Fuzzy concepts applied to the design of a database in predictive microbiology

Patrice Buche, Juliette Dibie-Barthelemy, Ollivier Haemmerlé, Rallou Thomopoulos

Fuzzy concepts applied to the design of a database in predictive microbiology

Patrice Buche [a,*], Juliette Dibie-Barthélemy [a], Ollivier Haemmerlé [b], Rallou Thomopoulos [c]

[a] *INRA, Département Mathématiques et Informatique Appliquées, Unité Mét@risk, 16 rue Claude Bernard, F-75231 Paris Cedex 5*

[b] *GRIMM-ISYCOM, Université de Toulouse le Mirail, Département de Mathématiques-Informatique, 5 allées Antonio Machado, F-31058 Toulouse Cedex*

[c] *INRA, Unité IATE, Bâtiment 31, 2 place Viala, F-34060 Montpellier Cedex 1*

{*Patrice.Buche,Juliette.Dibie*}*@inapg.fr,*
Ollivier.Haemmerle@univ-tlse2.fr,rallou@ensam.inra.fr

Abstract

This paper is dedicated to the use of fuzzy concepts in the design of a database in the field of predictive microbiology. Three characteristics of the data have guided this design: heterogeneity, incompleteness and imprecision. Three data models have been used to represent the data: the relational model, the conceptual graph model and the XML model. These models have been extended to be able to represent imprecise data as possibility distributions. They are queried simultaneously using the MIEL language. In this language, the preferences of the user are represented by fuzzy sets. Fuzzy pattern matching techniques are used to compare preferences to imprecise data. Fuzzy sets may be defined on a hierarchized domain of values, called a taxonomy (values of the domain are connected using the *a kind of* semantic link). The semantics of such a fuzzy set is precisely defined. The notion of *fuzzy set closure* is introduced to compare two fuzzy sets whose domain of values is a taxonomy.

Key words: flexible querying, fuzzy database, taxonomies, heterogeneous data, predictive microbiology

* corresponding author

1 Introduction

This article deals with three major issues in the field of databases: the integration of heterogeneous data, database incompleteness and data imprecision. They are presented in the framework of the development of a real database, in an actual application domain.

The problem of the integration of heterogeneous data is linked with the development of computer networks and databases distributed on the Web. The idea of integrating data from various electronic sources leads to the scientific field of data integration. Two kinds of architectures are usually distinguished in research involving data integration: the *data warehouses* [38] in which heterogeneous data are transformed in order to be integrated in a single global schema; and the *mediated architectures* [39] in which the data remain stored in the local databases, the mapping between the global schema and the local schemas being carried out by means of a *mediated schema*. In this article, we propose data integration based on a *mediated architecture*. More precisely, we use a global schema to integrate data expressed in three different formalisms: a relational database, a conceptual graph base and an XML base. This architecture is close to a *Global as Views* approach, in which the mediated schema is defined in terms of the local schemas to be integrated, as in the MOMIS [2] or TSIMMIS [13] systems. An original aspect of our approach is that our XML database is comparable to a data warehouse, since it contains data which have been modified in order to be expressed in the same vocabulary as the one used in the other two databases.

The problem of database incompleteness is a corollary of the *Open World Assumption*, which states that the lack of an answer to a query does not mean that the answer is negative; it is simply unknown. The Open World Assumption is often made in real applications in which it is impossible to gather all the pieces of information related to the application domain. Several ways have been proposed to make up for database incompleteness. The first one consists in allowing the user to express queries that are as large as possible in order to introduce flexibility in the query processing. For example, [31] introduces the possibility of expressing selection criteria by means of a disjunction of weighted elementary selection criteria. [3] shows that the fuzzy set theory was appropriate to represent preferences in the selection criteria. Implementations of these expressions of preferences have been proposed, such as FQUERY97 [44]. The second option consists in enlarging the query processing by generalizing the query, and then searching for relevant answers rather than exact answers only. This idea of query generalization was presented for example in [27]. The third way of palliating the incompleteness of a database consists in automatically complementing the database, for example, by means of data extracted from the Web. In this article, we propose a query language which

allows for the expression of preferences in the queries. Our language also allows for the automatic generalization of the queries, but for simplicity's sake, we do not present that issue here. Moreover, the XML base of our proposed architecture is automatically fed by means of data extracted from the Web.

The problem of data imprecision has been widely studied the last 20 over years. The *imprecision* of a piece of data is characterized by the fact that we know an information with certainty, without knowing that information precisely. The typical example of this notion of imprecision is a "vague" term, such as "young". The starting and end points of the term "young" are not precisely defined. [25] proposed one way of taking into account that imprecision by means of *OR-sets* that are disjunctions of possible values which are used instead of a single value. [30] proposed using the fuzzy set theory in order to express weights on these possible values. That is the way we chose to represent imprecision in our data.

The application domain we are interested in concerns microbiological risk assessment in food products. In the food industry, thousands of analyses (challenge tests) are processed every year in order to detect pathogenic microorganisms, thus enabling the production of safe food products. In the forthcoming years, the detection of an increasing number of pathogens may become mandatory. If the growth rate of these expenses is maintained, manufacturing any "sensitive" foodstuff would become non-profitable, and thus impossible. Predictive microbiology may provide a solution to this problem: it allows the food industry to reduce the number of scheduled challenge tests, because simulations based on modeling can replace challenge tests to a certain extent. Predictive microbiology requires a database containing experimental data (including challenge tests) and efficient modeling systems.

When designing such a database, three properties of the experimental data must be taken into account: they are heterogeneous, incomplete and imprecise. Data are heterogeneous because the information comes from different sources (bibliographical sources, industrial data, etc.) and also because knowledge about predictive microbiology, which is still a field of research, is evolving rapidly. Data are incomplete as it would be unrealistic to think that information on all possible food products and pathogens can be stored in the database. This is due to the fact that, as mentioned previously, producing experimental data is a laborious and costly task. Data may be imprecise because of the complexity of the biological processes involved: (i) several repetitions of the same experiment never produce exactly the same results; (ii) all measurement techniques have their own resolution threshold.

