# A Data Warehouse that Gathers Several Formalisms to Capture Data Heterogeneity and Incompleteness in the Field of Food Microbiological Safety

Patrice Buche*, Juliette Dibie-Barthélemy**
Ollivier Haemmerlé**, Rallou Thomopoulos***

*INRA - Mét@risk, 16 rue Claude Bernard, F-75231 Paris Cedex 05, France
Patrice.Buche@inapg.fr,
**UMR INA P-G/INRA BIA, 16 rue Claude Bernard, F-75231 Paris Cedex 05, France
{Juliette.Dibie,Ollivier.Haemmerle}@inapg.fr
***INRA - UMR IATE - bat. 31, 2 Place Viala 34060 Montpellier Cedex 1
rallou@ensam.inra.fr

**Résumé.** Notre travail porte sur la conception d'un entrepôt de données dans le domaine de l'analyse du risque alimentaire. Les données expérimentales stockées dans cet entrepôt ont trois caractéristiques principales qui ont guidé la conception de l'entrepôt : elles sont hétérogènes, elles peuvent être imprécises, et l'entrepôt est par nature incomplet. Nous proposons d'utiliser trois modèles de données : le modèle relationnel, le modèle des graphes conceptuels et le modèle XML. Ces modèles ont été étendus afin de pouvoir représenter des données imprécises sous la forme de distributions de possibilités. Les bases de données construites sur ces modèles sont interrogés simultanément avec le langage MIEL++. Dans ce langage, les préférences de l'utilisateur sont représentées par des sous-ensembles flous. Des techniques de mise en correspondance floue sont utilisées pour comparer préférences et données imprécises.

## 1 Introduction

Since 1999, our team has been working with industrial and academic partners on several projects which concern knowledge representation in the field of predictive microbiology. In the Sym'Previus (http ://www.symprevius.org) and e.dot projects (http ://www-rocq.inria.fr/verso/edot/), we work on the building of a data warehouse composed of data concerning the behaviour of pathogenic germs in food products. Those data are designed to be used in a tool dedicated to researchers in microbiology or to industrials. The goal is to help them in a decision support approach in order to prevent food products from contamination.

The information we have to store in our data warehouse presents several specificities. It is *weakly-structured* because information comes from heterogeneous sources (scientific literature, industrial partners...) and is still rapidly evolving as predictive microbiology is a research field. It is *imprecise* because of the complexity of the underlying biological processes, and because of the internal imprecision of the measurement tools. The data warehouse is *incomplete* by nature since the number of experiments is potentially infinite : it will never contain information about all the possible food products

and all the possible pathogenic germs in any possible experimental conditions.

Those three characteristics are taken into account in the following ways. The weakly structuration of the data led us to build a data warehouse composed of three bases : a relational database which contains the stable part of the information, a conceptual graph base which contains the weakly-structured part of the information and an XML base filled with data semi-automatically extracted from the Web. The imprecision of the data is represented by means of possibility distributions expressed by fuzzy sets, in each of the three bases. Finally, the incompleteness is partially solved by allowing the users to express large queries with expression of preferences in the selection criteria ; we also propose a mechanism of generalization of the queries. The knowledge of the application domain is represented by means of the *Sym'Previus ontology* which was built by experts of the domain during the Sym'Previus project.

The three bases are queried in a transparent way by means of a single user interface called the MIEL++ system. Our approach is a compromise between a data warehouse approach in which the data are formatted in order to fit the data warehouse schema [Jarke *et al.*, 2000], and a mediator approach in which the data are preserved in their orginal shape, their integration being done at query time [Wiederhold, 1995]. The MIEL++ system is a kind of mediated architecture between three different data warehouses ; each piece of information is stored in the most suited warehouse.

The aim of this paper is to present the data warehouse in its whole. In section 2 we make an overall presentation of the MIEL++ querying system. In the next sections, we present separately the two most innovative subsystems among the three which compose our data warehouse : the conceptual graph subsystem in section 3 and the XML subsystem in section 4. More detailed explanations about the relational database subsystem can be found in [Buche *et al.*, 2005].

# 2 Overall presentation

## 2.1 The MIEL++ system architecture

Fig. 1 presents an overview of the architecture of the MIEL++ system, which is composed of three subsystems : a relational database, a conceptual graph base and an XML base.

