reached. Algorithm addressing this issue was proposed by [Ras][10].

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 into local queries for a client site using task ontologies and definitions extracted at remote sites. These definitions may have so many different interpretations as the number of remote sites used to extract them. This paper will mainly focus on different interpretations of queries due to different interpretations of null values and also due to granularity levels of a given attribute which may easily vary from site to site. Information stored in task ontologies is used to find so called rough interpretations representing consensus of all involved sites.

2 Distributed information systems

In this section, we recall the notion of a distributed information system and a knowledgebase for a client site formed from rules extracted at remote sites. We introduce the notion of local queries and give example of their local 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. The set V_a is called the domain of attribute a. 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, to simplify the notation, we may use $a_{[S]}$ to denote that a is an attribute in S. Saying that object x has a property $a(x) = \{w_1, w_2, ..., w_k\}$, we mean that only one of the values in $a(x)$ is a property of x but we do not know which one. Also, the empty set is removed from the codomain of a since the null value is represented here by V_a.

By a *distributed information system* we mean a pair $DS = (\{S_i\}_{i \in I}, L)$ 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 showing which systems can directly communicate with each other.

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

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

Consistency basically means that information about objects in one of the systems is more general than in the other. Saying another words, two consistent systems can not have conflicting information about any object x which is stored in both of them.

Our next step is to introduce the notion of a knowledgebase containing definitions of attribute values in terms of other attribute values. These definitions have a form of association rules with only one decision value. Each information system S_i, $i \in I$, is coupled with a knowledgebase containing definitions of some foreign and some local attribute values for S_i. These definitions are used by a query answering system to replace global queries by local queries so the range of queries the system can successfully handle will significantly increase.

Let $S_j = (X_j, A_j, V_j)$ be an information system at site 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 $\bigcup\{V_j : j \in I\}$.

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:

- $\mathbf{0}, \mathbf{1} \in T_i$,
- $w \in T_i$ and $\sim w \in T_i$ for any $w \in V_i$,
- if $t_1, t_2 \in T_i$ then $(t_1 + t_2), (t_1 * t_2) \in T_i$.

The negation is used only at the level of atomic terms in the definition of $s(i)$-terms which is due to the incompleteness of information systems. If we allow to build queries (terms) without this restriction, then the negation would have to be moved to the level of atomic terms, by the query answering systems, before queries are processed. In order to do that, De Morgan's laws have to be used but they do not hold for incomplete information systems for most of the semantics including the optimistic semantics which is used in this paper as standard one. Also, it has to be noticed that attributes in the definition of S_i, $i \in I$, are not hierarchical. The definition of formulas, given below, is reduced to atomic formulas since, in this paper, only such formulas need to be investigated.

By a set of $s(i)$-formulas we mean a least set F_i such that:

- if $t_1, t_2 \in T_i$, then $(t_1 = t_2) \in F_i$ and $(t_1 \leq t_2) \in F_i$.

Definition of DS-terms (called global queries) and DS-formulas is quite similar to the definition of $s(i)$-terms and $s(i)$-formulas (we only replace T_i by $\bigcup\{T_i : i \in I\}$ and F_i by F).

We say that:

- $s(i)$-term t is *primitive* if it is of the form $\prod\{t : ((t = w) \vee (t =\sim w)) \wedge (w \in U_i)\}$ for any $U_i \subseteq V_i$,
- $s(i)$-term is in *disjunctive normal form* (DNF) if $t = \sum\{t_j : j \in J\}$ where each t_j is primitive.

Similar definitions can be given for DS-terms.

There is no need to give an example of a local query. The expression:

select $*$ from $Flights$
where $airline =$ "LOT"
and $departure\_time =$ "$morning$"
and $departure\_airport =$ "$Warsaw$"
and $aircraft =$ "$Boeing$"

is an example of a non-local query (because of attribute $aircraft$) in a database

$Flights(airline, departure\_time, arrival\_time,$
$departure\_airport, arrival\_airport)$.

Semantics of $s(i)$-terms and $s(i)$-formulas (local queries for the site i) can be defined as the interpretation M_i, called standard, in a distributed information system $DS = (\{S_j\}_{j \in I}, L)$ as follows:

- $M_i(\mathbf{0}) = \emptyset$, $M_i(\mathbf{1}) = X_i$
- $M_i(w) = \{x \in X_i : w \in a(x)\}$ for any $w \in V_{ia}$,
- $M_i(\sim w) = \{x \in X_i : w \notin a(x)\}$ for any $w \in V_{ia}$,
- if t_1, t_2 are s(i)-terms, then
  $M_i(t_1 + t_2) = M_i(t_1) \cup M_i(t_2)$,
  $M_i(t_1 * t_2) = M_i(t_1) \cap M_i(t_2)$,
  $M_i(t_1 = t_2) =$(if $M_i(t_1) = M_i(t_2)$ then T else F),
  $M_i(t_1 \leq t_2) =$(if $M_i(t_1) \subseteq M_i(t_2)$ then T else F)
  where T stands for $True$ and F for $False$

Clearly M_i is an optimistic interpretation of $s(i)$-terms which means that predicates $\{=, \leq\}$ should be read as *possibly equal* and *possibly included*, correspondingly. If the set of $s(i)$-formulas is reduced only to equality formulas,

X_i	a	b	c	d	e	f
x_1	$a1$		$c2$	$d1$	$e1$	$f1$
x_2		$b1$	$c1$	$d1$	$e1$	$f1$
x_3		$b1$	$c2$	$d2$	$e1$	$f1$
x_4	$a2$	$b1$				$f2$
x_5	$a2$	$b2$		$d2$		$f2$
x_6	$a1$		$c1$	$d1$	$e1$	$f2$
x_7	$a2$	$b2$	$c2$	$d2$		$f2$
x_8	$a2$	$b2$	$c2$		$e1$	$f1$

Table 1
Information System S_i

the sound and complete axiomatization, of the above semantics, for a complete distributed information system was given in paper by [Ras][11]. This semantics was extended to handle global queries in complete distributed information systems (see [Ras][11]). For incomplete distributed information systems, a different interpretation of $s(i)$-terms and $s(i)$-formulas was proposed by [Ras and Joshi][14]. Its sound and complete axiomatization was also given by [Ras and Joshi][14].