Figure 1 presents an overview of the architecture we propose, which integrates a relational database, a conceptual graph base and an XML base. The relational database contains the stable, well-structured part of the information: the

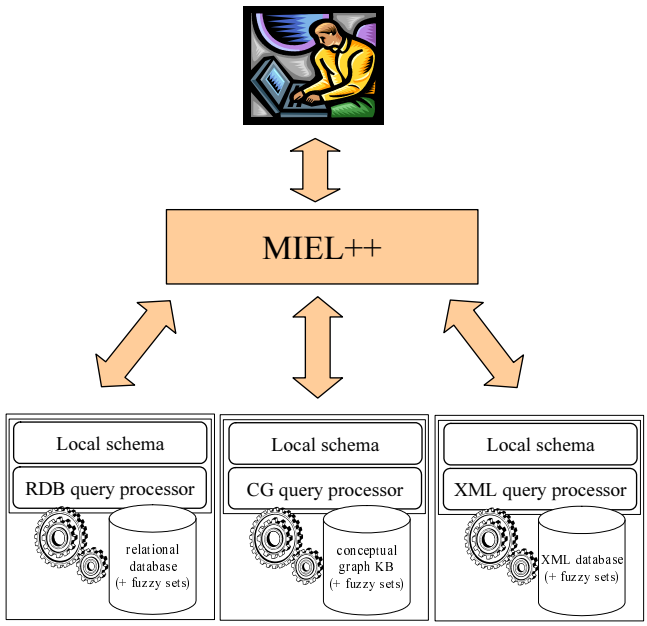


Fig. 1. The MIEL++ querying system.

relational model is widely used and very efficient RDBMS (relational database management systems) are available. The conceptual graph base contains the weakly-structured pieces of information which do not fit the relational schema. As changing a relational schema is quite an expensive operation, we decided to use an additional base in order to store information that was not expected when the schema of the database was designed, but that is useful nevertheless. We chose to use the conceptual graph model for many reasons, including (i) its graph structure, which appeared as a flexible way of representing complementary information and (ii) its readability for a non-specialist. The XML base contains information found semi-automatically on the Web by the AQWEB tool. AQWEB scans the Web, retrieves and filters documents which "look like" scientific publications. Relevant information extracted automatically from each document is stored in an XML document. XML documents are stored in an XML native database to enhance their querying.

The fuzzy set theory is used in our architecture at two levels: (i) in the data, in order to represent imprecise data; (ii) in the queries, since the unified query language we use, called MIEL, allows for the expression of preferences in the selection criteria. The data stored in the three bases of our database are expressed in different formalisms, but by means of a single domain ontology. That domain ontology consists of a taxonomy of concept names, partially ordered by the *a kind of* relation. We had to introduce a way of comparing fuzzy sets expressed on taxonomies to be able to compute the adequation of a fuzzy datum to a fuzzy query. This comparison, based on the notion of *fuzzy set*

closure, is an original aspect of our work.

This article is composed of four main sections. Section 2 introduces the MIEL query language used to query our database. Section 3 presents the three data models used in each base of our database; this presentation focuses on the representation of fuzzy values in each of these data models. Section 4 compares our approach to related works and section 5 provides information about implementation and experiments.

2 The MIEL query language

In the MIEL++ system, the query processing is done through the MIEL query language. The fuzzy sets we use in the data as well as in the queries may be defined on a continuous numeric definition domain or on a discrete hierarchized symbolic definition domain of values, which we call a *taxonomy* in this article (values of the domain are connected using the *a kind of* semantic link). The nature of the values that compose a taxonomy is specified in Section 2.3.

We first present the queries of the MIEL language, then the answer to a query and its associated relevance degree. To compute the relevance degree of an answer to a query, we have to compare fuzzy values (the one related to the query and the one related to the imprecise datum). In the last subsection, we study the comparison of fuzzy values defined on a hierarchized symbolic definition domain.

2.1 Queries in the MIEL language

Queries in the MIEL language are expressed in terms of a set of views (which is a standard concept in databases, i.e. a virtual table in which all the information needed by the user is brought together) and a set of conjunctive selection criteria of the form attribute/value. A list of retrieved attributes is expressed in each view.

Definition 1 A *query* Q is a set $\{< V_1, a_{11}, \ldots, a_{1n} >, \ldots, < V_m, a_{m1}, \ldots, a_{mn'} >, < a_1, v_1 >, \ldots, < a_l, v_l >\}$ where $V_1, \ldots, V_m$ are the names of the views in which the query is asked; a_{ij} are the projection attributes in V_i, $< a_1, v_1 >, \ldots, < a_l, v_l >$ are ordered pairs defining the selection criteria common to $V_1, \ldots, V_m$. The ordered pairs defining the selection criteria have the following meaning: $\forall i \in [1, l]$ a_i is a selection attribute common to $V_1, \ldots, V_m$ and v_i is the value associated with the selection attribute a_i. v_i is a fuzzy set on the underlying definition domain D_i of the attribute a_i. Its membership function

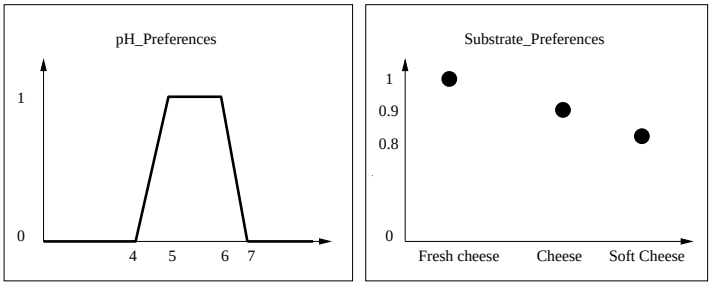


Fig. 2. Fuzzy sets pH_Preferences and Substrate_Preferences.

μ is defined by $\mu_{v_i} : D_i \rightarrow [0, 1]$ where D_i can be numeric or symbolic.

