Integration of heterogeneous, imprecise and incomplete data: an application to the microbiological risk assessment

Patrice Buche[1], Ollivier Haemmerlé[1,2], and Rallou Thomopoulos[1]

[1]UMR INA P-G/INRA BIA, 16 rue Claude Bernard, F-75231 Paris Cedex 05, France
[2]LRI (UMR CNRS 8623 - Université Paris-Sud) / INRIA (Futurs), Bâtiment 490, F-91405 Orsay Cedex, France
{Patrice.Buche,Ollivier.Haemmerle,Rallou.Thomopoulos}@inapg.fr

Abstract. This paper presents an information system developed to help the assessment of the microbiological risk in food products. UQS (Unified Querying System) is composed of two distinct bases (a relational database and a conceptual graph knowledge base) which are integrated by means of a uniform querying language. The specificity of the system is that both bases include fuzzy data. Moreover, UQS allows the expression of preferences into the queries, by means of the fuzzy set theory.

1 Introduction

Following several food safety problems, the Marrakech Agreement was signed in 1994 during the creation of the World Trade Organization. Included in this agreement, the SPS Agreement (Sanitary and Phytosanitary measures) concerns the international trade of food, and targets the safety and protection of human health. One important principle of the SPS Agreement is the study of risk analysis. Our research project is part of a French programme that aims at building a tool for microbiological risk analysis in food products. This tool is based on a relational database containing data extracted from scientific publications in microbiology. As changing the schema of the database 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 is useful nevertheless. We chose to use the conceptual graph model [1] for many reasons: (i) its graph structure which appeared as a flexible way of representing complementary information; (ii) its readability for a non-specialist; (iii) its interpretation in first order logic (FOL) which provides a robust theoretical framework; (iv) the availability of a development platform providing efficient algorithms; (v) the distinction between the terminological part and the assertional part of the knowledge (as in Description Logics, for example).

In UQS (Unified Querying System), both bases are queried simultaneously by a unified querying mechanism. Both have to deal with the following two specificities. Firstly, some of the data are imprecise, like data whose precision is limited by the measuring techniques. For instance by using a method allowing one to detect bacteria beyond a given concentration threshold (e.g. 10^2 cells per gramme), not detecting any bacterium means that their concentration is below this threshold, which is an imprecise value noted "$< 10^2$ cells/g". Secondly, the bases are incomplete, as they will never contain information about all possible food products and all possible pathogenic germs. Those two characteristics led us to propose, firstly the handling of imprecise values, and secondly the expression of different levels of preferences in the user's selection criteria so as to allow flexible querying. In the bibliography concerning databases, the fuzzy set framework has been shown to be a sound scientific way of modelling both flexible queries [2] and imprecise values by means of possibility distributions [3].

In this paper, we remind briefly the Conceptual Graph model and the fuzzy set theory in section 2. In section 3, we present our query language which allows the expression of preferences. Then both relational database system and Conceptual Graph knowledge base system are presented in section 4 and 5.

2 Preliminary notions

2.1 The Conceptual Graph model

The Conceptual Graph model (or CG) [1] is a knowledge representation model based on labelled graphs. We use the formalization presented in [4]. In the following, we present the *support* which contains the terminological knowledge, the *conceptual graphs* which contain the assertional knowledge, and the *specialization relation* on CGs.

The support The support provides the ground vocabulary used to build the knowledge base: the types of concepts used, the instances of these types, and the types of relations linking the concepts.

The *set of concept types* is partially ordered by a *kind of* relation. *Universal* and *Absurd* are respectively its greatest and lowest elements. Fig. 1 presents a part of the set of concept types used in the application.

The concepts can be linked by means of relations. The support contains the *set of relation types*. Each relation type has a given arity. An example of relation is *agent* which is a binary relation allowing one to link an *Action* with a *Germ* (which are both concept types).

The third set of the support is the *set of individual markers*, which represent the instances of the concepts. For example, *Celsius degree* can be an instance of *Degree*. The generic marker (noted $*$) is a particular marker referring to an unspecified instance of a concept.

