Knowledge Discovery Based Query Answering in Hierarchical Information Systems

Zbigniew W. Raś[1,2], Agnieszka Dardzińska[3], and Osman Gürdal[4]

[1] Univ. of North Carolina, Dept. of Comp. Sci., Charlotte, N.C. 28223
[2] Polish Academy of Sciences, Institute of Comp. Sci., Ordona 21, 01-237 Warsaw, Poland
[3] Bialystok Technical Univ., Dept. of Math., ul. Wiejska 45A, 15-351 Bialystok, Poland
[4] Johnson C. Smith Univ., Dept. of Comp. Sci. and Eng., Charlotte, NC 28216

Abstract. The paper concerns failing queries in incomplete Distributed Autonomous Information Systems ($DAIS$) based on attributes which are hierarchical and which semantics at different sites of $DAIS$ may differ. Query q fails in an information system S, if the empty set of objects is returned as an answer. Alternatively, query q can be converted to a new query which is solvable in S. By a refinement of q, we mean a process of replacing q by a new relaxed query, as it was proposed in [2], [7], and [8], which is similar to q and which does not fail in S. If some attributes listed in q have values finer than the values used in S, then rules discovered either locally at S or at other sites of $DAIS$ are used to assign new finer values of these attributes to objects in S. 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 a problem, we extract definitions of such attributes at some of the remote sites for S in $DAIS$ and next use them to approximate q in S. In order to do that successfully, we assume that all involved information systems have to agree on the ontology of some of their common attributes [14], [15], [16]. This paper shows that failing queries can be often handled successfully if knowledge discovery methods are used either to convert them to new queries or to find finer descriptions of objects in S.

1 Introduction

Distributed Autonomous Information System ($DAIS$) is a system that connects a number of information systems using network communication technology. Some of these systems have hierarchical attributes and information about values of attributes for some of their objects can be partially unknown. Our definition of system incompleteness differs from the classical approach by allowing 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. If we place a minimal threshold for weights to be allowed to use, we get information system of type λ. Its definition and also the definition of a distributed autonomous information system used in this paper was given by Raś and Dardzińska in [15]. Semantic inconsistencies among sites are due to different interpretations of attributes and their values among sites (for instance one site can interpret the concept *young* differently than another one). Ontologies ([1], [6], [9], [10], [17], [18], [19], [21]) can be 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 [3] is quite common before queries are answered by QAS. In [14], the notion of *rough semantics* was introduced and used to model semantic inconsistencies among sites due to different interpretations of incomplete values.

There are cases when a classical Query Answering System (QAS) fails to return an answer to a submitted query but still a satisfactory answer can be found. 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. 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. Another example of a failing query problem is when some of the attributes listed in a query are outside the domain of S. The way to approach this problem, proposed by Ras [13], is to extract definitions of such attributes at remote sites for S (if S is a part of a distributed information system) and next used them in S. This problem is very similar to the problem when the granularity of an attribute value used in a query q is finer than the granularity of the corresponding attribute used in S. By replacing such attribute values in q by more general values used in S, we retrieve objects from S which may satisfy q. Alternatively, we can compute definitions of attribute values used in q, at remote sites for S, and next use them by QAS to enhance the process of identifying objects in S satisfying q. This can be done if collaborating systems also agree on the ontology of some of their common attributes [14], [15], [16]. Additionally, the granularity level of the attribute which definition is remotely computed should be the same at the remote site and in q. This paper presents a new methodology, based on knowledge discovery, for the failing query problem.

2 Query Processing with Incomplete Data

Information about objects is collected and stored in information systems which are usually autonomous and reside at different locations. These systems are often incomplete and the same attribute may have different granularity level of its values at two different sites. For instance, at one information system, concepts *child*, *young*,*middle-aged*, *old*, *senile* can be used as values of the attribute *age*. At the other system, only integers are used as the values. If both systems agree on a semantical relationship among values of attributes belonging to these two granularity levels (their ontology), then they can use this attribute to communicate with each other. It is very likely that an attribute which is missing in one information system may occur at many others. Assume that user submits a query q to a Query Answering System (QAS) of S (called a client) and some of the attributes used in q either are not present in S or their granularity is more specific than the granularity of the same attributes at S. In both cases, S may look for a definition of each of these attributes at other information systems in $DAIS$ assuming that the granularity level of these attributes in these systems is matching their granularity level in q. All these definitions are stored in the knowledge base for S and next used to chase (see [4]) the missing values and, if needed, to refine the current values of attributes at S. Algorithm Chase for $DAIS$, based on rules, was given by Dardzińska and Raś in [5]. This algorithm can be modified easily and used for refinement of object descriptions in S.