Example 1 *Here is an example of a query* $Q = \{$*<OneFactorExperiment, Substrate, Germ,* pH_{min}*,* pH_{max}*, Factor, ResponseType >, <TwoFactorsExperiment, Substrate, Germ,* pH_{min}*,* pH_{max}*, Factor1, Factor2, ResponseType>,* < **Substrate, Substrate_Preferences**>, <**pH, pH_Preferences**>$\}$. *The ordered pairs defining the selection criteria are represented in bold. The two fuzzy sets pH_Preferences and Substrate_Preferences are given in figure 2.*

As we have mentioned in the introduction (see figure 1), the MIEL++ system allows one to scan the three bases: a relational database, a conceptual graph database and an XML database (respectively noted RDB, CGDB and XMLDB in the following). All these databases are queried in a transparent way to the end-user. A *wrapper* specific to each data model is used to translate a MIEL query into a query suited for the data model considered. These wrappers are not described in this paper. The main idea is that query patterns expressed in each data model are associated with a given view in the three wrappers. Asking a MIEL query on the whole database consists in instantiating each query pattern corresponding to the view considered by specifying the selection criteria and the projection attributes. The instanciated query patterns are then matched against the data stored in the three databases. More details about the wrappers may be found in [8] for RDB, [11] for CGDB and [10] for XMLDB.

2.2 Answers in the MIEL language

In order to be able to retrieve the answer to a query from the database, the MIEL++ querying system has to: (i) compare selection attribute values specified in the query and attribute values stored in the database which may both be represented as fuzzy values, and (ii) aggregate these elementary comparisons to compute the relevance degrees associated with the answer. In our MIEL++ querying system, we chose the two scalar measures that are generally used in fuzzy set theory to evaluate the compatibility between a fuzzy selection criterion and an imprecise datum: (i) a possibility degree of match-

ing [43] and (ii) a necessity degree of matching [17]. A comparison with other approaches is presented in section 4. Moreover, as the selection criteria of a query are conjunctive, we propose to aggregate the elementary comparisons between fuzzy values by means of the *min* operator.

We can thus give the following definition of the answer to a query with its associated relevance degrees.

Definition 2 An *answer* A to a query Q is a set of elementary answers $A_{V_1}, \ldots, A_{V_m}$. Each elementary answer A_{V_i}, corresponding to the view V_i, is a set of tuples, each of them of the form $\{< a_{11}, v_{11} >, \ldots, < a_{1n}, v_{1n} > , \ldots, < a_{m1}, v_{m1} >, \ldots, < a_{mn'}, v_{mn'} >, rd_\Pi, rd_N\}$ with a_{ij} being the projection attributes in V_i, v_{ij} the associated crisp or fuzzy values resulting from the execution of the query in the database; where all the selection criteria $< a_1, v_1 >, \ldots, < a_l, v_l >$ of Q are satisfied with the possibility degrees Π_1, ..., Π_l and the necessity degrees N_1, ..., N_l; where rd_Π, rd_N is the couple of relevance degrees of a given tuple t of the answer A to the query Q defined as follows: $rd_\Pi = \min_{i=1}^{l}(\Pi_i)$ and $rd_N = \min_{i=1}^{l}(N_i)$.

Example 2 *Part of an answer corresponding to the view OneFactorExperiment associated with the query Q of example 1 is given in figure 3. Note that the attribute pH is an imprecise value stored in the database and represented as an interval in two columns: min and max.*

rd_Π	rd_N	Substrate	Germ	pH min	pH max	Factor	Response
1.0	1.0	Pasteurized fresh cheese	Bacillus cereus	5.1	5.2	Temperature	Temporal cinetic
0.9	0.9	Melted cheese	Listeria	5.0	5.4	Temperature	Level of contamination
0.8	0.8	Camembert	Listeria	6.0	6.0	Temperature	Level of contamination

Fig. 3. Part of the answer corresponding to the view *OneFactorExperiment* associated with the query Q of example 1.

2.3 Comparison of fuzzy values defined on a taxonomy

In this section, we focus on fuzzy sets defined on a hierarchized symbolic definition domain, called a taxonomy. A taxonomy represents the *a kind of* semantic relationship between values. Such a fuzzy set can be defined only on part of the taxonomy. In that case, we consider that implicit degrees are defined on the whole taxonomy. In order to take those implicit degrees into account, we define the *fuzzy set closure*. The fuzzy set closure is necessary to be able to compare the fuzzy value of a query (with a semantic of preference)

and the fuzzy value of an imprecise datum (with a semantic of a possibility distribution), when both fuzzy values are defined on a taxonomy.

We first define the semantics of a fuzzy set defined on a taxonomy, then its membership function on the whole taxonomy, called fuzzy set closure.

2.3.1 Fuzzy set defined on a taxonomy

Definition 3 A *taxonomy* Ω is defined as a set of values, partially ordered by the *a kind of* relation.

Example 3 *Figure 4 presents a part of the taxonomy defined for the values taken by the attribute Substrate in our database.*

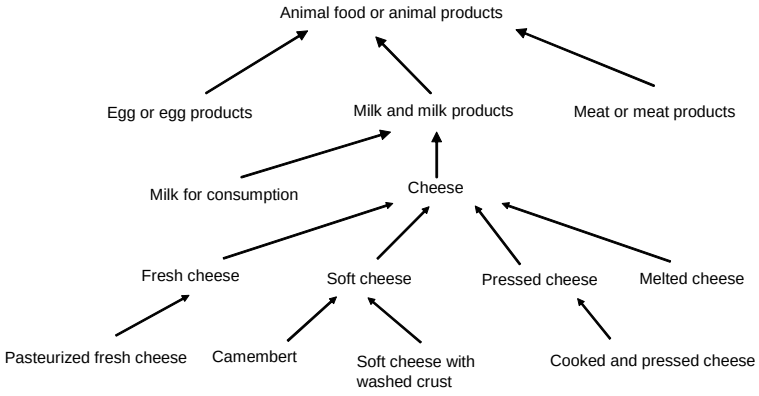


Fig. 4. Part of the taxonomy.

The values that compose a taxonomy have different interpretations in the relational, conceptual and XML models, which are specified in the corresponding sections.

Remark: "flat" symbolic domains, which are composed of unordered values are considered a particular case of taxonomy where no value is comparable to another (according to the *a kind of* relation).

Definition 4 Let f be a fuzzy set defined on a domain Ω_i which is a subset of values belonging to Ω. For all ordered pairs of values a and b belonging to $support(f)$, with b more specific than a, given μ_f, the membership function of f: $\Omega_i \rightarrow [0, 1]$, $\mu_f(a) \neq \mu_f(b)$ has the following underlying semantics:

- $\mu_f(a) > \mu_f(b)$ represents a semantics of restriction for b compared to a;
- the opposite case represents a semantics of reinforcement for b compared to a.