## 2.2 The Sym'Previus ontology

An ontology containing the terminological knowledge of the application domain has been developed during the Sym'Previus project. It is notably composed of :

1. a taxonomy of terms, composed of the set of attributes which can be queried on by the end user, and their corresponding reference domains. Each attribute has a reference domain which can be : (1) numeric, (2) "flat" symbolic (unordered constants such as a set of authors) or (3) hierarchized symbolic (constants partially ordered by the "kind-of" relation). Fig. 2 is a part of the taxonomy composed of the attribute Substrate and its hierarchized symbolic reference domain. The taxonomy contains the food products, the pathogenic germs...

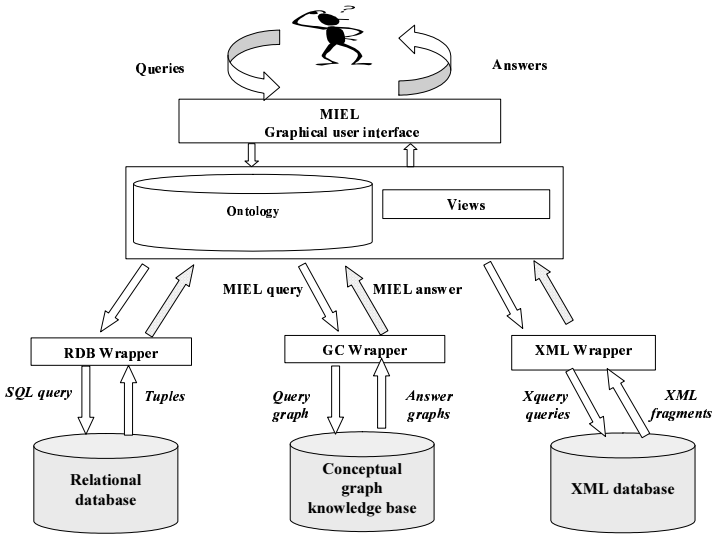


FIG. 1 – Overview of the MIEL++ system

2. a relational schema, which corresponds to the relational schema of the relational database of the MIEL++ system. That schema is composed of a set of signatures of the possible relations between the terms of the taxonomy. For example, the relation *FoodProductPH* is used to link a food product and its pH value.

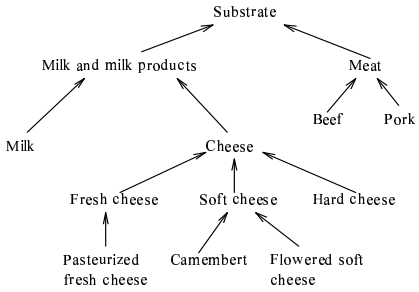


FIG. 2 – A part of the taxonomy corresponding to the attribute Substrate.

## 2.3 The MIEL++ query language

In the MIEL++ system, the query processing is done through the MIEL++ query language. This query processing relies firstly on a set of pre-written queries, called

views, which are given to help the end users express their queries and secondly on the Sym'Previus ontology which contains the vocabulary used to express the queries. We introduce the MIEL++ query language by presenting successively the underlying fuzzy set notions, the queries and the answers to a query.

### 2.3.1 Fuzzy set notions

We use the representation of fuzzy sets proposed in [Zadeh, 1965, Zadeh, 1978].

**Definition 1** A **fuzzy set** $f$ on a definition domain $Dom(f)$ is defined by a membership function $\mu_f$ from $Dom(f)$ to $[0, 1]$ that associates the degree to which $x$ belongs to $f$ with each element $x$ of $Dom(f)$.

**Definition 2** For any fuzzy set $f$ defined on a definition domain $Dom(f)$ with $\mu_f$ its membership function, we note **support(f)** $= \{x \in Dom(f) | \mu(x) > 0\}$ and **kernel(f)** $= \{x \in Dom(f) | \mu(x) = 1\}$.

The fuzzy set formalism can be used in two different ways : (1) in the database, in order to represent imprecise data expressed in terms of possibility distributions or (2) in the queries, in order to represent fuzzy selection criteria which express the preferences of the end user. In order to answer queries in a database involving fuzzy sets, we must be able to compare fuzzy sets. We present here the "possibility degree" which is classically used to evaluate the compatibility between a fuzzy selection criterion and an imprecise datum. A fuzzy set can be defined on a continuous or a discrete definition domain.