The conceptual graphs The CGs, built upon the support, express the factual knowledge. A CG is composed of: (i) a set of *concept vertices* (noted in rectangles) which represent the entities, attributes, states, events; (ii) a set of *relation vertices* (noted in ovals) which express the nature of the relationship between concepts; (iii) a set of edges linking relation vertices to concept vertices; (iv) a

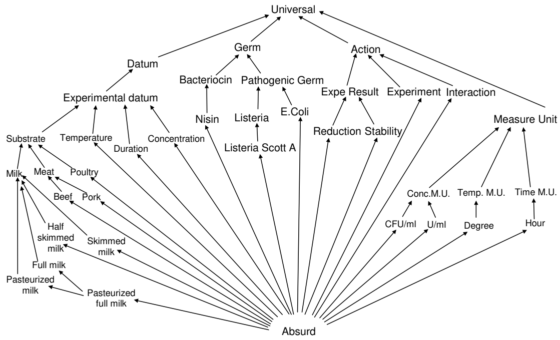

Fig. 1. A part of the concept type set for the microbial application

label for each vertex or edge: the label of a concept vertex is a pair composed of a concept type and a marker (individual or generic) of this type, the label of a relation vertex is its relation type, the label of an edge is its rank in the total order on the neighbours of a given relation vertex.

For example, the CG given in Fig. 2 is a representation of the following information: "the experiment E1 carries out an interaction I1 between Nisin and Listeria Scott A in skimmed milk and the result is reduction".

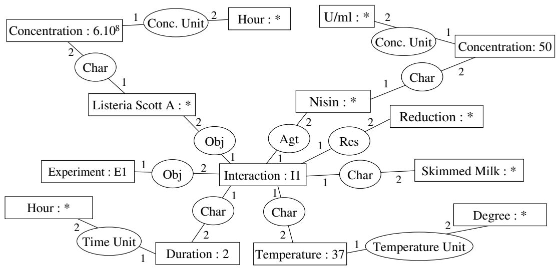

Fig. 2. An example of a Conceptual Graph

Specialization relation, projection operation The set of CGs is partially pre-ordered by the *specialization relation* (noted $\leq$), which can be computed by the *projection operation* (a graph morphism allowing a restriction of the vertex labels): $G' \leq G$ if and only if there is a projection of G into G'.

Definition 1 *A* **projection** Π *from a CG* G *into a CG* G' *is a pair* (f, g) *of mappings,* f *(resp.* g*) from the set of relation vertices (resp. concept vertices) of* G *to the set of relation vertices (resp. concept vertices) of* G' *such that: (i) the edges and their labels are preserved; (ii) the vertex labels may be restricted.*

An example is given in Fig. 3. The projection is a ground operation in the CG model since it allows the search for answers, which can be viewed as specializations of a query.

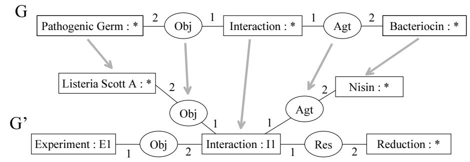

Fig. 3. There is a projection from G into G', $G' \leq G$ (G' is a specialization of G)

2.2 The fuzzy set theory

In this paper, we use the representation of fuzzy sets proposed in [5, 6].

Definition 2 *A* **fuzzy set** *A on a domain X is defined by a membership function μ_A from X to [0, 1] that associates the degree to which x belongs to A with each element x of X.*

A fuzzy set can be defined on a continuous or on a discrete domain, as illustrated in Fig. 4.

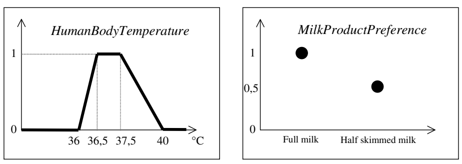

Fig. 4. Two fuzzy sets defined respectively on a continuous and on a discrete domain

The fuzzy set formalism can be used in two different ways:

- in the queries, in order to express preferences on the domain of a selection criterion. For example the fuzzy set *HumanBodyTemperature* in Fig. 4 may be interpreted as a preference on the required value of the criterion *Temperature*: a value between 36.5 and 37.5 degrees is fully satisfactory; values outside this interval may also be acceptable with smaller preference degrees;
- in the data, in order to represent imprecise data expressed in terms of possibility distributions. For example the fuzzy set *MilkProductPreference* in Fig. 4 may be interpreted as an imprecise datum if the kind of milk that was used in the experiment is not clearly known: it is very likely to be full milk, but half-skimmed milk is not excluded.