Let us recall that such a fuzzy set may either represent possible values in an imprecise datum or preferences in a query, which are two uses of the same homogeneous formalism [7].

Example 4 *The fuzzy set Substrate_Preferences of figure 2, also noted 1.0/Fresh Cheese + 0.9/Cheese + 0.8/Soft Cheese, is an example of fuzzy set defined on a taxonomy for the attribute Substrate. If the fuzzy set Substrate_Preferences is interpreted with a semantics of preferences, it means that the end-user is interested in cheese and implicitly in all subtypes of cheese found in the taxonomy. Moreover, the end-user is primarily interested in fresh cheese, and among the other cheeses, he/she is less interested in soft cheese. The same kind of interpretation is possible if the fuzzy set stands for an imprecise value stored in the database.*

2.3.2 Fuzzy set closure

In order to be able to compare fuzzy values defined on subsets of a taxonomy, these fuzzy values must be transformed into fuzzy sets defined on the whole taxonomy, so that the compared fuzzy sets have the same definition domain. The fuzzy set closure is computed using the following definition.

Definition 5 The *fuzzy set closure*, denoted f^*, of the fuzzy set f is defined on the entire set of values belonging to the taxonomy Ω. Given a value $a \in \Omega$, the membership function of f^* is deduced from that of f with the following rules:

- we call $E = \{a_1, a_2, \ldots, a_n\}$ the set of the smallest values belonging to $support(f)$ more general than a and not comparable [1]. If E is not empty then the membership degree associated with a is $\mu_{f^*}(a) = \max_{a_1,a_2,\ldots,a_n} \mu_f(a_i)$;
- otherwise $\mu_{f^*}(a) = 0$.

This point has been further developed in previous publications ([34,35] in particular) and the interpretation of Definition 5 with regard to the instances of concepts belonging to a taxonomy has been given in the framework of the conceptual graph model. The membership degree of a value a belonging to the taxonomy is also the degree attributed to the instances whose "minimal type" is a (see [32]), excluding the instances that simply "conform" to a. Contrary to other works in similar domains like description logics [33,37,22], here the membership of an instance to a concept is binary and not fuzzy. A fuzzy set represents the membership of concepts – i.e., values of the taxonomy – to the user's preferences (or to an imprecise datum), not the membership of instances to fuzzy concepts. Therefore the non-monotony of the degrees associated with the values of a taxonomy simply consists in associating with a set of instances

[1] Using the partial order induced by the *a kind of* relation.

– that have a given minimal type – a higher degree (reinforcement) or a lower degree (restriction) than the degree associated with instances that have a more general minimal type: the more specific concept may have characteristics that the more general concept does not have, and thus be preferred or rejected.

In the case where a value a of the taxonomy that does not appear in the original fuzzy set has several smallest, more general values that appear in the fuzzy set closure with different degrees, associating the maximum of these degrees with a in the fuzzy set closure is a choice that may be discussed. This step consists in a fusion of preferences (in a query) or knowledge (in a datum) [42,16]. Different choices are classically possible in this context. We distinguish two cases:

- if the fuzzy set expresses preferences in a query, the choice of the maximum does not allow us to exclude any possible answer. In our project, the lack of answers to a query makes this choice preferable, because it consists in enlarging the query rather than restricting it;
- if the fuzzy set represents an imprecise datum, the choice of the maximum allows us to preserve all the possible values of the datum, but it also makes the datum less specific. We chose this solution in order to homogenize the treatment of queries and data. In a way, it also contributes to enlarging the query, as a less specific datum may share more common values with the query (the possibility degree of matching can thus be higher, although the necessity degree can decrease).

Example 5 *Let us consider the taxonomy of figure 4. The fuzzy set closure of the fuzzy set Substrate_Preferences of figure 2 is given in figure 5.*

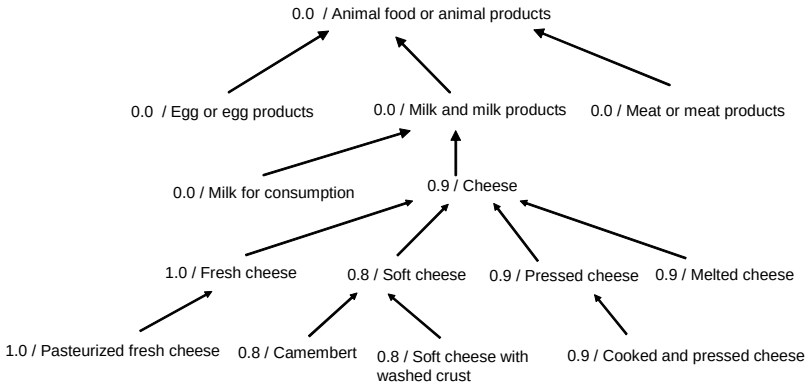


Fig. 5. The fuzzy set closure of the fuzzy set Substrate_Preferences.

3 Data model extensions to represent imprecise data

We distinguish three kinds of imprecision in the data of the database. Firstly, the level of contamination of a food substrate by a given pathogen may be known imprecisely because the resolution of the measurement technique has a given threshold. For example, in a given experiment, the level of contamination of milk by Listeria Monocytogenes is $< 10^2$ CFU/ml, 10^2 CFU/ml being the resolution threshold of the measurement technique. Secondly, when an experiment is repeated, one never obtains exactly the same result, because of biological variability. In general, experimental data are summarized by an interval of values. For example, the growth rate of Listeria Monocytogenes in skimmed milk at 15 Celsius degrees is between 3 and 5 hours. Thirdly, in the XMLDB, as documents are acquired semi-automatically on the Web, their real content is known imprecisely. More precisely, as we have mentioned in the introduction, the AQWEB software retrieves documents from the Web which "look like" scientific papers. In our current work, we have decided to extract data tables included in the document (of course, original documents are also stored in the XMLDB). This choice was made for two reasons: (i) data tables generally gather a summary of experimental data published in the document, and (ii) a table is a data structure which can be easily managed by an automatic process. Therefore, the XML documents of the XMLDB are built by a fuzzy semantic tagging of Web data tables, which maps each original value found in the table with a list of terms belonging to the MIEL taxonomy, ordered according to their relevance to the original value. Thus, a fuzzy set which represents an exclusive weighted disjunction of pertinent terms belonging to the MIEL taxonomy is associated with each original value.