We can propose a number of different interpretations for $s(i)$-terms and the same for $s(i)$-formulas when information systems are incomplete. For example, let us assume that information system S_i is represented by Table 1.

To propose an interpretation of $s(i)$-terms, we have to decide first how to interpret the attribute values in S_i. For instance, let us take the value $a1 \in V_a$. Three possible interpretations J_1, J_2, J_3 of $a1$ are given below:

- $J_1(a1) = \{(x_1, 1), (x_6, 1), (x_2, 1/3), (x_3, 1/3)\}$,
- $J_2(a1) = \{(x_1, 1), (x_6, 1)\}$,
- $J_3(a1) = \{(x_1, 1), (x_6, 1), (x_2, 1/2), (x_3, 1/2)\}$,

Confidence $1/3$ assigned to objects x_2, x_3 in the interpretation J_1 is justified by the fact that $1/3$ of the objects in S_i, for which the value of the attribute a is known, have property $a1$.

Interpretation J_2 is a pessimistic one so it assumes that the missing values, in the column of S_i corresponding to attribute a, are not equal to $a1$.

Interpretation J_3 checks the values of the objects from $J_2(a1)$ in the column of S_i corresponding to attribute f. Since $1/2$ of the objects in $J_2(a1)$ have the property $f1$ which is a property of both objects x_2, x_3, then the confidence $1/2$ is assigned to x_2, x_3.

After the interpretation of values of attributes in S_i is given, we have to decide what is the meaning of functors $+$, $*$, $\sim$ in S_i. Again, we can propose a number of interpretations for $+$, $*$, $\sim$. For instance, the interpretation J_1, defined already for atomic terms, can be extended as (we give only an example):

- $J_1(a1 * b1) = J_1(a1) \otimes J_1(b1) = \{(x_1, 1), (x_2, 1/3), (x_3, 1/3), (x_6, 1)\} \otimes \{(x_1, 1/2), (x_2, 1), (x_3, 1), (x_4, 1), (x_6, 1/2)\} = \{(x_1, 1 \cdot 1/2), (x_2, 1/3 \cdot 1), (x_3, 1/3 \cdot 1), (x_6, 1 \cdot 1/2)\} = \{(x_1, 1/2), (x_2, 1/3), (x_3, 1/3), (x_6, 1/2)\}$
- $J_1(a1 + b1) = J_1(a1) \oplus J_1(b1) = \{(x_1, 1), (x_2, 1/3), (x_3, 1/3), (x_6, 1)\} \oplus \{(x_1, 1/2), (x_2, 1), (x_3, 1), (x_4, 1), (x_6, 1/2)\} = \{(x_1, max(1, 1/2)), (x_2, max(1/3, 1)), (x_3, max(1/3, 1)), (x_6, max(1, 1/2))\} = \{(x_1, 1), (x_2, 1), (x_3, 1), (x_6, 1)\}$
- $J_1(\sim a1) = \{(x_2, 2/3), (x_3, 2/3), (x_4, 1), (x_5, 1), (x_7, 1), (x_8, 1)\}$

Clearly, this example shows that $s(i)$-terms may have many different interpretations at site $i \in I$. In a distributed scenario, if we want to talk about knowledge exchange between sites, we have to build a bridge between them so information about what semantics they use and what is the relationship between these semantics can be stored. In this paper, we suggest that task ontologies can be used to build such a bridge between sites of a distributed information system.

Now, we introduce the notion of (k, i)-rules, for any $i \in I$. The proposed definition is general enough to cover rules extracted from an information system at one of the sites of DS and next used them by its other sites either to build their knowledgebases or just use as definitions to handle global queries submitted to them.

By (k, i)-rule in $DS = (\{S_j\}_{j \in I}, L)$, $k, i \in I$, we mean a triple (c, t, s) such that:

- $c \in V_k - V_i$,
- t, s are $s(k)$-terms in DNF and they both belong to $T_k \cap T_i$,
- $M_k(t) \subseteq M_k(c) \subseteq M_k(t + s)$.

Any (k, i)-rule (c, t, s) in DS can be seen as a definition of c which is extracted from S_k but can be used in S_i because all attribute values occurring in the definition of c are local for S_i.

For any (k, i)-rule (c, t, s) in $DS = (\{S_j\}_{j \in I}, L)$, we say that:

- $(t \rightarrow c)$ is a weakly k-certain rule in DS,
- $(t + s \rightarrow c)$ is a weakly k-possible rule in DS.

The term *weakly* is used here because of the interpretation M_i which assigns

the set of objects which might have property t to any term t. Clearly, if the system S_i is becoming less incomplete, then the term *weakly* is less suitable here. For a complete system, the rule $(t \rightarrow c)$ is a k-certain rule.

Now, we are ready to define a knowledgebase D_{ki}. Its elements are called definitions of values of attributes from $V_k - V_i$ in terms of values of attributes from $V_k \cap V_i$.

Namely, D_{ki} is defined as a set of (k,i)-rules such that:
if $(c,t,s) \in D_{ki}$, then $(\exists t_1)[(\sim c, t1, s) \in D_{ki}]$.

The idea here is to have definition of $\sim c$ in D_{ki}, if already definition of c is in D_{ki}. It will allow us to approximate (learn) concept c from both sites, if needed.

By a knowledgebase for site i, denoted by D_i, we mean any subset of $\bigcup\{D_{ki} : (k,i) \in L\}$. If definitions are not extracted at a remote site, partial definitions (c,t) corresponding to $(t \rightarrow c)$, if available, can be stored in a knowledgebase at a client site.

For simplicity reason, we did not talk here about possible differences in semantics between sites $k, i \in I$. If these semantics differ, then rules in the knowledgebase D_{ki} have to be modified using semantical graph, in a task ontology for DS, which shows all relationships between semantics used in DS.