3 The query language

This section presents the query language used in both the Relational Database (RDB) and the Conceptual Graph Knowledge Base (CGKB). In the following, we present the notions we use in terms of domain relational calculus [7].

3.1 The views

In our system, the notion of view is central, since it is used in the RDB and in the CGKB. A view is a usual notion in databases, e.g. a virtual table in which all the information needed by the user is brought together. In UQS, the set of views can be compared to a mediated schema. We define a view by a set of attribute names which are consultable (they can be used as projection attributes or as selection attributes), and by a logical predicate which defines the way the view is computed. The nature of such a predicate will be precised in sections 4 and 5.

Definition 3 *A* **view** *V on $n (n > 0)$ attributes $a_1, \ldots, a_n$ is defined by $V = \{a_1, \ldots, a_n | \exists b_1, \ldots, b_m P_V(a_1, \ldots, a_n, b_1, \ldots, b_m)\}$ where P_V is a predicate which characterizes the construction of the view, $b_1, \ldots, b_m$ being the attributes belonging to the definition of the view without being consultable*[1]*.*

Example 1 BacteriocinInteraction $= \{PathogenicGerm,\ Bacteriocin,\ ExpeResult,\ Substrate,\ Duration,\ Temperature \mid P_{BacteriocinInteraction}(PathogenicGerm,\ Bacteriocin,\ ExpeResult,\ Substrate,\ Duration,\ Temperature)\}$. *That view concerns the interaction of bacteriocins (which are kinds of bacteria) on pathogenic germs.*

3.2 The queries

A query in UQS is always asked on a given view, by precising a set of projection attributes and a set of selection criteria using the form <attribute/value>.

Definition 4 *A* **query** *Q asked on a view V defined on n attributes $\{a_1, \ldots, a_n\}$ is defined by $Q = \{a_1, \ldots, a_l | \exists a_{l+1}, \ldots, a_n (P_V(a_1, \ldots, a_n) \wedge (a_{l+1} = v_{l+1}) \wedge \ldots \wedge (a_m = v_m))\}$ $1 \leq l \leq m \leq n$, where P_V is the predicate which characterizes the construction of the view V, $a_1, \ldots, a_l$ are the projection attributes, $a_{l+1}, \ldots, a_m$ are the selection attributes with their respective values $v_{l+1}, \ldots, v_m$ (the attributes $a_{m+1}, \ldots, a_n$ are not used in that query). The values v_i are given as selection values by the user. They can be precise values as well as a fuzzy sets. In the case of a fuzzy set, such a value is interpreted as an expression of preferences.*

A query is a partial instanciation of a given view by specifying the projection attributes and by giving selection values to some other attributes.

Example 2 $Q = \{PathogenicGerm,\ ExpeResult \mid \exists Substrate,\ Duration\ (P_{BacteriocinInteraction}(PathogenicGerm,\ Bacteriocin,\ ExpeResult,\ Substrate,\ Duration, Temperature) \wedge (Temperature = HumanBodyTemperature) \wedge (Bacteriocin =' Nisin'))\}$

The query Q expresses that we want to obtain the $PathogenicGerm$ and the $ExpeResult$ from the view $BacteriocinInteraction$ when the $Temperature$ is a $HumanBodyTemperature$ (see Fig. 4) and the $Bacteriocin$ is $Nisin$.