In this section, we present the main choices we made to represent imprecise data as possibility distributions in the three data models we use in the database. We then study the case of the RDB, CGDB and XMLDB.

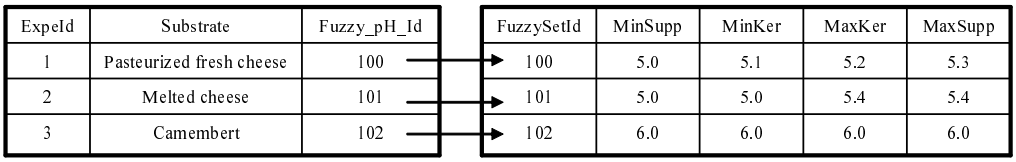
3.1 The relational model

The imprecise information represented as possibility distributions is stored in standard tables in the RDB.

A numerical imprecise datum is represented by means of a column which references the imprecise value stored in an additional table. The additional table contains, for each imprecise value, a tuple composed of a unique identifier and the 4 characteristic points of a trapezoidal fuzzy set.

Example 6 *Figure 6 gives an example of part of the Fuzzy_pH table which contains numerical fuzzy sets and part of the Substrate table which refer-*

ences the fuzzy pH values stored in the Fuzzy_pH table. The second row of the Fuzzy_pH table corresponds to an interval and the third raw, to a crisp value.



ExpeId	Substrate	Fuzzy_pH_Id
1	Pasteurized fresh cheese	100
2	Melted cheese	101
3	Camembert	102

Table Substrate

FuzzySetId	MinSupp	MinKer	MaxKer	MaxSupp
100	5.0	5.1	5.2	5.3
101	5.0	5.0	5.4	5.4
102	6.0	6.0	6.0	6.0

Table Fuzzy_pH

Fig. 6. Tables *Substrate* and *Fuzzy_pH*.

An imprecise datum defined on a taxonomic domain is also represented by means of a column which references the imprecise value stored in an additional table. The additional table contains, for each imprecise value, one or several tuple(s) composed of the unique identifier of the fuzzy set, an instance of a concept of the value domain and its associated membership degree in that fuzzy set. The taxonomic domain is stored in two tables: a table which contains all the values of the domain and a table which contains all the pairs ($value_i$, $value_j$) of the cover relation of the partial hierarchical order defined on the domain.

Example 7 *Figure 7 gives an example of part of the Imprecise_Substrate table and part of the Experiment table which references the imprecise Substrate values stored in the table Imprecise_Substrate. The first line of the Imprecise_Substrate table corresponds to a crisp value. The two other lines of the table correspond to an imprecise value defined by the fuzzy set 1.0/Melted Cheese + 0.8/Camembert (FuzzySetId=101). Moreover, figure 7 shows one part of two tables defining the associated taxonomic domain: part of the Substrate_Name reference table defining the value definition domain for the Substrate attribute and part of the Substrate_Hierarchy table defining the cover relation of the partial hierarchical order defined on the domain.*

3.2 The conceptual graph model

The conceptual graph model we use is based on the formalization presented by [28]. In the following, we first present the support which contains the terminological knowledge (the vocabulary) and the conceptual graphs which contain the factual knowledge (the data). Then, we present in an example the extension we have proposed to represent fuzzy values. A more detailed study on this subject can be found in [35] and [36].

The *support* provides the ground vocabulary used to build the knowledge base: the types of concepts, the instances of these concepts, and the types of rela-

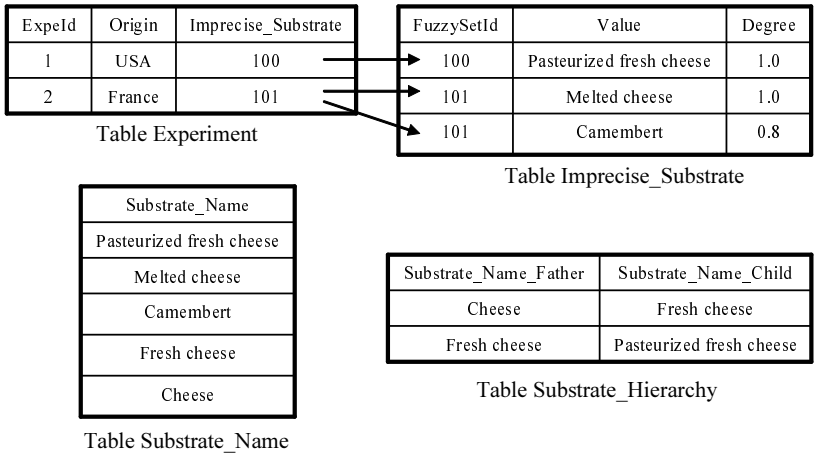


Fig. 7. Tables *Experiment*, *Imprecise_Substrate*, *Substrate_Name* and *Substrate_Hierarchy*.

tions used to link the concepts. It also defines some constraints on the use of this ground vocabulary. The concept types, which represent the taxonomic domain, are partially ordered by the *a kind of* relation. The individual markers represent the instances of the concepts. The generic marker * refers to a generic and unspecified instance of a concept. The relation types allow one to express the nature of links between concepts. The relation types are also partially ordered by the *a kind of* relation.

Example 8 *Figure 4 presents an example of a set of concept types: Fresh cheese is a kind of Cheese.*

The *conceptual graphs* built upon the support express the factual knowledge (the data). The concept vertices (noted in rectangles) represent the entities, attributes, states and events. They are labeled by two kinds of information: a concept type and a marker which can be individual or generic. The relation vertices (noted in ovals) express the nature of the relationship between concepts. We have proposed an extension of the conceptual graph model to the representation of fuzzy numerical and symbolic values which only concerns concept vertices. A fuzzy set can appear in two ways in a concept vertex: (i) as a fuzzy marker to represent a numerical imprecise value; or (ii) as a fuzzy type defined on the concept type set to represent an imprecise value defined on a hierarchized definition domain.