3 Semantic inconsistencies and distributed autonomous knowledge systems

In this section, we introduce the notion of a distributed autonomous knowledge system ($DAKS$) and next present problems related to the construction of its query answering system QAS. We discuss the process of handling semantics inconsistencies in knowledge extracted at different sites of $DAKS$. Next, we outline the transformation steps for global queries so some of their atomic terms can be replaced by subterms extracted at different sites of $DAKS$ by knowledge discovery techniques. The goal of the query transformation process is to replace a global query by a local one for a site specified by a user. This way, the global query can be processed locally. Also, we show how global queries can be processed if two sites involved in a query resolution use different granularity levels for the same attribute.

By Distributed Autonomous Knowledge System ($DAKS$) we mean $DS = (\{(S_i, D_i)\}_{i \in I}, L)$ where $(\{S_i\}_{i \in I}, L)$ is a distributed information system, $D_i = \bigcup\{D_{ki} : (k,i) \in L\}$ is a knowledgebase for $i \in I$.

Figure 1 shows an example of $DAKS$ and its query answering system QAS that handles global queries.

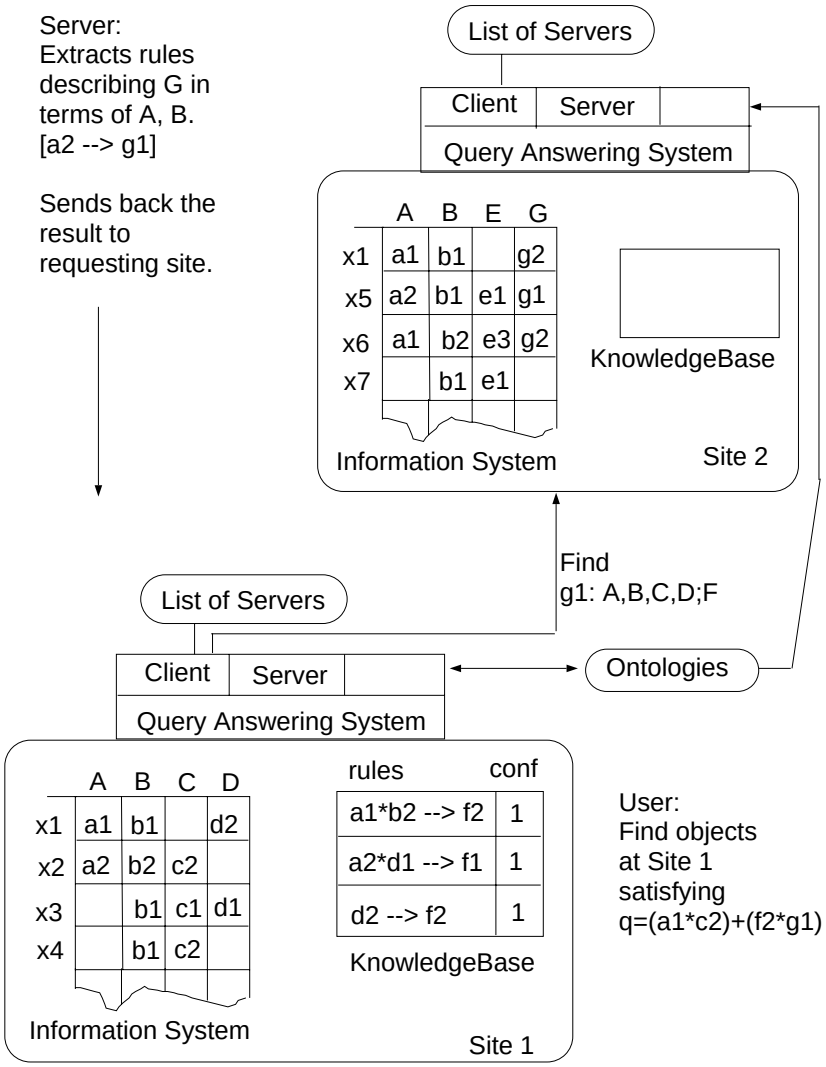

Fig. 1. Sample Model of DAKS

For simplicity reason only two sites of $DAKS$ are considered in our example. A user queries site 1 asking for all its objects satisfying $q = (a_1 * c_2) + (f_2 * g_1)$. The user is interested only in objects at site 1. The first part of the query, which is $(a_1 * c_2)$, can be handled by local QAS because both its attribute values are within the domain of the information system at site 1. For instance, taking optimistic interpretation of all attribute values at site 1 (null values are treated as supporting values), objects x_1 and x_2 will satisfy query $(a_1 * c_2)$. Let us consider the second part of the query q, which is $(f_2 * g_1)$. Attribute value f_2 is not in the domain V_1 of the information system S_1 at site 1 but its definition is in knowledgebase D_1 at site 1. This definition can be used to replace the value f_2 in q by a term which defines f_2. In our example, either a partial definition $(f_2, a_1 * b_2 + d_2)$ or a definition $(f_2, a_1 * b_2 + d_2, \sim (a_1 * b_2 + d_2) * \sim d_2)$ of f_2 can be generated from D_1. The attribute value g_1 used in a query q

is neither in the domain D_1 nor its definition is in D_1. In this case we have to search for a remote site, where definition of g_1 can be extracted from its information system. In our example site 2 satisfies this requirement. Expression (g_1, a_2) can be seen as a partial definition of g_1 which can be extracted from S_2. Alternatively, expression $(g_1, a_2, \sim a_1* \sim a_2)$ can be used as a definition of g_1 found at site 2. To simplify the problem further, assume that partial definitions of f_2 and g_1 are used to replace query q by a new query which can be handled locally by QAS at site 1. This new query approximating q, described by a term $(a_1 + c_2) + (a_1 * b_2 + d_2) * a_2$, can be seen as a lower approximation of q in rough sets terminology. If we use definition of f_2 and definition of g_1 instead of partial definitions, query q can be replaced by a rough query. Rough queries are especially useful when the boundary area in a rough query representation is small. In a distributed scenario, similar to the one presented in this paper, this boundary area is getting smaller and smaller when more and more sites are used to search for definitions of non-local attributes (f_2 and g_1 in our example). Now, let's go back to term $(a_1 * c_2) + (a_1 * b_2 + d_2) * a_2$. If the distribution law can be applied then our term would be transformed to $(a_1 * c_2) + (a_1 * b_2 * a_2 + d_2 * a_2)$ and next assuming that properties $a_1 * a_2 = 0$ and $0 * t = 0$ hold we would get its final equivalent form which is $a_1 * c_2 + d_2 * a_2$. However if each of our three terms $a_1 * b_2$, d_2, a_2 is computed under three different semantics, then there is a problem with the above transformation process and with a final meaning of $(a_1 * b_2 * a_2 + d_2 * a_2)$.

