Solving Failing Queries through Cooperation and Collaboration

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

[1] University of North Carolina, Department of Computer Science, Charlotte, N.C. 28223, USA
[2] Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland
[3] Bialystok Technical Univ., Dept. of Mathematics, ul. Wiejska 45A, 15-351 Bialystok, Poland

Abstract. Sometime Query Answering Systems (QAS) for a Distributed Autonomous Information System ($DAIS$) may fail by returning the empty set of objects as an answer for a query q. Systems in $DAIS$ can be incomplete, have hierarchical attributes, and the semantics of attributes and their values may differ between sites. Also, if there are no objects in S matching q, the query may fail when submitted to S. Alternatively, QAS for S may try to relax the query q as it was proposed in [8], [9], and [2]. It means that q can be replaced by a new more general query. Clearly, the goal is to find possibly the smallest generalization of q which will not fail in S. Smaller generalizations guarantee higher confidence in objects returned by QAS. Such QAS is called cooperative (only one site is involved). Queries may also fail in S when some of the attributes listed in q are outside the domain of S. To resolve this type of queries, assuming that S is a part of $DAIS$, we may extract definitions of such attributes from information systems residing at some of the remote sites for S and next used them to approximate q in S. In order to do that successfully, we assume that all involved systems have to agree on the ontology of some of their common attributes [15], [16], [17]. QAS based on the above strategy is called collaborative (minimum two sites are involved). Similarly, a query may fail in S when the granularity of an attribute used in q is finer than the granularity of the same attribute in S. This paper shows how to use collaboration and cooperation approach to solve failing queries in $DAIS$ assuming that attributes are hierarchical. Some aspects of a collaboration strategy dealing with failing query problem for non-hierarchical attributes have been presented in [15], [16].

1 Introduction

Distributed Autonomous Information System ($DAIS$) is a system that connects a number of autonomous information systems using network communication technology. In this paper, we assume that some of these systems have hierarchical attributes and some of them are incomplete. Incompleteness is understood by allowing to have a set of weighted attribute values as a value of an attribute.

Additionally, we assume that the sum of these weights has to be equal 1. The definition of an information system of type λ and distributed autonomous information system used in this paper was given by Raś and Dardzińska in [16]. The threshold λ was used to check the new weights assigned to values of attributes by *Chase* algorithm [5]. If the weight assigned by *Chase* to one of the attribute values for a given object x was less than the allowed threshold value, then this attribute value was deleted from the set of possible values for x. Semantic inconsistencies among sites in $DAIS$ are due to different interpretations of attributes and their values (for instance one site can interpret the concept *young* differently than another site). Ontologies ([10], [11], [18], [19], [20], [1], [23], [7], [21]) are usually used to handle differences in semantics among information systems. If two systems agree on the ontology associated with attribute *young* and its values, then attribute *young* can be used as a semantical bridge between these systems. Different interpretations are also due to the way each site is handling null values. Null value replacement by a value predicted either by statistical or some rule-based methods is quite common before queries are answered by QAS. In [15], the notion of *rough semantics* and a method of its construction was proposed. The rough semantics can be used to model and properly handle semantic inconsistencies among sites due to different interpretations of incomplete values. There are cases when a Query Answering System (QAS) either for an autonomous or for a Distributed Information System ($DAIS$) may fail to return a satisfactory answer to a submitted query. For instance, let us assume that an information system S has hierarchical attributes and there is no single object in S which description matches a query q submitted by a user. Assuming that a distance measure between objects in S is defined, then by generalizing q, we may identify objects in S which descriptions are nearest to the description q. We may also face failing query problem when some of the attributes listed in a query are outside the domain of a queried information system S. The way to solve this problem, proposed by Ras [14], is to extract definitions of such attributes at one of the remote sites for S in $DAIS$ and next used them in S. This problem is similar to the problem when the granularity of the attribute value used in a query q is finer than the granularity of the corresponding attribute used in S. By replacing this attribute value in q by the one used in S, we can retrieve objects from S which possibly satisfy q. Instead of doing that, we may compute definitions of this attribute value at one of the remote sites for S in $DAIS$ and next used them by QAS to enhance the process of identifying which objects in S satisfy that query. However, to do that, we need to know that both systems involved in a collaboration process also agree on the ontology of some of their common attributes [15], [16], [17]. Additionally, the granularity level of the attribute which definition is remotely computed should be the same at the remote site and in q. In this paper, we present a new methodology for the failing query problem in $DAIS$ addressing all the above problems and solutions to handle them.

2 Query Processing with Incomplete Data

In real life, information about objects is collected and stored in information systems which are autonomous and reside at different locations. These systems are either complete or incomplete and their attributes may have different granularity levels. For instance, at one of them, only concepts *child*, *young*,*middle-aged*, *old*, *senile* can be used as values of the attribute *age*. At another information system, integers are used as values of the attribute *age*. If both systems agree on the semantics related to attribute *age* and its values mentioned earlier, they can easily use this attribute to communicate with each other. Also, it is very possible that an attribute is missing in one of the systems while it occurs at many others. Assume that user submits a query to a Query Answering System (QAS) for S (called a client) but some of the attributes used in the query are either missing in S or their granularity is finer than the granularity of the same attributes at S. In such cases, QAS can request definitions of these attributes from other information systems in $DAIS$. These definitions are stored in the knowledge base for S and used to chase (see [4]) the missing values and, if needed, to refine the current values of attributes in S. Algorithm Chase for $DAIS$, based on rules, was given by Dardzińska and Raś in [4]. This algorithm can be easily modified and used for refinement of object descriptions in S. To extend this algorithm to a distributed environment, the problem of semantic inconsistencies (due to different interpretations of incomplete values) among sites has to be resolved first. For instance, it can be done by taking rough semantics [15], mentioned earlier. In any case, collaborating sites have to agree on the ontology associated with their common attributes. To have a complete model for solving queries in a distributed environment, we should also consider user ontology related to attributes and their values which clearly may differ from the ontology linked with an information system. However, for simplicity reason, this paper is only focused on "system-system" type of collaboration.