Example 9 *The fuzzy conceptual graph of figure 8 is composed of four concept vertices: the generic concept vertex pH, the individual concept vertex Experiment where E1 is an instance of Experiment, the concept vertex Numerical-Value with a fuzzy marker which represents a numerical imprecise value of pH and the concept vertex with a fuzzy type which represents an imprecisely known substrate defined by the fuzzy set 1.0/Fresh Cheese + 0.5/Cheese. The fuzzy*

conceptual graph is also composed of three relation vertices. For example, the binary relation type HasForValue allows one to link a pH with a numerical value.

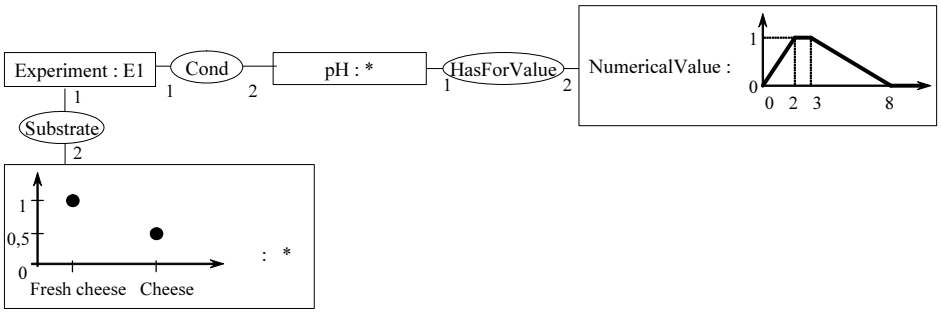


Fig. 8. A fuzzy conceptual graph.

A pair *ct:m* is composed of a concept type *ct* (possibly fuzzy) and an instance – or marker – *m* (possibly fuzzy) that conforms to this type. Representing numerical values in the conceptual graph model implies the use of a dedicated concept type (*NumericalValue*), due to the unicity of the minimal concept type associated with each individual marker (see [36] for more details).

3.3 The XML model

We have extended the tree-based model proposed in the Xyleme Project [1,41] to represent the XML database. A more detailed study on this subject can be found in [10]. The taxonomy is stored in an XML tree in which each concept is represented by a given XML element. An XML database is a set of fuzzy data trees, each of them representing an XML document. A fuzzy data tree is a data tree that allows one to represent fuzzy values. According to definition of [1,41], a data tree is a triple (t, l, v) where (t, l) is a labeled tree and v is a partial value function that assigns values to nodes of t. In a fuzzy data tree, the function v can assign a crisp or a fuzzy value to a node. A fuzzy value is represented by a data tree depending on its type:

- a numerical imprecise value is represented by a data tree that is composed of a root labeled *CFS* (for Continuous Fuzzy Set) and four leaves labeled *minSup*, *minKer*, *maxKer*, *maxSup* of respective values min(support(f)), min(kernel(f)), max(kernel(f)) and max(support(f));
- an imprecise value defined on a hierarchized symbolic definition domain is represented by a data tree that is composed of a root labeled *DFS* (for Discrete Fuzzy Set) and such that for each instance of a concept of the definition domain, there exists a node labeled *ValF* that has two children labeled *Item* and *MD* (for Membership Degree).

Example 10 *Figure 9 provides an example of a fuzzy data tree corresponding to part of the XML document obtained from the fuzzy tagging of a Web data table. The node* originalVal *has a crisp value* Red onion*. It is an example of a value present in one of the cells of the Web data table. The node* finalVal *is a fuzzy value. It is composed of three elements:* Tree onion*,* Welsh onion *and* Red cabbage*, which are the three nearest terms of the taxonomy compared to the original value* Red onion*. The value associated with each term of the taxonomy represent the possibility degree that this term is the best match of the original value (see [9] for more explanations about the calculus of this degree).*

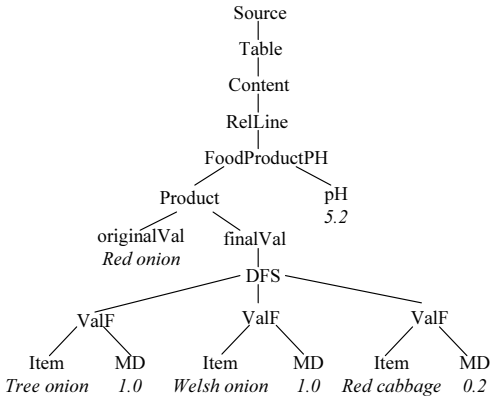


Fig. 9. A fuzzy data tree.

4 Related works

As already mentioned in the introduction, we use a mediator approach in order to integrate three heterogeneous databases, their data being integrated at query time [39]. Two of our databases store data preserved in their original shape (in the RDB and CGDB subsystems), the third one follows a data warehouse approach since it is an XML database in which the data extracted from the Web are transformed in order to fit the schema [23]. In each subsystem, we propose extending the data model in order to represent imprecise data. Data integration is performed using MIEL query processing during which fuzzy sets expressing user's preferences are enlarged using the taxonomy and then compared to imprecise data. Therefore, our approach must be related to two kinds of research: fuzzy database models and fuzzy querying systems.

Fuzzy database models: The fuzzy set framework has been proposed in order to represent imprecise values by means of possibility distributions [43]. Several authors have developed this approach in the context of databases [29], especially for the relational database model [4] and object-oriented database model [24]. In our RDB subsystem, imprecise data representation is very similar to that proposed in [19]. The main originality of our approach lies on the

representation of imprecise data in the CGDB subsystem and fuzzy semantic annotations in the XMLDB. Research has been proposed to introduce fuzziness in the conceptual graph model, [26,40,12]. We presented in [36,35] the reasons why we proposed a new definition of fuzziness in conceptual graphs. Our main criticism of these works is the absence of non-ambiguous semantics. We have preferred to introduce a less expressive but non-ambiguous formalism: the fuzzy sets are only used in the concept vertices, and they express possibility distributions in the data or expression of preferences in the queries.