[1] For readability reasons, we do not mention the attributes $b_1, \ldots, b_m$ in the following definitions and examples

Definition 5 *An* **answer** *A to a query Q in UQS is a set of tuples, each of the form $\{v_1, \ldots, v_l, \delta\}$, $v_1, \ldots, v_l$ corresponding to the values (which can be fuzzy values) associated with each projection attribute $a_1, \ldots, a_l$ of Q, δ being the degree of adequation of the answer to the query, presented in [8].*

The query processing in UQS is the following: when a query is asked, the system searches for the considered view both in the RDB and in the CGKB. Then the RDB engine and/or the CGKB engine are run in parallel, each subsystem building partial answers to the query. The global answer results from the merging of the partial answers. Note that all the views of the system need not exist in both RDB and CGKB parts of the system.

In the two following sections we present the RDB subsystem very briefly, then the CGKB subsystem in more details. We will explain the link between a query in UQS and its translation, by means of "wrappers", into a query designed for the RDB and/or into a query designed for the CGKB.

4 The RDB subsystem

4.1 Presentation of the subsystem

The first subsystem is composed of an RDB implemented in Oracle. It is composed of about 90 tables, which contain data extracted from about 500 scientific publications in microbiology. A preliminary version of this subsystem has been presented in [9]. An extended version of this work is under submission.

4.2 The RDB wrapper

The access to the RDB is done by means of views, which consist in pre-written SQL queries.

Definition 6 *$\mathcal{V}_{db} = \{Vr_1, \ldots, Vr_n\}$ is the set of views on the relational database, with $P_{Vr_1}, \ldots, P_{Vr_n}$ the predicates characterizing each view.*

Remark 1 *The predicate P_{Vr_i} corresponding to a view Vr_i is the translation in terms of relational calculus of the SQL query which defines the view.*

Thus the querying mechanism consists in specifying the values of the selection attributes, then in asking that complemented SQL query on the RDB.

5 The CG subsystem

5.1 Extension of the CG model to the representation of fuzzy values

The second subsystem is composed of a knowledge base expressed in terms of CGs. In order to allow the storage of data with the same expressivity as the data stored in the RDB, we chose to extend the CG model to the representation of numerical and fuzzy values in concept vertices. This work is presented in details in [8]. We only recap it through examples.

A fuzzy set can appear in two ways in a concept vertex (Fig. 5): (i) as a marker: this fuzzy set can be continuous or discrete; (ii) as a fuzzy type defined on a subset of the set of concept types. In the extension of the model to the representation of fuzzy sets, we have extended the projection operation in order to take into account the concept vertices with a fuzzy type or a fuzzy marker. Intuitively, the notion of specialization for fuzzy sets is based on the inclusion relation: if A and B are fuzzy sets, A is a specialization of B if and only if A is included in B.

Fig. 5 presents an example of a projection involving fuzzy markers. There is a projection because, in addition to the usual projection criteria, the fuzzy marker in the second CG is more specific than the fuzzy marker in the first CG (its characteristic function is lower on the whole definition domain).

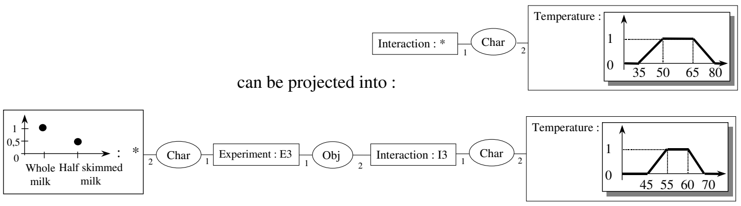

Fig. 5. An example of a projection involving fuzzy concepts

5.2 Presentation of the subsystem

Definition 7 *In UQS, the CGKB, which contains the weakly structured knowledge of our system, is a set of connected, possibly cyclic CGs.*

At the moment, the CGKB contains about 100 CGs corresponding to scientific publications which do not fit the RDB schema. These CGs have been built manually by analyzing the pertinent sentences of these publications. The CG presented in Fig. 2 belongs to the CGKB.

5.3 The CG wrapper

The CG subsystem relies on a set of *schema graphs* which allows us to define views on the CGKB.