Definition 1:
We say that $S = (X, A, V)$ is an incomplete information system of type λ, if the following four conditions hold:

- X is the set of objects, A is the set of attributes, and $V = \bigcup\{V_a : a \in A\}$ is the set of values of attributes,
- $(\forall x \in X)(\forall a \in A)[a_S(x) \in V_a$ or $a_S(x) = \{(v_i, p_i) : v_i \in V_a \wedge$ $p_i \in [0,1] \wedge 1 \leq i \leq m\}]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}) \rightarrow \sum_{i=1}^{m} p_i = 1]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}) \rightarrow (\forall i)(p_i \geq \lambda)]$.

Definition 2:
By a Distributed Autonomous Information System ($DAIS$), we mean any collection of incomplete information systems of type λ.

Now, let us assume that S_1, S_2 are incomplete information systems, both of type λ. The same objects (from X) are stored in both systems and the same attributes (from A) are used to describe them. The meaning and granularity of values of attributes from A in both systems S_1, S_2 is also the same. Additionally, we assume that $a_{S_1}(x) = \{(v_{1i}, p_{1i}) : 1 \leq m_1\}$ and $a_{S_2}(x) = \{(v_{2i}, p_{2i}) : 1 \leq m_2\}$.

We say that containment relation Ψ holds between S_1 and S_2, if the following two conditions hold:

- $(\forall x \in X)(\forall a \in A)[card(a_{S_1(x)}) \geq card(a_{S_2(x)})]$,
- $(\forall x \in X)(\forall a \in A)[[card(a_{S_1}(x)) = card(a_{S_2}(x))] \rightarrow [\sum_{i \neq j} |p_{2i} - p_{2j}| > \sum_{i \neq j} |p_{1i} - p_{1j}|]]$.

Instead of saying that containment relation holds between S_1 and S_2, we can equivalently say that S_1 was transformed into S_2 by containment mapping Ψ. This fact can be presented as a statement $\Psi(S_1) = S_2$ or $(\forall x \in X)(\forall a \in A)[\Psi(a_{S_1}(x)) = \Psi(a_{S_2}(x))]$. Similarly, we can either say that $a_{S_1}(x)$ was transformed into $a_{S_2}(x)$ by Ψ or that containment relation Ψ holds between $a_{S_1}(x)$ and $a_{S_2}(x)$.

So, if containment mapping Ψ converts an information system S to S', then S' is more complete than S. Saying another words, for a minimum one pair $(a, x) \in A \times X$, either Ψ has to decrease the number of attribute values in $a_S(x)$ or the average difference between confidences assigned to attribute values in $a_S(x)$ has to be increased by Ψ.

To give an example of a containment mapping Ψ, let us take two information systems S_1, S_2 both of the type λ, represented as Table 1 and Table 2.

X	a	b	c	d	e
x_1	$\{(a_1, \frac{1}{3}), (a_2, \frac{2}{3})\}$	$\{(b_1, \frac{2}{3}), (b_2, \frac{1}{3})\}$	c_1	d_1	$\{(e_1, \frac{1}{2}), (e_2, \frac{1}{2})\}$
x_2	$\{(a_2, \frac{1}{4}), (a_3, \frac{3}{4})\}$	$\{(b_1, \frac{1}{3}), (b_2, \frac{2}{3})\}$		d_2	e_1
x_3		b_2	$\{(c_1, \frac{1}{2}), (c_3, \frac{1}{2})\}$	d_2	e_3
x_4	a_3		c_2	d_1	$\{(e_1, \frac{2}{3}), (e_2, \frac{1}{3})\}$
x_5	$\{(a_1, \frac{2}{3}), (a_2, \frac{1}{3})\}$	b_1	c_2		e_1
x_6	a_2	b_2	c_3	d_2	$\{(e_2, \frac{1}{3}), (e_3, \frac{2}{3})\}$
x_7	a_2	$\{(b_1, \frac{1}{4}), (b_2, \frac{3}{4})\}$	$\{(c_1, \frac{1}{3}), (c_2, \frac{2}{3})\}$	d_2	e_2
x_8		b_2	c_1	d_1	e_3

Table 1. Information System S_1

X	a	b	c	d	e
x_1	$\{(a_1, \frac{1}{3}), (a_2, \frac{2}{3})\}$	$\{(b_1, \frac{2}{3}), (b_2, \frac{1}{3})\}$	c_1	d_1	$\{(e_1, \frac{1}{3}), (e_2, \frac{2}{3})\}$
x_2	$\{(a_2, \frac{1}{4}), (a_3, \frac{3}{4})\}$	b_1	$\{(c_1, \frac{1}{3}), (c_2, \frac{2}{3})\}$	d_2	e_1
x_3	a_1	b_2	$\{(c_1, \frac{1}{2}), (c_3, \frac{1}{2})\}$	d_2	e_3
x_4	a_3		c_2	d_1	e_2
x_5	$\{(a_1, \frac{3}{4}), (a_2, \frac{1}{4})\}$	b_1	c_2		e_1
x_6	a_2	b_2	c_3	d_2	$\{(e_2, \frac{1}{3}), (e_3, \frac{2}{3})\}$
x_7	a_2	$\{(b_1, \frac{1}{4}), (b_2, \frac{3}{4})\}$	c_1	d_2	e_2
x_8	$\{(a_1, \frac{2}{3}), (a_2, \frac{1}{3})\}$	b_2	c_1	d_1	e_3

Table 2. Information System S_2

It can be easily checked that the values assigned to $e(x_1)$, $b(x_2)$, $c(x_2)$, $a(x_3)$, $e(x_4)$, $a(x_5)$, $c(x_7)$, and $a(x_8)$ in S_1 are different than the corresponding values in S_2. In each of these eight cases, an attribute value assigned to an object in S_2 is less general than the value assigned to the same object in S_1. It means that $\Psi(S_1) = S_2$.