Definition 1:
We say that $S = (X, A, V)$ is a partially 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,
- $(\forall x \in X)(\forall a \in A)[a_S(x) \in V_a \text{ or } a_S(x) = \{(v_i, p_i) : 1 \le i \le m\}]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \le i \le m\}) \to \sum_{i=1}^{m} p_i = 1]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \le i \le m\}) \to (\forall i)(p_i \ge \lambda)]$.

An example of an information system of type $\lambda = \frac{1}{4}$ is given in Table 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	$\{(e_1, \frac{1}{2}), (e_2, \frac{1}{2})\}$
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

Assume now that the set $\{S_i, i \in J\}$, where $S_i = (X_i, A_i, V_i)$, represents information systems at all sites in $DAIS$. Query language for $DAIS$ is built, in a standard way (see [16]), from values of attributes in $\bigcup\{V_i : i \in J\}$ and from the functors *or* and *and*, denoted in this paper by $+$ and $*$, correspondingly.

To be more precise, by a query language for $DAIS$ we mean the least set Q satisfying the following two conditions:

- if $v \in \bigcup\{V_i : i \in J\}$, then $v \in Q$,
- if $t_1, t_2 \in Q$, then $t_1 * t_2$, $t_1 + t_2 \in Q$.

For simplicity reason, we assume that user is only allowed to submit queries to QAS in Disjunctive Normal Form (DNF).

The semantics of queries for $DAIS$ used in this paper was proposed by Raś & Joshi in [16]. It has all the properties required for the query transformation process to be sound [see [16]]. For instance, they proved that the following distributive property holds: $t_1 * (t_2 + t_3) = (t_1 * t_2) + (t_1 * t_3)$.

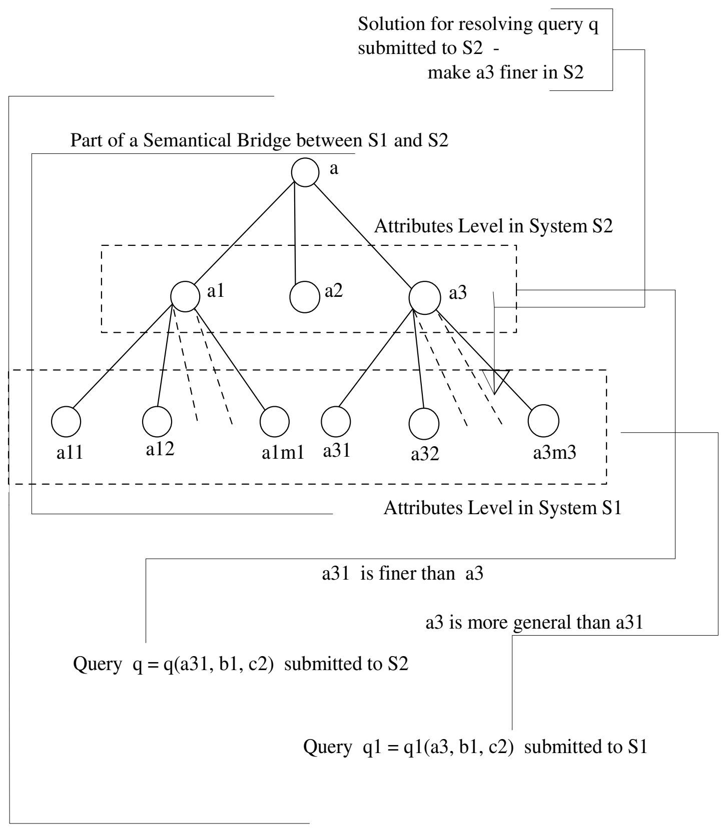

Fig. 1. Hierarchical attribute a with two levels of granularity

To recall their semantics, 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 (for predicate calculus expression) 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:

- $N_S(v) = \{(x, p) : (v, p) \in a(x)\}$, 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:

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

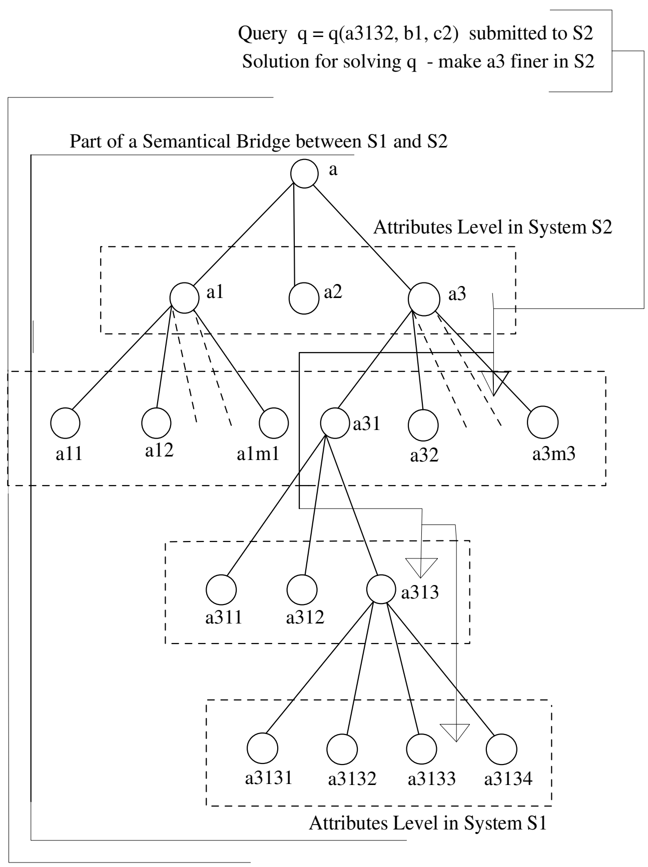

Fig. 2. Hierarchical attribute a with four levels of granularity

So, it means that the interpretation N_S is undefined for queries outside the domain V. To have such queries processed by QAS, they have to be converted to queries built only from attribute values in V.

Assume now that two information systems $S_1 = (X, A, V_1)$, $S_2 = (X, A, V_2)$ are partially incomplete and they are both of type λ. Although attributes in S_1, S_2 are the same, they may still differ in granularity of their values. Additionally, we assume that the set $\{a_{1i} : 1 \leq i \leq m\}$ contains all children of a_1 which means that semantically a_1 is equivalent to the disjunction of a_{1i}, where $1 \leq i \leq m$. Saying another words, we assume that both systems agree on the ontology related to attribute a and its values which is represented as a tree structure in Fig. 1.

Two types of queries can be submitted to S_1.

The first type is represented by query $q_1 = q_1(a_3, b_1, c_2)$ which is submitted to S_1 (see Fig. 1). The granularity level of values of attribute a used in q_1 is more general than their granularity level allowed in S_1. It means that $N_{S_1}(q_1)$ is not defined. In this case q_1 can be replaced by a new query $q_2 = q(\sum\{a_{3i} : 1 \leq i \leq m_3\}, b_1, c_2)$ which is in the domain of N_{S_1} and the same can be handled by QAS for S_1.

The second type is represented by query $q = q(a_{31}, b_1, c_2)$ which is submitted to S_2 (see Fig. 1). The granularity level of values of the attribute a used in q is finer than their granularity level allowed in S_2. It means that $N_{S_2}(q)$ is not defined. The problem now is more complex but still it can be solved. Namely, it is sufficient to learn definitions of a_{31} at other sites of $DAIS$ in terms of b_1 and c_1 or in terms of values which are finer than b_1 and c_1. When this is done, the objects in S_2 having property a_{31} can be identified by following the query processing strategy similar to the one presented in [16].