Definition 8 *A* **schema graph** *S associated with a view V on n attributes $\{a_1, \ldots, a_n\}$ is a pair $\{g, C\}$ where g is an acyclic CG and $C = \{c_1, \ldots, c_n\}$ is a set of distinct concept vertices of g. Each c_i has a_i as a concept type.*

A schema graph is thus a CG with a set of distinguished concept vertices, which correspond to the attributes of the view. The graph presented in Fig. 6 is a schema graph for the view *BacteriocinInteraction*, the concepts of C are framed in bold.

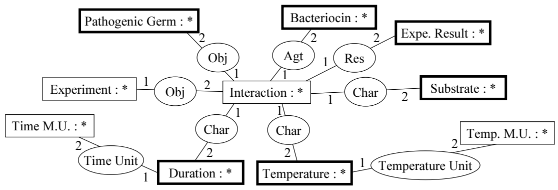

Fig. 6. An example of a schema graph for the view BacteriocinInteraction

Definition 9 *$\mathcal{V}_{cg} = \{Vg_1, \ldots, Vg_m\}$ is the set of views on the CGKB, with $P_{Vg_1}, \ldots, P_{Vg_m}$ the predicates characterizing each view.*

A CG can be interpreted in terms of FOL by means of the Φ operator [1]. The FOL which can be represented by CGs is limited to conjunctive formulae with only existential quantifiers, without negation.

Definition 10 *The predicate P_{Vg_i} associated with a view Vg_i corresponds to the logical interpretation $\Phi(g_i)$ of the schema graph g_i which defines the view.*

Example 3 *If a view V_{g_1} is defined by the following CG, and if the only attributes which can be used in a query are* PathogenicGerm *and* Bacteriocin
[PathogenicGerm:]$\overset{2}{—}$(obj)$\overset{1}{—}$[Interaction:*]$\overset{1}{—}$(agt)$\overset{2}{—}$[Bacteriocin:*]*
then the view V_{g_1} is $\{x, y | \exists z P_{Vg_1}(x, y, z)\}$ with $P_{Vg_1} = (PathogenicGerm(x) \wedge Interaction(z) \wedge Bacteriocin(y) \wedge obj(z, x) \wedge agt(z, y))$.

5.4 Query processing

When a query is asked on the CGKB, the schema graph corresponding to the considered view is specialized by the instantiation of concept vertices in order to take into account the selection attributes, giving a *query graph*.

Example 4 *The query graph presented in Fig. 7, which is a specialization of the schema graph presented in Fig. 6, corresponds to the query Q presented in Example 2. Note that the "instantiation" of the selection attributes is done in two different ways: the selection attribute $< Temperature : HumanBodyTemp. >$ is instantiated by defining a marker (which is a fuzzy one in that example), while the selection attribute $< Bacteriocin, Nisin >$ is instantiated by restricting the concept type "Bacteriocin" to its subtype "Nisin". This results from our choice to let the designer of a knowledge base the possibility to define instances of a concept type by means of individual markers or by means of subtyping [10].*

The following step of the query processing consists in projecting the query graph into all the CGs of the CGKB. In other words, we search for assertions in our KB which contain information more precise than that of the query graph.

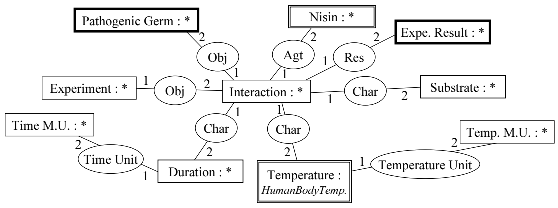

Fig. 7. An example of a query graph

Remark 2 *Given two CGs G and H, it is proven*[2] *that* $G \leq H$ *iff* $\Phi(G) \rightarrow \Phi(H)$ *[1, 11]. The logical formula associated with a query graph, which is a specialization of the associated schema graph, implies then the logical formula associated with the schema graph itself. We are currently working on the extension of that property when the considered CGs involve fuzzy sets [12]. When the query graph* Q_G *can be projected into a fact graph* F_G*, then* F_G *is a specialization of* Q_G*:* $\Phi(F_G)$ *implies* $\Phi(Q_G)$*.*

For each projection of the query graph, we extract the values of the projection attributes, in order to build the result of the query. For example, if we ask the query of Fig. 7 on a CGKB containing the CG of Fig. 2, the resulting tuple would be: $<' Listeria',' Reduction' >$. Note that the question of the existence of a projection of a graph into another graph is NP-complete. However there are polynomial cases, for instance the question of the existence of a projection of an acyclic graph into a graph. In UQS, we use the polynomial algorithm of [13].