Fuzzy querying systems: Our approach can be compared to two categories of research proposed in the literature in order to introduce enlargement in the querying: fuzzy information retrieval systems and fuzzy database querying systems. In fuzzy information retrieval systems, a query composed of a set of terms is enlarged to similar terms using fuzzy pseudo-thesauri [14]. Similarity is based on the co-occurrence frequency of terms in a given set of documents. We cannot use this approach in our context, as our taxonomy of terms is not computed from a set of documents but given by experts, and the relations between terms in the taxonomy are not fuzzy. The fuzzy set framework has also been shown to be a sound scientific choice for modeling flexible database queries [3,5]. It is a natural way of representing the notion of preference using a gradual scale. We can distinguish at least two types of approaches to compare user's preferences to an imprecise datum retrieved from a database, both represented as a fuzzy set. In the first one [6,21,20], a measure is proposed to evaluate how close to each other the shapes of two fuzzy sets are, taking into account similarity relations. In the second one, a measure is used to evaluate the possibility and necessity degrees of equality between a fuzzy value and a fuzzy requirement [15,30]. [18] proposed an extension of fuzzy pattern matching which integrates similarity relations, but it does not take into account the case of hierarchically organized domains, like taxonomies. For instance, terms may be added to the support of the fuzzy set in the enlargement mechanism, but more specific terms than these may stay outside of it, which is a major drawback for taxonomic domains. As we wished to evaluate the extent to which an imprecise datum corresponds to the user's requirement, our approach remains close to [18]. The originality of our approach is computing an automatic enlargement of user's preferences, not via a similarity relation introduced in the fuzzy pattern matching, but rather by using the fuzzy set closure presented in section 2.3.2.

5 Implementation and experiments

The MIEL++ system has been fully implemented in the Java language. A MIEL++ query is executed using a three tier process architecture. This architecture has been designed to minimise the data transfers between the user

desktop and the MIEL++ server: (i) the MIEL++ Java client running under a usual Web browser implements the graphical user interface; (ii) the MIEL++ Java server process running under a RMI (Remote Method Invocation) server implements all the calculus computed by the engine (query completion using the notion of closure, database scanning, fuzzy pattern matching calculus), and (iii) the servers of the three databases. In the current version, the RDB subsystem has been transferred to one of our partners who is in charge of its commercialization (see http://www.symprevius.org/). The RDB contains about 10,000 experimental data. The CGDB and XMLDB subsystems are still prototypes. The CGDB contains about 150 data graphs and the XMLDB, about 500 data trees. We have tested the efficiency of the fuzzy set closure operation presented in this paper on the RDB. In this experimentation, we have defined 7 queries with our partners, who are experts in microbiology. The MIEL processing of those 7 queries retrieved 497 data from the RDB. Of those data, only 5 were retrieved using the fuzzy set; the 492 remaining data were retrieved using the fuzzy set closure. Those data are indeed all relevant since, according to definition 5, they correspond to more specific values than those expressed in the fuzzy set. This result shows that this operation is essential.

6 Conclusion

In this paper we have proposed extending three data models to represent imprecise data. As we have mentioned in section 4, the extensions we have proposed for the CGDB and the XMLDB represent one contribution of our approach. In the field of microbiology, taxonomies are used to classify data (for example by food products or micro-organisms). These taxonomies have been incorporated in the database to index the data. They are used by the MIEL system to enlarge the querying. To do that, we have defined the concept of fuzzy set whose domain of value is a taxonomy and we have proposed a way of computing the fuzzy set closure which allows for: (i) enlarging the querying to more specific values found in the taxonomy, and (ii) comparing preferences expressed in the query to imprecise data stored in the database using the same definition domain. To the best of our knowledge, this has never been done before; it is the second contribution of our approach. Another aspect of our current research, as examined in this paper, involves the introduction of viewpoints in the taxonomies considered. An important point will also be to extend our results, in a meaningful way, to other sorts of relations.

References

[1] V. Aguiléra, S. Cluet, P. Vetri, D. Vodislav, and F. Wattez. Querying the XML documents on the web. In *Proceedings of the ACMSIGIR Workshop on XML and I.R.*, Athens, July 2000.

[2] D. Beneventano, S. Bergamaschi, S. Castano, A. Corni, R. Guidetti, G. Malvezzi, M. Melchiori, and M. Vincini. Information integration: The MOMIS project demonstration. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 611–614, Cairo, Egypt, September 2000. Morgan Kaufman Publishers.

[3] P. Bosc, L. Lietard, and O. Pivert. Soft querying, a new feature for database management system. In *Proceedings DEXA'94 (Database and EXpert system Application), Lecture Notes in Computer Science #856*, pages 631–640. Springer-Verlag, 1994.

[4] P. Bosc, L. Lietard, and O. Pivert. *Fuzziness in Database Management Systems*, chapter Fuzzy theory techniques and applications in data-base management systems, pages 666–671. Academic Press, 1999.

[5] P. Bosc and O. Pivert. SQLf: a relational database language for fuzzy querying. *IEEE Transactions on fuzzy systems*, 3(1):1–17, February 1995.

[6] P. Bosc and O. Pivert. On representation-based querying of databases containing ill-known values. In *Proceedings of ISMIS'97*, pages 477–486, 1997.

[7] P. Bosc and H. Prade. An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain or imprecise databases. In A. Motro and P. Smets, editors, *Uncertainty and Management in Information Systems: From Needs to Solutions*, pages 285–324. Kluwer Academic Publishers, 1997.

[8] P. Buche, C. Dervin, O. Haemmerlé, and R. Thomopoulos. Fuzzy querying of incomplete, imprecise and heterogeneously structured data in the relational model using ontologies and rules. *IEEE Transactions on Fuzzy Systems*, 13(3):373–383, June 2005.

[9] P. Buche, J. Dibie-Barthélemy, O. Haemmerlé, and G. Hignette. Fuzzy semantic tagging and flexible querying of XML documents extracted from the web. *to appear in the Journal of Intelligent Information Systems*, 2005.

[10] P. Buche, J. Dibie-Barthélemy, O. Haemmerlé, and M. Houhou. Towards flexible querying of XML imprecise data in a dataware house opened on the web. In *Proceedings of the 6th International Conference Flexible Querying Answering Systems, FQAS 2004*, pages 28–40, Lyon, France, June 2004. Lecture Notes in AI #3055, Springer.

[11] P. Buche, O. Haemmerlé, and R. Thomopoulos. Integration of heterogeneous, imprecise and incomplete data: an application to the microbiological risk

assessment. In *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems, ISMIS2003*, pages 98–107, Maebashi, Japan, October 2003. Lecture Notes in AI #2871, Springer.