For instance, let us assume the scenario where partial definition $(f_2, a_1 * b_2)$ was extracted under semantics M_1, partial definition (f_2, d_2) under semantics M_2, and (g_1, a_2) under semantics M_3. Null value is interpreted below as a set of all possible values for a given attribute. Also, x has the same meaning as the pair $(x, 1)$.

Semantics M_1 (see [Ras and Joshi][14]) is defined as:

- $M_1(v) = \{(x, k) : v \in a(x)\ \&\ k = card(a(x))\}$ if $v \in V_a$,
- $M_1(t_1 * t_2) = M_1(t_1) \otimes M_1(t_2)$,
- $M_1(t_1 + t_2) = M_1(t_1) \oplus M_1(t_2)$.

To define $\otimes$, $\oplus$, let us assume that $P_i = \{(x, p_{<x,i>}) : p_{<x,i>} \in [0, 1]\ \&\ x \in X\}$ where X is a set of objects. Then, we have:

- $P_i \otimes P_j = \{(x, p_{<x,i>} \cdot p_{<x,j>})\colon x \in X\}$,
- $P_i \oplus P_j = \{(x, max(p_{<x,i>}, p_{<x,j>}))\colon x \in X\}$.

Semantics M_2 is defined as:

- $M_2(v) = \{x : v \in a(x)\ \&\ card(a(x)) = 1\}$ if $v \in V_a$,
- $M_2(t_1 * t_2) = M_2(t_1) \cap M_2(t_2)$,
- $M_2(t_1 + t_2) = M_2(t_1) \cup M_2(t_2)$.

Finally, semantics M_3 is defined as:

- $M_3(v) = \{x : v \in a(x)\}$ if $v \in V_a$,
- $M_3(t_1 * t_2) = M_3(t_1) \cap M_3(t_2)$,
- $M_3(t_1 + t_2) = M_3(t_1) \cup M_3(t_2)$.

Assume now, the following relationship between semantics M_i and M_j:
$M_i \preceq M_j$ iff $[\ (x,k) \in M_i(a) \longrightarrow (\exists k_1 \geq k)[(x,k_1) \in M_j(a)]\ ]$.

It can be easily proved that $\preceq$ is a partial order relation.

In our example, for any $v \in V_a$, we have:
$M_2(v) \preceq M_1(v) \preceq M_3(v)$.

We say that functors $+$ and $*$ preserve monotonicity property for semantics N_1, N_2 if the following two conditions hold:

- $[N_1(t_1) \preceq N_2(t_1)\ \&\ N_1(t_2) \preceq N_2(t_2)]$ implies $N_1(t_1 + t_2) \preceq N_2(t_1 + t_2)$,
- $[N_1(t_1) \preceq N_2(t_1)\ \&\ N_1(t_2) \preceq N_2(t_2)]$ implies $N_1(t_1 * t_2) \preceq N_2(t_1 * t_2)$.

Let $(\Omega, \preceq)$ be a partially ordered set of semantics. We say that it preserves monotonicity property for $+$ and $*$, if $+$ and $*$ preserve monotonicity property for any N_1, N_2 in Ω.

It can be easily checked that $(\{M_1, M_2, M_3\}, \preceq)$ preserves monotonicity property for $+$ and $*$.

We adopt, in this paper, the definition of ontology proposed by Mizoguchi [6]. He claims that ontology should consist of *task* ontology which characterizes the computational architecture of a (distributed) knowledge system which performs a task and *domain* ontology which characterizes the domain knowledge where the task is performed.

To answer a query at a client site, many remote sites may have to be accessed. At each of these sites, the same attributes can be used for knowledge (definitions) extraction. Semantics assigned to them may differ at all or some of these sites. In a query transformation process new terms are formed to replace the current subterms. These new terms are constructed from the above definitions. Since they are extracted not necessarily from the same site of $DAKS$, such a query transformation process is not legal unless a common, possibly optimal, semantics for all these terms is found.

We claim that one way to solve this problem is to assume that partially ordered set of semantics $(\Omega, \preceq)$, preserving monotonicity property for $+$ and $*$, is a part of global ontology (or task ontology in Mizoguchi's [6] definition).

In our example, we evaluate query q taking first M_2 and next M_3 as two com-

mon semantics for all subterms obtained during the transformation process of q because $M_2(v) \preceq M_1(v) \preceq M_3(v)$. In general, if $(\Omega, \preceq)$ is a lattice and $\{M_i\}_{i\in I}$ is a set of semantics involved in transforming query q, then we should take $M_{min} = \bigcap\{M_i\}_{i\in I}$ (the greatest lower bound) and $M_{max} = \bigcup\{M_i\}_{i\in I}$ (the least upper bound) as two common semantics for processing query q.

Common semantics is also needed because of the necessity to prove soundness and possibly completeness of axioms to be used in a query transformation process.