We assume again that $S_1 = (X, A, V_1)$, $S_2 = (X, A, V_2)$ are incomplete information systems, both of type λ. Although attributes in both systems are the same, they may also differ in granularity of their values. Let us assume that $a_{S_1}(x) = \{(a_{[1,i]}, p_{[1,i]}) : 1 \leq i \leq m_1\}$ and $a_{S_2}(x) = \{a_1\}$.

We say that containment relation Φ holds between S_2 and S_1, if the following two conditions hold:

- $(\forall i \leq m_1)[a_{[1,i]}$ is a child of a_1 in the ontology part representing hierarchical attribute $a]$,
- either $\{a_{[1,i]} : 1 \leq i \leq m_1\}$ does not contain all children of a_1 or weights in $\{p_{[1,i]} : 1 \leq i \leq m_1\}$ are not all the same.

Instead of saying that containment relation holds between S_2 and S_1, we can equivalently say that S_2 was transformed into S_1 by containment mapping Φ. This fact can be written as $\Phi(S_2) = S_1$ or $(\forall x \in X)(\forall a \in A)[\Phi(a_{S_2}(x)) = \Phi(a_{S_1}(x))]$. Similarly, we can either say that $a_{S_2}(x)$ was transformed into $a_{S_1}(x)$ by Φ or that containment relation Φ holds between $a_{S_2}(x)$ and $a_{S_1}(x)$.

So, if containment mapping Φ converts an information system S to S', then information about any object in S' is more precise than about the same object in S. Clearly, if $\{a_{[1,i]} : 1 \leq i \leq m\}$ contains all children of a_1, then semantically a_1 has the same meaning as $\{(a_{[1,i]}, 1/m) : 1 \leq i \leq m\}$. It can be easily checked that objects x_1, x_2 and also objects x_3, x_4 in Table 3 have equivalent representations.

Clearly, we assume here that $c_{[3,1]}, c_{[3,2]}$ are the only children of c_3, and $V_b = \{b_1, b_2, b_3\}$.

This example also shows that, if $S = (X, A, V)$, $d \notin A$, and $V_d = \bigcup\{d_i : 1 \leq i \leq k_d\}$, then information stored in $S' = (X, A \cup \{d\}, V \cup V_d)$ is equivalent to the information stored in S, where $(\forall x \in X)[d(x) = \{(d_i, \frac{1}{k_d}) : 1 \leq i \leq k_d\}]$. Let us assume that information system S' represented by Table 3 satisfies the following two constraints: $V_d = \{d_1, d_2, d_3\}$, $V_a = \{a_1, a_{[1,1]}, a_{[1,2]}, a_2, a_{[2,1]}, a_{[2,2]}\}$.

X	a	b	c	d
x_1	a_1		c_1	
x_2	a_1	$\{(b_1, \frac{1}{3}), (b_2, \frac{1}{3}), (b_3, \frac{1}{3})\}$	c_1	
x_3	a_2	b_3	c_3	
x_4	a_2	b_3	$\{(c_{[3,1]}, \frac{1}{2}), (c_{[3,2]}, \frac{1}{2})\}$	

Table 3. Information System S extended to S'

There is a clear conceptual similarity between failing query $q_1 = a_{[2,1]}$ and $q_2 = d_2$ submitted to either S or S'. Both q_1 and q_2 will fail in S' because the granularity level of attributes used in queries is finer than the granularity of the same attributes in S'. Clearly, we can generalize q_1 to $q_1' = a_2$ and retrieve two objects x_3, x_4 which possibly satisfy q_1. By generalizing q_2 we will retrieve all objects in S'. If we assume that S' is a part of $DAIS$, then an alternate way to solve this problem is to look for definitions of $a_{[2,1]}$ and d_2 at remote sites for S'.

3 How to Solve Failing Queries in DAIS

Assume now that a query $q(B)$ is submitted to an information system $S = (X, A, V)$, where B is the set of attributes used in q and $A \cap B \neq \emptyset$. All attributes in $B - [A \cap B]$ are called foreign for S. If S is a part of $DAIS$, then for definitions of foreign attributes for S we may look at its remote sites (see [15]). We assume here that two information systems can collaborate only if they agree on the ontology of attributes which are present in both of them. Clearly, the same ontology does not mean that a common attribute is of the same granularity at both sites. Similarly, the granularity of values of attributes used in a query may differ from the granularity of values of the same attributes in S. In [15], it was shown that query $q(B)$ can be processed at site S by discovering definitions of values of attributes from $B - [A \cap B]$ at any of the remote sites for S and use them to answer $q(B)$. With each certain rule discovered at a remote site, a number of additional rules (implied by that rule) is also discovered.

For instance, let us assume that two attributes *age* and *salary* are used to describe objects in S' and in S. Information system S' represents a remote site for S. Both systems accept the ontology, in $LISP$-like notation, given below:

- age(child($\leq$17),
  young(18,19,...,29),
  middle-aged(30,31,...,60),
  old(61,62,...,80),
  senile(81,82,...,$\geq$100))

- salary(low(10K,20K,30K,40K),
  medium(50K,60K,70K),
  high(80K,90K,100K),
  very-high(110K,120K,$\geq$130K))

Now, assume that the constraints $V_{age} = \{young,\ middle\text{-}aged,\ old,\ senile\}$, $V_{salary} = \{n \cdot 10K : n \geq 1\}$ are satisfied by S' and also the certain rule $(age, young) \longrightarrow (salary, 40K)$ is extracted from S'. Jointly with that rule, the following rules are also discovered:

- $[(age, young) \longrightarrow (salary, low)]$,
- $[(age, N) \longrightarrow (salary, 40K)]$, where $N = 18, 19, ..., 29$,
- $[(age, N) \longrightarrow (salary, low)]$, where $N = 18, 19, ..., 29$.