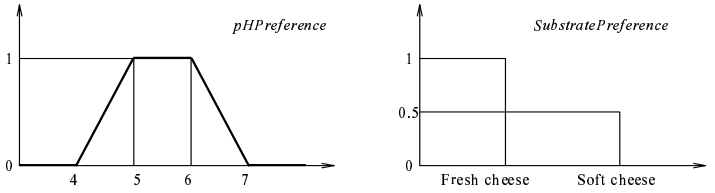


FIG. 3 – The fuzzy set pHPreference is a continuous fuzzy set and the fuzzy set SubstratePreference noted (1/Fresh cheese + 0.5/Soft cheese) is a discrete one.

**Definition 3** Let $f$ and $g$ be two fuzzy sets defined on the same definition domain $Dom$, representing respectively a selection criterion and an imprecise datum, and $\mu_f$ and $\mu_g$ being their respective membership functions. The **possibility degree of matching** between $f$ and $g$ is $\Pi(f, g) = sup_{x \in Dom}(min(\mu_f(x), \mu_g(x)))$.

### 2.3.2 The queries

In the MIEL++ language, a query is asked in a view, which is a pre-written query allowing the system to hide the complexity of the data warehouse schema. A view is characterized by its set of queryable attributes and by its actual definition.

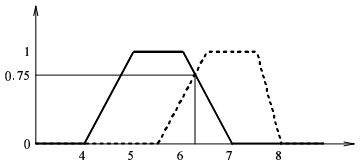


FIG. 4 – The possibility degree of matching between the two fuzzy sets is 0.75.

A query is then an instanciation of a given view by the end user, who specifies, among the set of queryable attributes of the view, the selection attributes and their corresponding searched values, and the projection attributes of the query.

**Definition 4** A **query** $Q$ asked on a view $V(a_1, \ldots, a_n)$ is defined by $Q = \{V, a_1, \ldots, a_l, <a_{l+1}, v_{l+1}>, \ldots, <a_m, v_m>\}_{1 \leq l \leq m \leq n}$ where $a_1, \ldots, a_l$ represent the list of the projection attributes and $a_{l+1}, \ldots, a_m$ represent the conjunction of the selection attributes with their respective values $v_{l+1}, \ldots, v_m$. The crisp or fuzzy values $v_i$, $i \in [l+1, m]$, must belong to the reference domain of the attributes $a_i$.

In the case where the fuzzy value of a selection attribute has a hierarchized symbolic reference domain, the fuzzy set used to represent the fuzzy value can be defined on a subset of this reference domain. We consider that such a fuzzy set implicitly defines degrees on the whole reference domain of the selection attribute. For example, if users are interested in their query by Milk, we assume that they are also interested in all the specializations of Milk. In order to take those implicit degrees into account, the *fuzzy set closure* has been defined in [Thomopoulos *et al.*, 2003a, Buche *et al.*, 2005]. The fuzzy set closure is systematically used when a comparison involving two fuzzy sets defined on a hierarchical domain is considered.

### 2.3.3 The answers

An answer to a query $Q$ must (1) satisfy all the selection criteria of $Q$ and (2) associate a constant value with each projection attribute of $Q$.

**Definition 5** Let $<a, v>$ be a selection criterion and $v'$ a value of the attribute $a$ stored in the database. The selection criterion $<a, v>$ is satisfied iff $\Pi(v, v') > 0$ in the meaning of definition 3 : the intersection between the two fuzzy sets is non-empty.

As the selection criteria of a query are conjunctive, we use the *min* operator to compute the adequation degree associated with the answer.

**Definition 6** An **answer** $A$ to a query $Q = \{V, a_1, \ldots, a_l, <a_{l+1}, v_{l+1}>, \ldots, <a_m, v_m>\}$, is a set of tuples, each of the form $\{v_1, \ldots, v_l, ad\}$, where $v_1, \ldots, v_l$ correspond to the crisp or fuzzy values associated with each projection attribute $a_1, \ldots, a_l$ of $Q$, where all the selection criteria $a_{l+1}, \ldots, a_m$ of $Q$ are satisfied with the respective possibility degrees $\Pi_{l+1}, \ldots, \Pi_m$ and where $ad$ is the adequation degree of the answer $A$ to the query $Q$ defined as follows : $ad = min_{i=l+1}^{m}(\Pi_i)$.