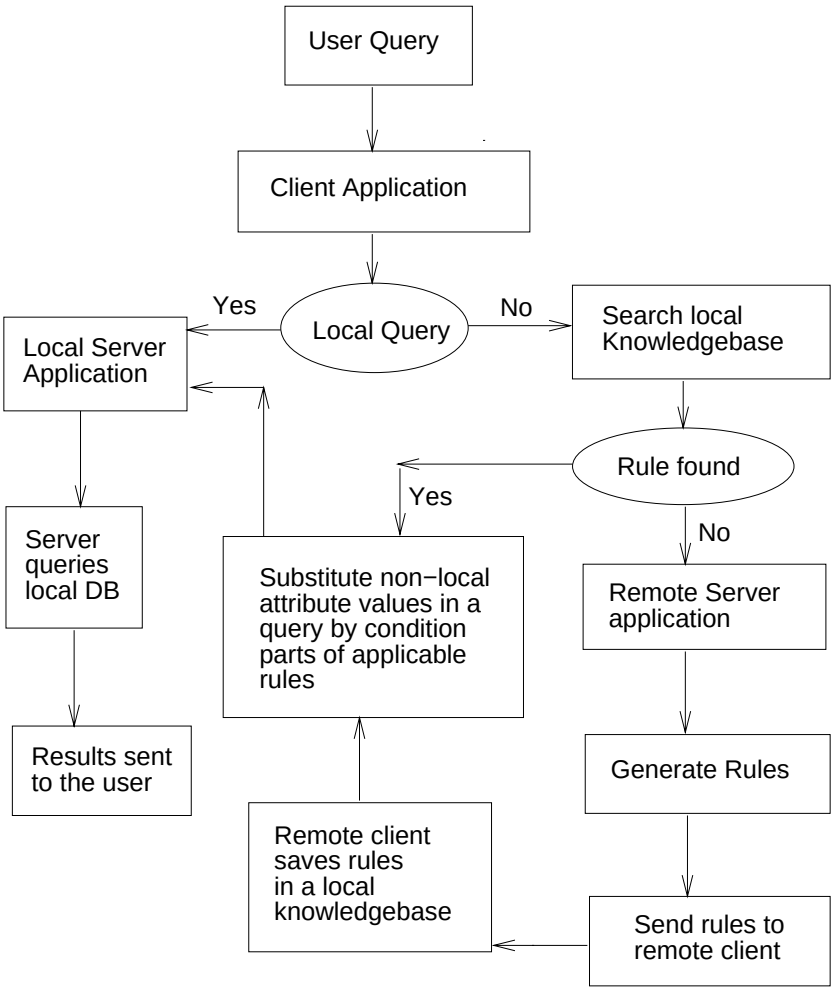

Fig. 2. Query and Rough Query Processing by DAKS

Figure 2 gives a flowchart of a query transformation process in QAS assuming that local query semantics at all contacted sites is the same and $DAKS$ is consistent (granularity levels of the same attributes at remote sistes and the client site are the same). This flowchart will be replaced by two similar flowcharts (corresponding to M_{min} and M_{max}) if semantics at contacted remote sites and a client site differ. Semantics M_{min} and M_{max} can be seen jointly as a rough semantics represented in a very general way by Figure 3.

Saying another words, an optimal, non-rough semantics (if it only exists) suitable for the whole query transformation process is somewhere between two semantics M_{min} and M_{max}.

The word *rough* is used here in the sense of rough sets theory (see [Pawlak][8]).

If we increase the number of sites from which definitions of non-local attributes are collected and then resolve inconsistencies among them (see [Ras][10]), the local confidence in resulting definitions is expected to be higher since they represent consensus of more sites. At the same time, if the number of remote sites involved in a query transformation process is increased, the number of different semantics may increase as well which in result may also increase the roughness of the answer to the query.

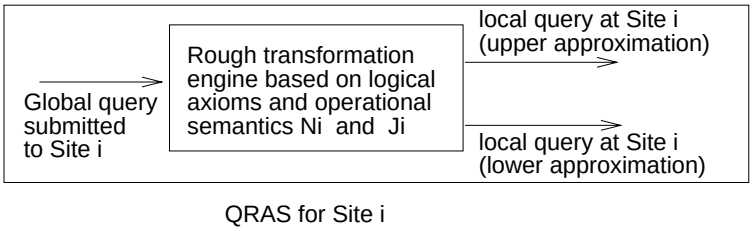

Fig. 3. Query Rough Answering System (QRAS)

Assume now that one of the users needs to retrieve objects from the system at *Site 1* satisfying a global query q and *Site 2* of $DAKS$ is contacted to find definition of an attribute value w listed in q. Attributes in the overlap between systems at *Site 1* and *Site 2* have different granularity levels. Let's say that attributes a, b from the overlap are used in the definition of w. Assume that values of attribute a at *Site 1* are more general than its values at the *Site 2*. Knowing the semantics relationship between these two granularity levels, we can replace values of a at *Site 2* by values from *Site 1* to learn w. Definitions of w extracted at *Site 2* can be directly applied at site *Site 1* to clarify the meaning of q for the *Site 1*. Assume now that values of attribute b at *Site 1* are more specific than at the *Site 2*. Again, knowing the semantics relationship between these two granularity levels, we learn definitions of values of b at *Site 2* which are generalizations of values requested by *Site 1*. This way, a global query submitted to *Site 1* is replaced by a more general query which is local so it can be handled locally at *Site 1*.

We claim that the semantics relationships between different granularity levels of the same attribute, due to its presence at minimum two sites of $DAKS$, should be represented by a hierarchical weighted graph as a part of a global (task) ontology for $DAKS$. The need of weighted graphs instead of hierarchical attributes is justified by the fact that some sites in $DAKS$ may not agree on the same relationships between attribute values of multiple granularity.

4 Query processing based on reducts

In this section we recall the notion of a reduct (see [Pawlak][8]) and show how the partially ordered set of semantics $(\Omega, \preceq)$ can be used to improve query answering process in $DAKS$ as introduced in [Ras][12].

Let us assume that $S = (X, A, V)$, is an information system and $V = \bigcup\{V_a : a \in A\}$. Let $B \subset A$. We say that $x, y \in X$ are indiscernible by B, denoted $[x \approx_B y]$, if $(\forall a \in B)[a(x) = a(y)]$.

Now, assume that both B_1, B_2 are subsets of A. We say that B_1 depends on B_2 if $\approx_{B_2} \subset \approx_{B_1}$. Also, we say that B_1 is a covering of B_2 if B_2 depends on B_1 and B_1 is minimal.