The assumption that the extracted rules have to be certain, in order to generate from them additional rules of high confidence, can be relaxed to "almost" certain. Stronger relaxation is rather risky since, for instance, the rule $r = [(age, N) \longrightarrow (salary, 40K)]$ may occur to be a surprising rule, as defined by Suzuki [22]. If the granularity of values of the attribute *salary* in S is represented by the set $\{low,\ medium,\ high,\ very\text{-}high\}$, then the rules

- $[(age, N) \longrightarrow (salary, 40K)]$, where $N = 18, 19, ..., 29$,

can be used to convert S into a new system with a finer granularity of the attribute *salary* than its granularity in S. This conversion is especially needed when the granularity of values of the attribute *salary* used in query $q(B)$ is finer than its granularity in S. Clearly, we can always replace the user query by a more general query to match the granularity of values of its attributes with a granularity used in S. But, the user may not agree to have his query changed that way.

Now, to be more general, let us assume that $G(A)$ is the set of attributes in $S = (X, A, V)$ which granularity level in $q(B)$ is finer than their granularity level in S. By D we denote the set of rules extracted from each of the remote sites S_1 of S by $ERID(S_1, \lambda_1, \lambda_2)$. Parameters λ_1, λ_2 represent thresholds for minimum support and minimum confidence of rules extracted by $ERID$, correspondingly.

ERID is the algorithm for discovering rules from incomplete information systems, presented by Dardzińska and Raś in [4], [6].

A knowledge-base $L(D)$ is defined as a subset of $\{(t \rightarrow v) \in D : (\exists c \in G(A))[v \in V_c]\}$ containing rules extracted from a remote site S_1 of S only if the granularity level of attributes from $G(A)$ in S_1 is either equal or finer than their granularity in $q(B)$.

The type of incompleteness in [16] is the same as in this paper but we have to assume that any attribute value a_1 in S can be replaced by $\{(a_{1i}, 1/m) : 1 \leq i \leq m\}$, where $\{a_{1i} : 1 \leq i \leq m\}$ is the set of all children of a_1 in the ontology associated with a_1 and accepted by S.

By replacing descriptions of objects in S by new finer descriptions recommended by rules in $L(D)$, we can easily construct a new system $\Phi(S)$ in which $q(B)$ will fail (QAS will return either empty set of objects or set of weighted objects with weights below the threshold value provided by user). In this paper we propose an automated refinement process for object descriptions in S which guarantees that QAS will not fail on $\Phi(S)$ assuming that it does not fail on S. But before we continue this subject any further, another issue needs to be discussed first.

Foreign attributes for S can be seen as attributes which are 100% incomplete in S, that means values of such attributes have to be ascribed to all objects in S. Stronger the consensus among sites on a value to be ascribed to x, *finer* the result of the ascription process for x can be expected. Assuming that systems S_1, S_2 are storing the same sets of objects and using the same attributes to describe them, system S_1 is *finer* than system S_2, if $\Psi(S_2) = S_1$.

We may have a number of rules in the knowledge-base $L(D)$, associated with information system S, which describe the same value of an attribute $c \in G(A)$. For instance, let us assume that $t_1 \rightarrow v_c$, $t_2 \rightarrow v_c$ are such rules. Now, if the granularity of attribute c is the same in these two rules and in a query $q = v_c * t_3$ submitted to QAS and at the same time the granularity of c is more general in S, then these two rules will be used to refine the descriptions of objects in S. First of all, we have to identify all objects in S which have a property $t_3 * (t_1 + t_2)$. This property should be replaced by $(t_3 * t_1) + (t_3 * t_2)$. Otherwise, we may have a problem in identifying correct objects in S. But, to have this replacement done successfully, we have to decide first on the interpretation of functors *or* and *and*, denoted by $+$ and $*$, correspondingly. In this paper, we adopt the semantics of terms proposed by Raś & Joshi in [17] as their semantics has all the properties required for the query transformation process to be sound [see [17]]. It was proved that, under their semantics, the following distributive property holds: $t_1 * (t_2 + t_3) = (t_1 * t_2) + (t_1 * t_3)$.

So, let us assume that $S = (X, A, V)$ is an information system of type λ and t is a term constructed in a standard way (term in predicate calculus) from values of attributes in V seen as *constants* and from two functors $+$ and $*$. By $N_S(t)$, we mean the standard interpretation of a term t in S defined as (see [17]):

- $N_S(v) = \{(x,p) : (v,p) \in a(x)\}$, for any $v \in V_a$,
- $N_S(t_1 + t_2) = N_S(t_1) \oplus N_S(t_2)$,
- $N_S(t_1 * t_2) = N_S(t_1) \otimes N_S(t_2)$,

where, for any $N_S(t_1) = \{(x_i, p_i)\}_{i \in I}$, $N_S(t_2) = \{(x_j, q_j)\}_{j \in J}$, we have:

- $N_S(t_1) \oplus N_S(t_2) =$
  $\{(x_i, p_i)\}_{i \in (I-J)} \cup \{(x_j, p_j)\}_{j \in (J-I)} \cup \{(x_i, max(p_i, q_i))\}_{i \in I \cap J}$,
- $N_S(t_1) \otimes N_S(t_2) = \{(x_i, p_i \cdot q_i)\}_{i \in (I \cap J)}$.

Now, we are ready to discuss the failing query problem in $DAIS$. Let $S = (X, A, V)$ represents one of the sites in $DAIS$, called a client. For simplicity reason, we assume that $A = A_1 \cup A_2 \cup \{a, b, d\}$, $V_a = \{a_1, a_2, a_3\}$, $V_b = \{b_{[1,1]}, b_{[1,2]}, b_{[1,3]}, b_{[2,1]}, b_{[2,2]}, b_{[2,3]}, b_{[3,1]}, b_{[3,2]}, b_{[3,3]}\}$, $V_d = \{d_1, d_2, d_3\}$, and that the semantics of attributes $\{a, b, c, d\}$, used in $DAIS$, is consistent with the ontology defined in LISP-like notation as:

$$
\begin{array}{l}
[a(\\
\quad a_1[a_{[1,1]}, a_{[1,2]}, a_{[1,3]}], \\
\quad a_2[a_{[2,1]}, a_{[2,2]}, a_{[2,3]}], \\
\quad a_3[a_{[3,1]}, a_{[3,2]}, a_{[3,3]}]), \\
\quad b(\\
\quad b_1[b_{[1,1]}, b_{[1,2]}, b_{[1,3]}], \\
\quad b_2[b_{[2,1]}, b_{[2,2]}, b_{[2,3]}], \\
\quad b_3[b_{[3,1]}, b_{[3,2]}, b_{[3,3]}]), \\
\quad c(\\
\quad c_1[c_{[1,1]}, c_{[1,2]}, c_{[1,3]}], \\
\quad c_2[c_{[2,1]}, c_{[2,2]}, c_{[2,3]}], \\
\quad c_3[c_{[3,1]}, c_{[3,2]}, c_{[3,3]}]), \\
\quad d(\\
\quad d_1[d_{[1,1]}, d_{[1,2]}, d_{[1,3]}], \\
\quad d_2[d_{[2,1]}, d_{[2,2]}, d_{[2,3]}], \\
\quad d_3[d_{[3,1]}, d_{[3,2]}, d_{[3,3]}])].
\end{array}
$$

So, the set $\{a_1, a_2, a_3\}$ represents the values of attribute a at its first granularity level. The set $\{a_{[1,1]}, a_{[1,2]}, a_{[1,3]}, a_{[2,1]}, a_{[2,2]}, a_{[2,3]}, a_{[3,1]}, a_{[3,2]}, a_{[3,3]}\}$ represents the values of attribute a at its second granularity level. Clearly, the value a_1 can be refined to any value from $\{a_{[1,1]}, a_{[1,2]}, a_{[1,3]}\}$. Similarly, a_2 can be refined to any value from $\{a_{[2,1]}, a_{[2,2]}, a_{[2,3]}\}$ and a_3 can be refined to any value from $\{a_{[3,1]}, a_{[3,2]}, a_{[3,3]}\}$. Attributes in $\{b, c, d\}$ and their values have analogous representation and properties. So, $a_{[i,j]}$ is finer than a_i, for any i, j.

Now, let us assume that query $q = a_{[i,1]} * b_i * c_{[i,3]} * d_i$ is submitted to S, where $1 \leq i \leq 3$. We present several possible strategies based on cooperation and collaboration among sites in $DAIS$ which QAS can follow to solve q.

The first one is to generalize $a_{[i,1]}$ to a_i and $c_{[i,3]}$ to c. Generalizing $c_{[i,3]}$ to c is equivalent to the removal of $c_{[i,3]}$ from q. This way query q is replaced by

a new more general query $q_1 = a_i * b_i * d_i$. Now, either there are objects in S satisfying q_1 or further generalization of q_1 is needed. In both cases, we can only say that objects matching q_1 may satisfy q. Clearly, further generalization of q is decreasing the chance that retrieved objects will match q. Since the granularity of attributes used in S is either the same or finer than the granularity of attributes in q_1, we can easily identify objects in S satisfying q_1.

To use cooperative query answering approach (see [8], [9], [2]) in solving q, we have to replace it by an equivalent query $q_2 = a_{[i,1]} * [b_{[i,1]} + b_{[i,2]} + b_{[i,3]}] * c_{[i,3]} * d_i = [a_{[i,1]} * b_{[i,1]} * c_{[i,3]} * d_i] + [a_{[i,1]} * b_{[i,2]} * c_{[i,3]} * d_i] + [a_{[i,1]} * b_{[i,3]} * c_{[i,3]} * d_i]$ and next generalize its term $c_{[i,3]}$ to c. The resulting query will be: $q_3 = [a_{[i,1]} * b_{[i,1]} * d_i] + [a_{[i,1]} * b_{[i,2]} * d_i] + [a_{[i,1]} * b_{[i,3]} * d_i]$. Now, we notice that the granularity level of the attribute a in q_3 is finer than the granularity level of a in S. The easiest way to approach this problem is to replace $a_{[i,1]}$ by a_i in q_3, which will give us a new query $q_4 = [a_i * b_{[i,1]} * d_i] + [a_i * b_{[i,2]} * d_i] + [a_i * b_{[i,3]} * d_i]$. If QAS returns the empty set of objects as the response to q_4, we can follow the cooperative query answering approach to find the nearest objects in S matching q_4. If QAS returns a non-empty set of possible objects as the response to q_4, we can either accept that set or look for alternative methods to answer queries.

Now, let us consider query q_1, q_3, or q_4. Each of these queries is a generalization of q by removal of the attribute value $c_{[1,2]}$ from it. In this section we concentrate on q_1 but the same discussion covers equally well the remaining two queries. Assume that there is a non-empty set of objects in S satisfying q_1. Clearly, some of these objects may have a property $c_{[i,3]} * a_{[i,1]}$ but can we identify them with a reasonably high accuracy?

Following the approach proposed by Raś [14], we can discover rules at the remote sites for S and use them to predict which of these objects satisfy the property $a_{[i,1]} * c_{[i,3]}$ and the same which of them satisfy query q. To be more precise, we search $DAIS$ for a site which has overlapping attributes with S, including attributes a, c. The granularity level of attributes a, c has to be the same as their granularity level in q. For instance, if S_1 is identified as such a site, we extract definitions of terms $a_{[i,1]}$ and $c_{[i,3]}$ from that site in terms of common attributes for S and S_1. These definitions are used to identify which objects retrieved by q_1 will match query q with a high probability of success. This type of approach is classified as collaborative approach.