# 3 The conceptual graph subsystem

## 3.1 The schema of the conceptual graph database

The flexibility of the conceptual graph model [Sowa, 1984, Mugnier et Chein, 1996] played an important part in the choice of that knowledge representation model in the MIEL++ system : no static schema is used and we can build pieces of information which have different shapes by easily adding or removing graph vertices.

In the conceptual graph model, the *support*, which contains the ground vocabulary, must be defined. A conceptual graph is always built on a given support. We now summarize how the support is built in the conceptual graph subsystem.

The **concept type set** is used to represent the main part of the ontology of the MIEL++ system, since it is a partially ordered set, designed to contain the concepts of a given application. It is built as follows. A concept type $t_a$ is associated with each attribute $a$ of the taxonomy. If $a$ is a hierarchized attribute, then a concept type $t_{v_i}$ is associated with each element $v_i$ of the reference domain of $a$. The $t_a$'s and $t_{v_i}$'s are inserted into the concept type set, w.r.t. the partial order of that reference domain.

Note that the hierarchized structure of the concept type set allowed us to store the attribute names and the values belonging to hierarchized reference domains into the same set. Fig. 5 represents a part of the concept type set of the MIEL++ conceptual graph database. The attribute *Substrate* and its hierarchized reference domain presented in Fig. 2 appear as a partial subgraph of that concept type set.

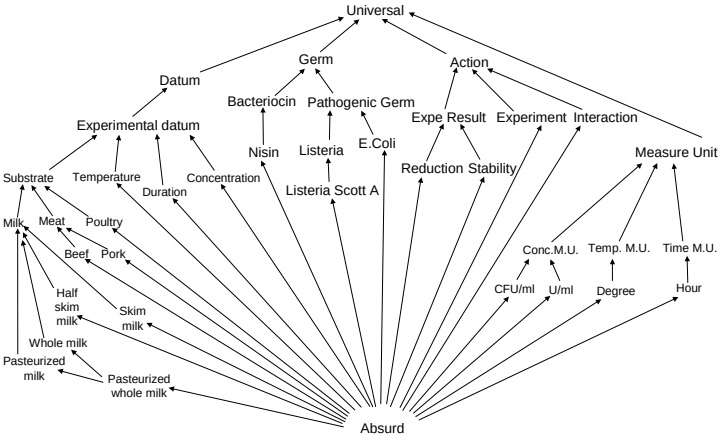


FIG. 5 – A part of the concept type set of the MIEL++ conceptual graph database

The **set of individual markers** is used to store the reference domain of each attribute $a$ that has a *flat symbolic* or a *numerical* reference domain. More precisely, all the values of the reference domains of the *flat symbolic* attributes as well as the values of $\mathbb{R}$ are inserted into the set of individual markers.

We do not detail the **set of relation types** since, as often in the conceptual

graph model, it does not play an important part in our conceptual graph database, the semantics being mainly contained in the concept vertices.

In order to allow a homogeneous expressivity between the relational database and the conceptual graph database, we proposed an extension of the conceptual graph model to the representation of fuzzy values presented in [Thomopoulos *et al.*, 2003b]. A fuzzy set can appear in two ways in a concept vertex : (i) as a *fuzzy type* when the reference domain of the fuzzy set is hierarchized. A fuzzy type is a fuzzy set defined on a subset of the concept type set ; (ii) as a *fuzzy marker* when the reference domain of the fuzzy set is *"flat symbolic"* or *numerical*. A fuzzy marker is a fuzzy set defined on a subset of the set of individual markers.

The conceptual graph database is composed of a set of conceptual graphs, each of them representing an elementary datum.