By a reduct of A in S (for simplicity reason we say A-reduct of S) we mean any covering of A.

Example. Assume the following scenario:

- $S_1 = (X_1, \{c, d, e, g\}, V_1)$, $S_2 = (X_2, \{a, b, c, d, f\}, V_2)$, $S_3 = (X_3, \{b, e, g, h\}, V_3)$ are information systems,
- User submits a query $q = q(c, e, f)$ to the query answering system QAS associated with system S_1,
- Systems S_1, S_2, S_3 are parts of $DAKS$.

Attribute f is non-local for a system S_1 so the query answering system associated with S_1 has to contact other sites of $DAKS$ requesting a definition of f in terms of $\{d, c, e, g\}$. Such a request is denoted by $< f : d, c, e, g >$. Assume that the system S_2 is contacted. The definition of f, extracted from S_2, involves only attributes $\{d, c, e, g\} \cap \{a, b, c, d, f\} = \{c, d\}$. There are three f-reducts (coverings of f) in S_2. They are: $\{a, b\}, \{a, c\}, \{b, c\}$. The optimal f-reduct is the one which has minimal number of elements outside $\{c, d\}$. Let us assume that $\{b, c\}$ is chosen as an optimal f-reduct in S_2.

Then, the definition of f in terms of attributes $\{b, c\}$ will be extracted from S_2 and the query answering system of S_2 will contact other sites of DKS requesting a definition of b (which is non-local for S_1) in terms of attributes $\{d, c, e, g\}$. If definition of b is found, then it is sent to QAS of the site 1. Figure 4 shows the process of resolving query q in the example above. Since the set $\{e\}$ is one of the coverings of b in the system contacted by S_2, then the search for a definition of f is completed with a help of two systems.

Now, let us go back to the process of resolving global query $q = q(c, e, f)$ visually shown in Figure 4. Clearly, the resulting query depends only on attributes c and e so it can be processed by local QAS for the site 1. If semantics

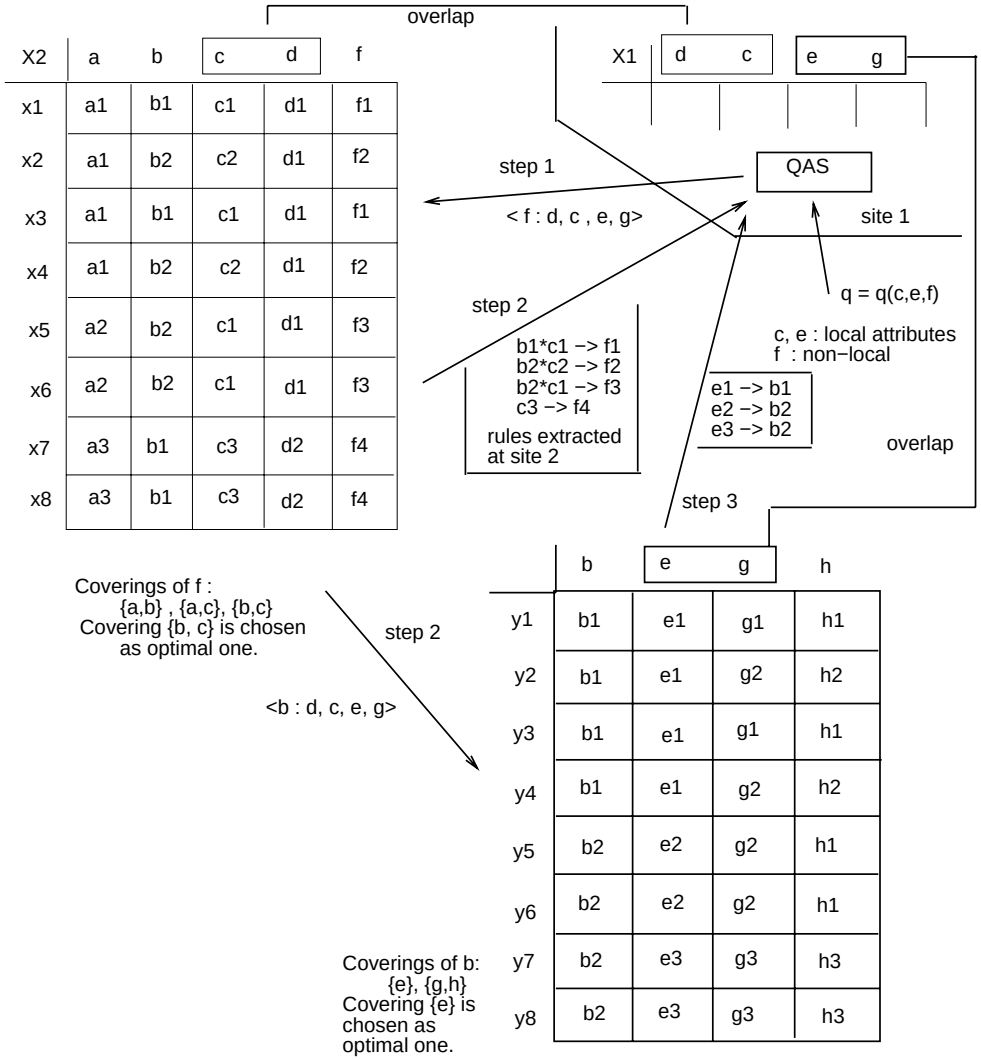

Fig. 4. Process of resolving a query by QAS in $DAKS$

of queries differ at our three involved sites then previously presented transformation steps for query q are not legal because different semantics can not be mixed. One way to handle this problem is to construct a rough-semantics common for all three sites.

To consider the most general scenario, assume now that $\{S_1, S_2, ..., S_{k_1}\}$ is a group of systems in $DAKS$ which are contacted by S in order to help S to understand a global query q. Also, let us assume that $\{S_1, S_2, ..., S_{k_1}\}$ is one of the possible groups of systems which can be contacted by S in the process of solving q. Now, each system S_i, $(1 \leq i \leq k_1)$, may still have to contact recursively a new group of systems $S_{(i,1)}, S_{(i,2)}, ..., S_{(i,n_i)}$ in $DAKS$ to look for the best definitions of subterms of q which are not understood by S. This process will continue till all systems contacted in last recursive calls are able to provide optimal definitions of values of attributes in terms of attributes local for S.