Distributive property $t_1 * (t_2 + t_3) = (t_1 * t_2) + (t_1 * t_3)$, mentioned earlier, is also important in the collaborative approach because of the possibility to replace values of attributes used in q by terms defining them which are disjuncts. For instance, if rules $u_1 \longrightarrow a_{[i,1]}$, $w_1 \longrightarrow a_{[i,1]}$, $u_2 \longrightarrow c_{[i,3]}$, and $w_2 \longrightarrow c_{[i,3]}$ are extracted at S_1, then the term $(u_1 + w_1) * (u_2 + w_2)$ can be used to replace $a_{[i,1]} * c_{[i,3]}$ in query q. Now, assuming that the attribute values used in $(u_1 + w_1) * (u_2 + w_2)$ are all local in S and their granularity level is the same as in S, then objects having the property $a_{[i,1]} * c_{[i,3]}$ can be easily identified in S. A rule extracted at S_1 can be used to get new finer descriptions of objects in S only if they do not contradict with the current descriptions of objects in S.

For instance, if a rule $r = [b_1 \longrightarrow a_{[1,1]}]$ is extracted at S_1 and there is an object x in S such that $b(x) = b_1$ and $a(x) = a_2$, then the rule r can not be used for refinement of attribute values in S.

Finally, it can happen that the replacement of $a_{[i,1]} * c_{[i,3]}$ by $(u_1 + w_1) * (u_2 + w_2)$ may convert q to a query q_5 which will fail in S. Saying another words, system QAS will return the empty set of objects as the response to q_5. In this case, a possible solution is to start with a query q_4 and look for its optimal finest replacement taking into consideration attribute c and attribute value a_i. We can either refine a_i to $a_{[i,1]}$ or a_i to $a_i * c_i$. In order to decide which one to follow, we consider two sequences: $[a_{[i,1]}, b_{[i,1]} + b_{[i,2]} + b_{[i,3]}, d_i]$ and $[c_i, a_i, b_{[i,1]} + b_{[i,2]} + b_{[i,3]}, d_i]$. If the first sequence has more objects in S_1 supporting it, then the refinement a_i to $a_{[i,1]}$ is optimal and it is tried first. Otherwise, the refinement a_i to $a_i * c_i$ is chosen. If the refined query does not fail in S, then the process is continued.

4 Solving Queries by Collaboration

In this section we present one more example to clarify further all proposed approaches to query processing based on cooperation and collaboration.

Let us assume that we have two information systems S and S_1 which for a simplicity reason are complete. They are presented as Table 4 and Table 5, correspondingly.

Y	a	b	c	e	f
y_1	$a_{[1,2]}$	b_1	c_2	e_1	f_1
y_2	$a_{[1,2]}$	b_1	c_2	e_1	f_1
y_3	$a_{[1,1]}$	b_1	c_1	e_1	f_2
y_4	$a_{[2,1]}$	b_2	c_2	e_2	f_2
y_5	$a_{[1,1]}$	b_2	c_2	e_2	f_3

Table 4. Information System S

Assume now that query $q = a_{[1,2]} * b_{[1,1]} * f_{[1,2]}$ is submitted to S. One way to solve it is to generalize q to $q_1 = a_{[1,2]} * b_1 * f_1$. Objects y_1, y_2 possibly satisfy q. Another option for solving q is to use help from S_1 to get finer descriptions of objects in S. The first step is to extract definitions of attribute values $b_{[1,1]}$, $f_{[1,2]}$ in terms of the common attributes for S and S_1. Clearly, we assume here that both systems agree on the ontology of their common attributes and their values. The following certain rules can be extracted from S_1:

$[a_1 * c_2 \longrightarrow b_{[1,1]}]$, $[f_1 * c_2 \longrightarrow b_{[1,1]}]$, $[f_2 * a_1 \longrightarrow b_{[1,1]}]$,
$[b_2 \longrightarrow f_{[1,2]}]$, $[c_1 \longrightarrow f_{[1,2]}]$.

X	a	b	c	d	f
x_1	a_1	b_2	c_1	d_1	$f_{[1,2]}$
x_2	a_2	b_1	c_2	d_2	$f_{[2,1]}$
x_3	a_1	$b_{[1,1]}$	c_2	d_2	$f_{[1,2]}$
x_4	a_1	$b_{[1,1]}$	c_2	d_1	$f_{[2,1]}$
x_5	a_2	b_2	c_1	d_2	$f_{[1,2]}$

Table 5. Information System S_1

Now, we replace query q_1 by a new finer query $q_2 = a_{[1,2]} * b_{[1,1]} * f_1$. Since the rules defining $b_{[1,1]}$ and extracted from S_1 do not contradict information about objects stored in S, we can use them to replace q_2 by a new query

$q_3 = a_{[1,2]} * [a_1 * c_2 + f_1 * c_2 + f_2 * a_1] * f_1 = [a_{[1,2]} * a_1 * c_2 * f_1] + [a_{[1,2]} * f_1 * c_2 * f_1] + [a_{[1,2]} * f_2 * a_1 * f_1] = [a_{[1,2]} * c_2 * f_1]$.

It can be easily checked that both objects satisfy query q_2. Now, we replace query q_1 by a new finer query $q_3 = a_{[1,2]} * b_1 * f_{[1,2]}$. Since the rules defining $f_{[1,2]}$ and extracted from S_1 contradict the information about objects stored in S, we can not use them to replace q_3 by a new query. So, $q_2 = a_{[1,2]} * b_{[1,1]} * f_1$ is the smallest generalization of the query q submitted to S which can be answered with a help from the system S_1.

Clearly, in a general scenario, we may have many successful generalizations of a failing query submitted to S. In order to decide which one is optimal, we need to define a distance measure ρ between queries. We start with a definition of a distance between values of hierarchical attributes.

$\rho[a_{[i_1,i_2,...,i_n]}, a_{[j_1,j_2,...,j_m]}] = \{$
if $i_1 \neq j_1$, then 1;
if $[(\forall p \leq k)[i_p = j_p] \wedge [i_{k+1} \neq j_{k+1}]]$, then $\frac{1}{2^k}$;
if $[n < m] \wedge [(\forall p \leq n)[i_p = j_p]$, then $\frac{1}{2^n}$;
if $[m < n] \wedge [(\forall p \leq m)[i_p = j_p]$, then $\frac{1}{2^m}$;
if $[m = n] \wedge [(\forall p \leq m)[i_p = j_p]$, then $0\}$.