3 How to Handle Failing Queries in DAIS

In this section, the problem of failing queries in $DAIS$ is presented in a more detailed way. Namely, let us assume that a query $q(B)$ is submitted to an information system $S = (X, A, V)$, where B is the set of all 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 [14]). We assume here that two information systems can collaborate in solving q only if they agree on the ontology related to attributes used in both of them. Clearly, the same ontology does not mean that a common attribute has values of the same granularity at both sites. Similarly, as we have seen in the previous section, the granularity of values of an attribute used in a query may differ from the granularity of its values in S. In [14], 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 a 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 at one of the remote sites which accepts the ontology given below:

– age(child(≤17),
  young(18,19,...,29),
  middle-aged(30,31,...,60),
  old(61,62,...,80),
  senile(81,82,...,≥100))

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

Now, assume that the certain rule $(age, young) \longrightarrow (salary, 40K)$ is extracted at a remote site. Jointly with that rule, the following certain 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 rules. Stronger relaxation is risky since, for instance, the rule $r =[(age, N) \longrightarrow (salary, 40K)]$ may occur to be a surprising rule, as defined by Suzuki [20]. If both attributes *age* and *salary* are local in $S = (X, A, V)$ and the granularity of values of the attribute *salary* in S is more general than the granularity of values of the same attribute used in some rules listed above, then these rules can be used to convert S into a new information system which has finer information about objects in X than the information about them in S with respect to attribute *salary*. Clearly, this step will help us to solve $q(B)$ in a more precise way. Otherwise, we have to replace the user query by a more general one to match the granularity of values of its attributes with a granularity used in S. But, clearly, any user prefers to see his query unchanged.

Assume now that $D_{S'}$ is a set of all rules extracted at a remote site S' for $S = (X, A, V)$ by the algorithm $ERID(S', \lambda_1, \lambda_2)$ [4]. Parameters λ_1, λ_2 represent thresholds for minimum support and minimum confidence of these rules. Additionally, we assume that $L(D_{S'}) = \{(t \rightarrow v_c) \in D_{S'} : c \in G(A, q(B))\}$, where $G(A, q(B))$ is the set of all attributes in $q(b)$ which granularity of values in S is more general than their granularity in $q(B)$ and S'. The type of incompleteness in [15] is the same as in this paper but we also 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 $L(D_{S'})$, we can easily construct a new system $\Phi(S)$ in which $q(B)$ will fail (QAS will return either the 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 (either exact or partially incomplete) of such attributes have to be ascribed to all objects in S. Stronger the consensus among sites in $DAIS$ on a value to be ascribed to x, *finer* the result of the ascription process for x can be expected.

We may have several rules in the knowledge-base $L(D_{S'})$, associated with information system S, which describe the same value of an attribute $c \in G(A, q(B))$. 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 both of these rules, the same in a query $q(B) = v_c * t_3$ 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 identify objects in S satisfying $q(B)$. This can be done by replacing query $q(B)$ by $t_3 * (t_1 + t_2)$. Then, the resulting term is replaced by $(t_3 * t_1) + (t_3 * t_2)$ which is legal under semantics N_S. If the granularity level of values of attributes used in $t_3 * (t_1 + t_2)$ is in par with granularity of values of attributes in S, then QAS can answer $q(B)$.

Let us discuss more complex scenario partially represented in Figure 2. As we can see, attribute a is hierarchical. The set $\{a_1, a_2, a_3\}$ represents the values of attribute a at its first granularity level. The set $\{a_{[1,1]}, a_{[1,2]}, ..., a_{[1,m_1]}\}$ 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 $q(B) = q(a_{[3,1,3,2]}, b_1, c_2)$ is submitted to S_2 (see Figure 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 $q = q(B)$. 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.

But before we follow the above process, we have to discover rules at these sites of $DAIS$ which are remote for S_2 and which agree with S_2 on the ontology of attributes in $\{a, b, c\}$. These rules should describe values of any granularity of attribute a in terms of values of attributes b, c which granularity is consistent with their granularity in S_2. Clearly, if a rule $t_1 \rightarrow a_{[3,1,3,4]})$ is discovered, then also the rules $t_1 \rightarrow a_{[3,1,3])}$, $t_1 \rightarrow a_{[3,1])}$, $t_1 \rightarrow a_{[3])}$ are discovered as well.