[12] T.H. Cao. *Foundations of Order-Sorted Fuzzy Set Logic Programming in Predicate Logic and Conceptual Graphs.* PhD thesis, University of Queensland, Australia, 1999.

[13] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papaconstantinou, J.D. Ullman, and J. Widom. The TSIMMIS project: integration of heterogeneous information sources. In *Proceedings of the 16th Meeting of the Information Processing Society of Japan*, pages 7–18, Tokyo, Japan, 1994.

[14] M. De Cock, S. Guadarrama, and M. Nikravesh. Fuzzy thesauri for and from the www. *In Nikravesh, M., Zadeh, L., Kacprzyk, J., eds.: Soft Computing for Information Processing and Analysis*, pages 275–284, 2004.

[15] J.C. Cubero and M.A. Vila. A new definition of functional dependency in fuzzy relational databases. *Journal of Intelligent Systems*, 9:441–448, February 1994.

[16] D. Dubois and H. Prade. A review of fuzzy sets aggregation connectives. *Information Sciences*, (36):85–121, 1985.

[17] D. Dubois and H. Prade. *Possibility Theory - An Approach to Computerized Processing of Uncertainty.* Plenum Press, New York, 1988.

[18] D. Dubois and H. Prade. *Fuzziness in Database Management Systems, P. Bosc and J. Kacprzyk eds.*, chapter Tolerant fuzzy pattern matching : an introduction, pages 42–58. Heidelberg: Physica Verlag, 1995.

[19] J. Galindo, J.C. Cubero, O. Pons, and J.M. Medina. A server for fuzzy SQL queries. In *Proceedings of the 1998 workshop FQAS'98 (Flexible Query-Answering Systems)*, pages 161–171, Roskilde, Denmark, May 1998. Springer-Verlag.

[20] R. George, R. Srikanth, B. P. Buckles, and F.E. Petry. *An approach to modelling impreciseness and uncertainty in the object-oriented data model*, pages 325–337. John Wiley and Sons, Inc, 1997.

[21] R. George, A. Yazici, B. P. Buckles, and F.E. Petry. *Modeling Impreciseness and Uncertainty in the Object-Oriented Data Model - A Similarity-Based Approach*, pages 63–95. Advances in Fuzzy systems- Applications and Theory, Vol. 13, World scientific, 1997.

[22] B. Hollunder. An Alternative Proof Method for Possibilistic Logic and its Application to Terminological Logics. In Ramon Lopez de Mantaras and David Poole, editors, *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 327–335, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers.

[23] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis. *Fundamentals of Data Warehouses.* Springer Verlag, 2000.

[24] J. Lee, J. Kuo, and N. Xue. A note on current approaches to extending fuzzy logic to object-oriented modeling. *International Journal of Intelligent Systems*, 16(7):807–820, 2001.

[25] W. Lipski. On databases with incomplete information. *J. ACM*, 28(1):41–70, 1981.

[26] S.K. Morton. *Conceptual graphs and fuzziness in artificial intelligence*. PhD thesis, University of Bristol, 1987.

[27] A. Motro. *Query Generalization: a Method for Interpreting Null Answers*, pages 597–616. Benjamin/Cummings Publishing Company, 1986.

[28] M.L. Mugnier and M. Chein. Représenter des connaissances et raisonner avec des graphes. *Revue d'Intelligence Artificielle*, 10(1):7–56, 1996.

[29] H. Prade. Lipski's approach to incomplete information data bases restated and generalized in the setting of Zadeh's possibility theory. *Information Systems*, 9(1):27–42, 1984.

[30] H. Prade and C. Testemale. Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences*, 34:115–143, 1984.

[31] F. Rabitti and P. Savino. Retrieval of multimedia documents by imprecise query specification. In F. Bancilhon, C. Thanos, and D. Tsichritzis, editors, *Advances in Database Technology - EDBT'90. International Conference on Extending Database Technology*, volume 416 of *Lecture Notes in Computer Science*, pages 203–218, Venice, Italy, March 1990. Springer.

[32] J.F. Sowa. *Conceptual structures - Information processing in Mind and Machine*. Addison-Welsey, 1984.

[33] U. Straccia. A Fuzzy Description Logic. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 594–599. AAAI Press, 1998.

[34] R. Thomopoulos. *Représentation et interrogation élargie de données imprécises et faiblement structurées*. PhD thesis, Institut National Agronomique Paris-Grignon, 2003.

[35] R. Thomopoulos, P. Buche, and O. Haemmerlé. Different kinds of comparisons between fuzzy conceptual graphs. In *Proceedings of the 11th International Conference on Conceptual Structures, Lecture Notes in Artificial Intelligence 2746*, pages 54–68, Dresden, Germany, 2003. Springer.

[36] R. Thomopoulos, P. Buche, and O. Haemmerlé. Representation of weakly structured imprecise data for fuzzy querying. *Fuzzy sets and System*, 140:111–128, 2003.

[37] C.B. Tresp and R. Molitor. A Description Logic for Vague Knowledge. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI'98)*, pages 361–365. J. Wiley and Sons, 1998.

[38] J. Widom. Research problems in data warehousing. In *Proceedings of the International Conference on Information and Knowledge Management*, 1995.

[39] G. Wiederhold. Mediation in information systems. *ACM Computing Surveys*, 27(2):265–267, june 1995.

[40] V. Wuwongse and M. Manzano. Fuzzy conceptual graphs. In *Proceedings of the First International Conference on Conceptual Structures, Lecture Notes in Artificial Intelligence #699*, pages 430–449, Quebec City, Canada, August 1993. Springer-Verlag.

[41] Lucie Xyleme. A dynamic warehouse for xml data of the web. *IEEE Data Engineering Bulletin*, 2001.

[42] R.R. Yager. Connectives and quantifiers in fuzzy sets. *Fuzzy Sets and Systems*, 40:39–75, 1991.

[43] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[44] S. Zadrozny and J. Kacprzyk. Implementing fuzzy querying via the internet/WWW: Java applets, activeX controls and cookies. In *Proceedings of the 1998 workshop FQAS'98 (Flexible Query-Answering Systems)*, pages 358–369, Roskilde, Denmark, May 1998. Springer-Verlag.