Now, the distance between queries
$q_1 = a_{i1} * b_{i2} * c_{i3} * .... * d_{ip}$, $q_2 = a_{j1} * b_{j2} * c_{j3} * .... * d_{jp}$
is defined as follows:

$\rho[q_1, q_2] = \sum\{\rho[a_{ik}, a_{jk}] : 1 \leq k \leq p\}$.

In the previous example, the distance between query $q = a_{[1,2]} * b_{[1,1]} * f_{[1,2]}$ submitted to S and its initial generalization $q_1 = a_{[1,2]} * b_1 * f_1$ is equal to $\frac{1}{2} + \frac{1}{2} = 1$, whereas the distance from q to $q_2 = a_{[1,2]} * b_{[1,1]} * f_1$ is equal $\frac{1}{2}$. So, query q_2 is the optimal generalization for q.

Now, let us assume that the query $q_6 = a_1 * b_1 * e_1$ is submitted to S_1 represented by Table 5. The query will fail because of the attribute e which is not listed in S_1. Clearly, we can generalize q_6 to $q_7 = a_1 * b_1 = a_1 * b_1 + a_1 * b_{[1,1]}$ and get $\{x_3, x_4\}$ as the set of objects possibly satisfying q_6. Now, assuming that systems S_1, S_2 agree on the ontology of their common attributes, we can extract definitions of e_1 from the system S. These definitions have a form of rules:

$$f_1 \longrightarrow e_1,\ a_{[1,2]} \longrightarrow e_1,\ b_1 \longrightarrow e_1,\ a_{[1,1]} * c_1 \longrightarrow e_1,\ f_2 * a_{[1,1]} \longrightarrow e_1.$$

Now we can replace query $q_6 = a_1 * b_1 * e_1$ by a new query
$q_8 = a_1 * b_1 * [f_1 + a_{[1,2]} + b_1 + f_2 * a_{[1,1]} + a_{[1,1]} * c_1] = [a_1 * b_1 * f_1] + [a_1 * b_1 * a_{[1,2]}] + [a_1 * b_1 * b_1] + [a_1 * b_1 * f_2 * a_{[1,1]}] + [a_1 * b_1 * a_{[1,1]} * c_1] = [a_1 * b_1]$.

This query transformation process shows that objects x_3, x_4 satisfy query q_6.

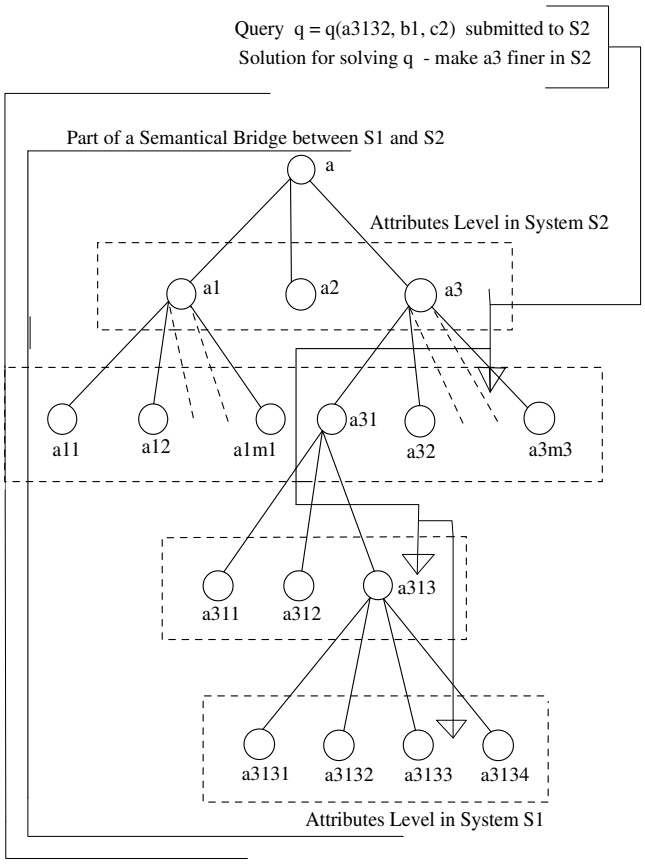

Fig. 1. Hierarchical attribute a with four levels of granularity

Let us discuss more complex scenario partially represented by Fig. 1. The set $\{a_1, a_2, a_3\}$ represents the values of a hierarchical attribute a at its first granularity level. The set $\{a_{[1,1]}, a_{[1,2]}, ..., a_{[1,m_1]}\}$ represents the values of attribute a at its second granularity level. The set $\{a_{[3,1]}, a_{[3,2]}, ..., a_{[3,m_3]}\}$ represents the remaining values of attribute a at its second granularity level. We assume here that the value a_1 can be refined to any value from $\{a_{[1,1]}, a_{[1,2]}, ..., a_{[1,n_1]}\}$. Similar assumption is made for value a_3. The set $\{a_{[3,1,1]}, a_{[3,1,2]}, a_{[3,1,3]}\}$ represents the values of attribute a at its third granularity level which are finer than the value $a_{[3,1]}$.

Finally, the set $\{a_{[3,1,3,1]}, a_{[3,1,3,2]}, a_{[3,1,3,3]}, a_{[3,1,3,4]}\}$ represents the values of attribute a at its forth granularity level which are finer than the value $a_{[3,1,3]}$.

Now, let us assume that query $q = q(a_{[3,1,3,2]}, b_1, c_2)$ is submitted to S_2. Also, we assume that attribute a is hierarchical and ordered. It basically means that the difference between the values $a_{[3,1,3,2]}$ and $a_{[3,1,3,3]}$ is smaller than between the values $a_{[3,1,3,2]}$ and $a_{[3,1,3,4]}$. Also, the difference between any two elements in $\{a_{[3,1,3,1]}, a_{[3,1,3,2]}, a_{[3,1,3,3]}, a_{[3,1,3,4]}\}$ is smaller than between $a_{[3,1,3]}$ and $a_{[3,1,2]}$.