4 Conclusion

This paper shows how to solve the failing query problem if queried information system S is a part of $DAIS$. This is done by extracting certain groups of rules in $DAIS$ and next using them by QAS to make descriptions of objects in S finer and the same way to get more precise match between them and a query.

References

1. Benjamins, V. R., Fensel, D., Prez, A. G. (1998) Knowledge management through ontologies, in *Proceedings of the 2nd International Conference on Practical Aspects of Knowledge Management (PAKM-98)*, Basel, Switzerland.
2. Chu, W., Yang, H., Chiang, K., Minock, M., Chow, G., Larson, C. (1996) Cobase: A scalable and extensible cooperative information system, in *Journal of Intelligent Information Systems*, Vol. 6, No. 2/3, 223-259
3. Dardzińska, A., Raś, Z.W. (2003) 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, 42-51
4. Dardzińska, A., Raś, Z.W. (2003) 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, 31-35
5. Dardzińska, A., Raś, Z.W. (2003) 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, 24-30
6. Fensel, D., (1998), *Ontologies: a silver bullet for knowledge management and electronic commerce*, Springer-Verlag, 1998
7. Gaasterland, T. (1997) Cooperative answering through controlled query relaxation, in *IEEE Expert*, Vol. 12, No. 5, 48-59

8. Godfrey, P. (1997) Minimization in cooperative response to failing database queries, in *International Journal of Cooperative Information Systems*, Vol. 6, No. 2, 95-149
9. Guarino, N., ed. (1998) Formal Ontology in Information Systems, IOS Press, Amsterdam
10. Guarino, N., Giaretta, P. (1995) Ontologies and knowledge bases, towards a terminological clarification, in *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS Press
11. Pawlak, Z. (1991) Rough sets-theoretical aspects of reasoning about data, Kluwer, Dordrecht
12. Pawlak, Z. (1991) Information systems - theoretical foundations, in **Information Systems Journal**, Vol. 6, 205-218
13. Raś, Z.W. (1994) Dictionaries in a distributed knowledge-based system, in **Concurrent Engineering: Research and Applications**, Conference Proceedings, Pittsburgh, Penn., Concurrent Technologies Corporation, 383-390
14. Raś, Z.W., Dardzińska, A. (2004) Ontology Based Distributed Autonomous Knowledge Systems, in **Information Systems International Journal**, Elsevier, Vol. 29, No. 1, 47-58
15. Raś, Z.W., Dardzińska, A. (2004) Query answering based on collaboration and chase, in **Proceedings of FQAS 2004 Conference**, Lyon, France, LNCS/LNAI, No. 3055, Springer-Verlag, 125-136
16. Raś, Z.W., Joshi, S. (1997) Query approximate answering system for an incomplete DKBS, in **Fundamenta Informaticae Journal**, IOS Press, Vol. 30, No. 3/4, 313-324
17. Sowa, J.F. (2000a) Ontology, metadata, and semiotics, in B. Ganter & G. W. Mineau, eds., *Conceptual Structures: Logical, Linguistic, and Computational Issues*, LNAI, No. 1867, Springer-Verlag, 55-81
18. Sowa, J.F. (2000b) Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole Publishing Co., Pacific Grove, CA.
19. Sowa, J.F. (1999a) Ontological categories, in L. Albertazzi, ed., *Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics*, Kluwer Academic Publishers, Dordrecht, 307-340.
20. Suzuki E., Kodratoff Y. (1998), Discovery of Surprising Exception Rules Based on Intensity of Implication, in **Proceedings of the Second European Symposium, PKDD98**, LNAI, Springer-Verlag
21. Van Heijst, G., Schreiber, A., Wielinga, B. (1997) Using explicit ontologies in KBS development, in *International Journal of Human and Computer Studies*, Vol. 46, No. 2/3, 183-292