Figure 5 shows the tree of recursive calls constructed when solving query q.

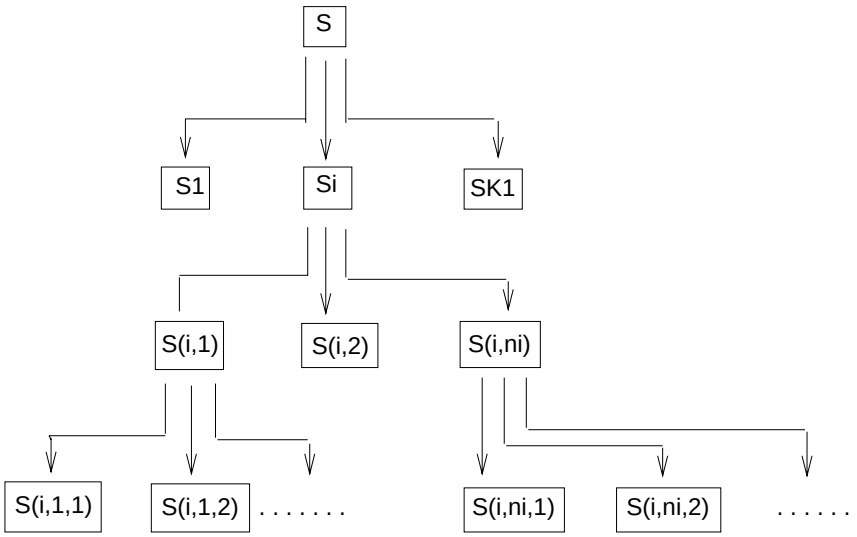

Fig. 5. Recursive calls tree driven by query q submitted to S

Clearly each node of this tree represents an autonomous information system with its own interpretation of queries. So, definitions of values of attributes have interpretations the same as interpretations assigned to systems from which these definitions have been extracted. Rules discovered from the system $S(i,k,j)$ have the same interpretation as the one assigned to queries for $S(i,k,j)$. To solve a query q submitted to S (from Figure 5) definitions received from all nodes of the tree have to be used which means that the problem related to mixing their corresponding local semantics requires our attention.

Assume that query answering systems linked with sites of $DAKS$ use only semantics from Ω, where $(\Omega, \preceq)$ is a partially ordered set of local semantics introduced in the previous section. Now, any global query submitted to S may generate many trees similar to the one at Figure 5. Clearly, the best tree for q will be among trees of smallest height and with nodes having local semantics assigned to them which are maximally close in $(\Omega, \preceq)$ to the local semantics assigned to S. If $\{M_i\}_{i \in I}$ is a set of all local semantics assigned to nodes of the tree T used to process query q, then we may process the query by extracting definitions of global attributes at *level* i of tree T and next use them at *level i-1* of T. This process has to be done recursively till the root of T is reached. During this recursive procedure we do not have to deal with inconsistencies between local semantics, since we already know that the final meaning of the resulting query will be between boundaries $M_{min} = \bigcap\{M_i\}_{i \in I}$ and $M_{max} = \bigcup\{M_i\}_{i \in I}$. Clearly to have these two boundaries uniquely defined we need to assume that $(\Omega, \preceq)$ is a lattice.

These two boundaries are generated by the Query Rough Answering System ($QRAS$) for $DAKS$ based on $(\Omega, \preceq)$. $QRAS$ is represented by Figure 6, where M(min), M(max) are two boundary semantics generated for global query q

submitted to S. Clearly, M(min) and M(max) stand for M_{min} and M_{max}.

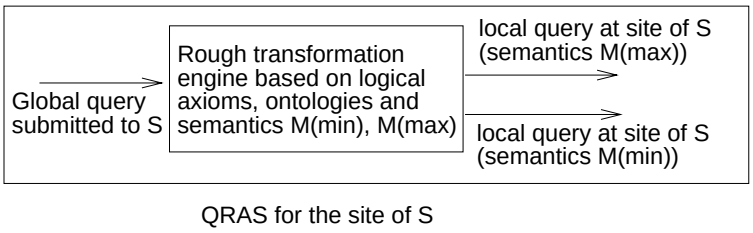

Fig. 6. Query Rough Answering System (QRAS)

Let us assume that $(\Omega, \preceq)$ presented by Figure 7 is a part of a global (task) ontology for $DAKS$ and query $q = q(a)$ is submitted to the query answering system of S which has local semantics M_8. Attribute a listed in q is not local for S. If we do not introduce the notion of a distance between nodes in $(\Omega, \preceq)$ reflecting definitions of M_i, $i \in I$, then M_2, M_3, and M_9 are the closest local semantics to M_8. So, all sites with local semantics M_8 assigned to their queries have the first preference to be asked for help by S in solving q. The next preference should be given to all sites with local semantics M_2, M_3, and M_9 assigned to them.

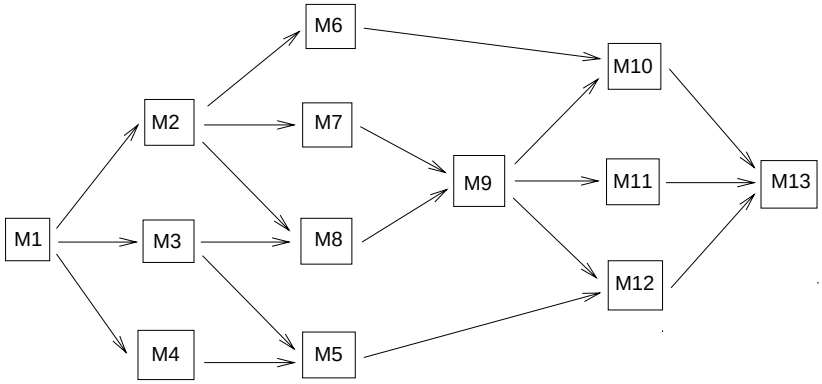

Fig. 7. Sample task ontology $(\Omega, \preceq)$

Assuming that systems $S_i(A_i)$, $i \in I$, can be asked by $S(A)$ for help to solve $q(a)$, the second parameter which is worth to consider here is the number of attributes in a-reduct of $S_i(A_i)$ which are outside the overlap $A \cap A_i$. The most preferable sites to be asked for help are all these sites which have that number the smallest. This way we may minimize the number of sites involved in processing query $q(a)$ which in turn may give us much smaller number of semantics used to process the initial query. Clearly, we assume here that this strategy is applied recursively till no attributes are outside the corresponding overlap.