Now, we outline a possible strategy which QAS can follow to solve q. Clearly, the best solution for answering q is to identify objects in S_2 which precisely match the query submitted by user. If this step fails, we should try to identify objects which match query $q(a_{[3,1,3]}, b_1, c_2)$. If we succeed, then we try queries $q(a_{[3,1,3,1]}, b_1, c_2)$ and $q(a_{[3,1,3,3]}, b_1, c_2)$. If we fail, then we should succeed with $q(a_{[3,1,3,4]}, b_1, c_2)$. If we fail with $q(a_{[3,1,3]}, b_1, c_2)$, then we try $q(a_{[3,1]}, b_1, c_2)$ and so on. Clearly, an alternate strategy is to follow the same steps in a reverse order. We start with a highest generalization of q which is $q(b_1, c_2)$. If we succeed in answering that query, then we try $q = q(a_{[3]}, b_1, c_2)$. If we succeed again, we try $q = q(a_{[3,1]}, b_1, c_2)$ and so on.

5 Conclusion

This paper shows how to combine cooperation and collaboration strategies in $DAIS$ to build intelligent query answering systems. These systems can often find exact answers to queries which initially fail when submitted to classical QAS. Our proposed strategy is based on query generalization which is followed by distributed knowledge discovery based query refinement.

References

1. Benjamins, V. R., Fensel, D., Prez, A. G., Knowledge management through ontologies, in *Proceedings of the 2nd International Conference on Practical Aspects of Knowledge Management (PAKM-98)*, Basel, Switzerland, 1998
2. Chu, W., Yang, H., Chiang, K., Minock, M., Chow, G., Larson, C., Cobase: A scalable and extensible cooperative information system, in *Journal of Intelligent Information Systems*, Vol. 6, No. 2/3, 1996, pp. 223-259

3. Dardzińska, A., Raś, Z.W., Rule-Based Chase Algorithm for Partially Incomplete Information Systems, in **Proceedings of the Second International Workshop on Active Mining (AM'2003)**, Maebashi City, Japan, October, 2003, pp. 42-51
4. Dardzińska, A., Raś, Z.W., On Rules Discovery from Incomplete Information Systems, in **Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining**, (Eds: T.Y. Lin, X. Hu, S. Ohsuga, C. Liau), Melbourne, Florida, IEEE Computer Society, 2003, pp. 31-35
5. Dardzińska, A., Raś, Z.W., Chasing Unknown Values in Incomplete Information Systems, in **Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining**, (Eds: T.Y. Lin, X. Hu, S. Ohsuga, C. Liau), Melbourne, Florida, IEEE Computer Society, 2003, pp. 24-30
6. Dardzińska, A., Raś, Z.W., Extracting Rules from Incomplete Decision Systems: System ERID, in **Foundations and Novel Approaches in Data Mining**, (Eds. T.Y. Lin, S. Ohsuga, C.J. Liau, X. Hu), Advances in Soft Computing, Springer, 2005, pp. 143-154
7. Fensel, D., *Ontologies: a silver bullet for knowledge management and electronic commerce*, Springer-Verlag, 1998
8. Gaasterland, T., Cooperative answering through controlled query relaxation, in *IEEE Expert*, Vol. 12, No. 5, 1997, pp. 48-59
9. Godfrey, P., Minimization in cooperative response to failing database queries, in *International Journal of Cooperative Information Systems*, Vol. 6, No. 2, 1997, pp. 95-149
10. Guarino, N., Ed., Formal Ontology in Information Systems, IOS Press, Amsterdam, 1998
11. Guarino, N., Giaretta, P., Ontologies and knowledge bases, towards a terminological clarification, in *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS Press, 1995
12. Pawlak, Z., Rough sets-theoretical aspects of reasoning about data, Kluwer, Dordrecht, 1991
13. Pawlak, Z., Information systems - theoretical foundations, in **Information Systems Journal**, Vol. 6, 1991, pp. 205-218
14. Raś, Z.W., Dictionaries in a distributed knowledge-based system, in **Concurrent Engineering: Research and Applications**, Conference Proceedings, Pittsburgh, Penn., Concurrent Technologies Corporation, 1994, pp. 383-390
15. Raś, Z.W., Dardzińska, A., Ontology Based Distributed Autonomous Knowledge Systems, in **Information Systems International Journal**, Elsevier, Vol. 29, No. 1, 2004, pp. 47-58
16. Raś, Z.W., Dardzińska, A., Query answering based on collaboration and chase, in **Proceedings of FQAS 2004 Conference**, Lyon, France, LNCS/LNAI, No. 3055, Springer-Verlag, 2004, pp. 125-136
17. Raś, Z.W., Joshi, S., Query approximate answering system for an incomplete DKBS, in **Fundamenta Informaticae Journal**, IOS Press, Vol. 30, No. 3/4, 1997, pp. 313-324
18. Sowa, J.F., Ontology, metadata, and semiotics, in B. Ganter & G. W. Mineau, eds., *Conceptual Structures: Logical, Linguistic, and Computational Issues*, LNAI, No. 1867, Springer-Verlag, 2000, pp. 55-81
19. Sowa, J.F., Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole Publishing Co., Pacific Grove, CA., 2000
20. Sowa, J.F., Ontological categories, in L. Albertazzi, ed., *Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics*, Kluwer Academic Publishers, Dordrecht, 1999, pp. 307-340.

21. Staab, S., Studer, R. (Eds), Handbook on Ontologies, International Handbooks on Information Systems, Springer, 2004
22. Suzuki E., Kodratoff Y., Discovery of Surprising Exception Rules Based on Intensity of Implication, in **Proceedings of the Second European Symposium, PKDD98**, LNAI 1510, Springer-Verlag, 1998, pp. 10-18
23. Van Heijst, G., Schreiber, A., Wielinga, B., Using explicit ontologies in KBS development, in *International Journal of Human and Computer Studies*, Vol. 46, No. 2/3, 1997, pp. 183-292