6 Conclusion and perspectives

In this paper, we have presented a work which is part of a food risk control application using two different knowledge sources: a relational database and a CG knowledge base. These two knowledge sources allow the user: (i) to insert data involving fuzzy values represented in terms of fuzzy sets as well as (ii) to query the base with the expression of preferences, also represented by means of fuzzy sets. The integration of these two subsystems is done by means of a uniform querying language plugged into two wrappers which realize the translation of the query into queries fitting one subsystem or the other.

Our system is composed of a common interface developed in Java. That interface queries both subsystems and merges their partial answers into a global answer. The CG subsystem consists in a 5.000 lines supplement of C++, implemented on the CoGITaNT platform [14]. It includes all the mechanisms presented in this paper. It has been successfully presented to our microbiologist partners and is now operational.

Among the multiple perspectives induced by this work, two are to be studied very soon. The first one is the study of enlargement querying mechanisms,

---
[2] G and H must be normalized, i.e. each individual marker appears at most once

extending those yet implemented in the RDB [9] and in the CGKB [10]. The second one is the integration of our system into a more ambitious project, called "e.dot", which involves three computer science laboratories and a firm[3]. The goal is to build a data warehouse composed of our bases, completed by data extracted from the Web.

References

1. J.F. Sowa. *Conceptual structures - Information processing in Mind and Machine.* Addison-Welsey, 1984.
2. 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*, volume 856, pages 631–640. Springer-Verlag, 1994.
3. 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.
4. M.L. Mugnier and M. Chein. Représenter des connaissances et raisonner avec des graphes. *Revue d'Intelligence Artificielle*, 10(1):7–56, 1996.
5. L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
6. L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.
7. J.D. Ullman. *Principles of database and knowledge-base systems.* Computer Science Press, 1988.
8. R. Thomopoulos, P. Buche, and O. Haemmerlé. Representation of weakly structured imprecise data for fuzzy querying. *to appear in Fuzzy Sets and Systems*, 2003.
9. P. Buche and S. Loiseau. Using contextual fuzzy views to query imprecise data. In *Proceedings DEXA'99 (Database and EXpert system Application), Lecture Notes in Computer Science #1677*, pages 460–472, Florence, Italy, August 1999. Springer.
10. P. Buche and O. Haemmerlé. Towards a unified querying system of both structured and semi-structured imprecise data using fuzzy views. In *Proceedings of the 8th International Conference on Conceptual Structures, Lecture Notes in Artificial Intelligence #1867*, pages 207–220, Darmstadt, Germany, August 2000. Springer.
11. M. Chein and M.L. Mugnier. Conceptual graphs, fundamental notions. *Revue d'Intelligence Artificielle*, 6(4):365–406, 1992.
12. R. Thomopoulos, P. Bosc, P. Buche, and O. Haemmerlé. Logical interpretation of fuzzy conceptual graphs. In *Proceedings of the NAFIPS'2003 Conference (to appear)*, Chicago, USA, July 2003.
13. M.L. Mugnier and M. Chein. Polynomial algorithms for projection and matching. In *Proceedings of the 7th annual Workshop on Conceptual Graphs, Lecture Notes in Artificial Intelligence #754*, pages 239–251, Las Cruces, NM, USA, July 1992. Springer-Verlag.
14. D. Genest and E. Salvat. A platform allowing typed nested graphs: How cogito became cogitant. In *Proceedings of the 6th International Conference on Conceptual Structures (ICCS'1998), Lecture Notes in Artificial Intelligence #1453*, pages 154–161, Montpellier, France, August 1998. Springer.

[3] LRI/IASI, Paris XI University, INRIA/Verso, Xyleme and INA P-G, www.inria.fr/edot