Query answering system for handling global queries in S can be implemented as a heuristic algorithm with a preference to contact first all sites in $DAKS$ with local semantics which are maximally close in $(\Omega, \preceq)$ to the semantics assigned to S. This strategy seems to be the most reasonable because assuming

that the number of sites in $DAKS$ is rather large, we should not have any problems to identify enough number of sites worth to contact. Also, by getting help only from sites with local semantics which is either equal or close to the semantics assigned to S, the roughness of the resulting local semantics of global queries at S is guaranteed to be rather small.

5 Conclusion

Clearly, the easiest way to solve semantics inconsistencies problem is to apply the same local semantics at all remote sites. However when databases are incomplete and we replace their null values using rule-based chase algorithms based on rules locally extracted then we are already committed to the semantics used by these algorithms. If we do not keep track what and how the null values have been replaced by rule-based chase algorithms, there is no way back for us. Also, it sounds rather unrealistic that one local semantics for incomplete databases can be chosen as a standard. We claim that in such cases a partially ordered set of local semantics $(\Omega, \preceq)$ or its equivalent structure should be a part of *task* ontologies to provide this new meta-data information needed to solve the semantics inconsistencies problem.

References

[1] Andreasen, T., Nilsson, J.F.,Thomsen, H.E., "Ontology-based quering", in *Proceedings of the Flexible Query Answering Systems*, Advances in Soft Computing, Physica-Verlag, 2000, 15-26

[2] Cardoso, J., Sheth, A., "Semantic e-workflow composition", in *International Journal of Intelligent Information Systems*, Kluwer, will appear

[3] Maluf, D., Wiederhold, G., "Abstraction of representation for interoperation", in *Foundations of Intelligent Systems*, Proceedings of the Tenth International Symposium, LNCS/LNAI, Springer-Verlag, No. 1325, 1997, 441-455

[4] Meersman, R.A., "Semantic ontology tools in IS design", in *Foundations of Intelligent Systems*, Proceedings of Eleventh International Symposium, LNCS/LNAI, Springer-Verlag, No. 1609, 1999, 30-45

[5] Meersman, R.A., "Ontologies and databases: more than a fleeting resemblance", in *Proceedings of OES/SEO Workshop*, Rome, Italy, September 14-15, 2001

[6] Mizoguchi, R., "Ontological engineering: foundation of the next generation knowledge processing", in *Proceedings of Web Intelligence: Research and Development*, LNCS/LNAI, Springer-Verlag, No. 2198, 2001, 44-57

[7] Navathe, S., Donahoo, M., "Towards intelligent integration of heterogeneous information sources", in *Proceedings of the Sixth International Workshop on Database Re-engineering and Interoperability*, 1995

[8] Pawlak, Z., "Rough classification", in *International Journal of Man-Machine Studies*, Vol. 20, 1984, 469-483

[9] Prodromidis, A.L. & Stolfo, S., "Mining databases with different schemas: Integrating incompatible classifiers", in *Proceedings of The Fourth Intern. Conf. onn Knowledge Discovery and Data Mining*, AAAI Press, 1998, 314-318

[10] Ras, Z., "Dictionaries in a distributed knowledge-based system", in *Concurrent Engineering: Research and Applications, Conference Proceedings*, Pittsburgh, Penn., Concurrent Technologies Corporation, 1994, 383-390

[11] Ras, Z., "Resolving queries through cooperation in multi-agent systems", in *Rough Sets and Data Mining* (Eds. T.Y. Lin, N. Cercone), Kluwer Academic Publishers, 1997, 239-258

[12] Ras, Z., "Query answering based on distributed knowledge mining", in *Intelligent Agent Technology, Research and Development*, Proceedings of IAT'01 (Eds. N. Zhong, J. Lin, S. Ohsuga, J. Bradshaw), World Scientific, 2001, 17-27

[13] Ras, Z., Dardzinska, A., "Handling semantics inconsistencies in query answering based on distributed knowledge mining", in *Foundations of Intelligent Systems*, Proceedings of ISMIS'02 Symposium, LNCS/LNAI, No. 2366, Springer-Verlag, 2002, 66-74

[14] Ras, Z., Joshi, S.,"Query approximate answering system for an incomplete DKBS", in *Fundamenta Informaticae*, IOS Press, Vol. 30, No. 3/4, 1997, 313-324

[15] Ras, Z., Żytkow, J., "Mining for attribute definitions in a distributed two-layered DB system", *Journal of Intelligent Information Systems*, Kluwer, Vol. 14, No. 2/3, 2000, 115-130

[16] Ras, Z., Żytkow, J.,"Discovery of equations to augment the shared operational semantics in distributed autonomous BD System", in *PAKDD'99 Proceedings*, LNCS/LNAI, No. 1574, Springer-Verlag, 1999, 453-463

[17] Subrahmanian, V.S., "Heterogeneous agent systems", in *Foundations of Intelligent Systems*, Proceedings of Eleventh International Symposium, LNCS/LNAI, Springer-Verlag, No. 1609, 1999, 46-55

[18] Sumi, K., Mizoguchi, R., "Consensus formation support system via ontology generation", in *Proceedings of the PRICAI'02 International Workshop on Intelligent Media Technology for Communicative Reality*, 2002