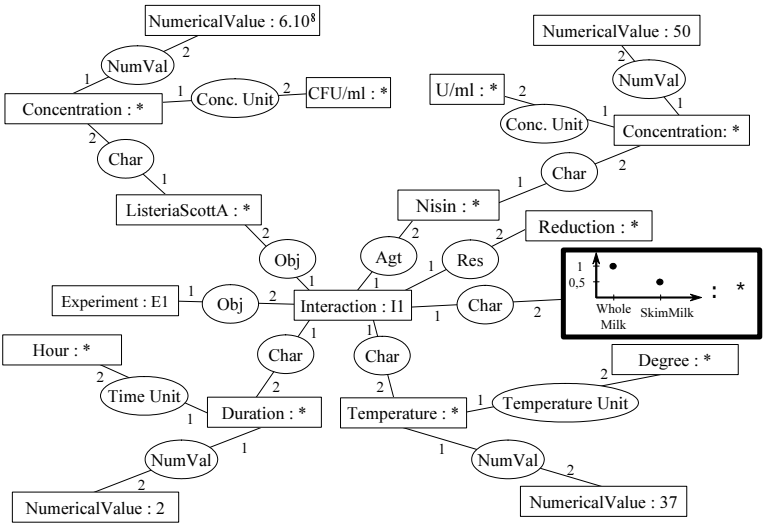


FIG. 6 – An example of conceptual graph extracted from the MIEL++ conceptual graph database. The concept vertex framed in bold is a concept with a fuzzy type.

## 3.2 Query processing in the conceptual graph subsystem

### 3.2.1 The views

The conceptual graph subsystem relies on a set of *view graphs* which allow us to define views on the conceptual graph database. A view graph is a pre-defined "empty" query which has to be instantiated in order to become an actual query graph.

When a query is asked in the conceptual graph subsystem, the view graph corresponding to the considered view is specialised by instantiating concept vertices in order to take into account the selection attributes. The result is a *query graph*.

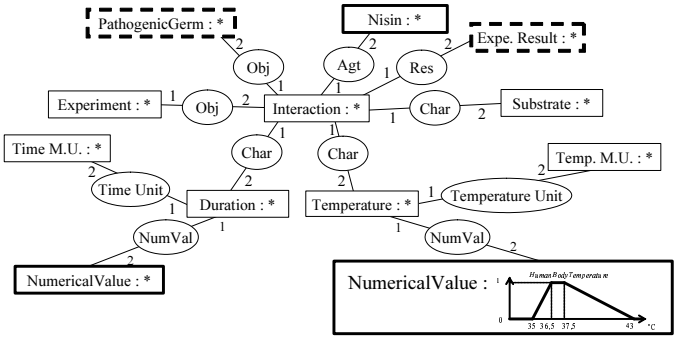


FIG. 7 – An example of a query graph. The selection attributes are framed in bold, the projection attributes are dashed. One of the selection criteria is expressed by a concept with a numerical fuzzy marker.

### 3.2.2 The query processing

In the conceptual graph subsystem of the MIEL++ system, the query processing consists in searching for conceptual graphs which contain a more precise information than the information contained in the query (we search for specializations of the query graph) or, at least, for conceptual graphs which contain "approximate" answers. In order to find such conceptual graphs, we propose to use the $\delta$-projection operation which is a flexible mapping operation between two conceptual graphs. The $\delta$-projection is presented in [Thomopoulos *et al.*, 2003a]. It is adapted from the classic projection operation, by taking into account the possibility degree of matching (see definition 3).

The query processing in the conceptual graph subsystem consists in selecting the view graph, building the query graph, and $\delta$-projecting that query graph into all the conceptual graphs of the database. Every time a $\delta$-projection into a fact graph $A_G$ is found, the conceptual graph $A_G$ is considered an answer graph. A tuple with the adequation degree $\delta$ is built using this answer graph by extracting the values of the projection attributes. For example, if we asked the query of Fig. 7 on a conceptual graph database containing the conceptual graph of Fig. 6, the resulting tuple would be : $('ListeriaScottA', 'Reduction', \delta = 1)$.

# 4 The XML subsystem

## 4.1 The XML base

The XML base has been built in the MIEL++ system in order to store information retrieved from the Web. More precisely, we focus on tables included in scientific papers, which contain experimental data. In order to obtain an efficient querying of the XML base, the tables extracted from the Web are transformed into SML documents by means of a semantic enrichment process according to the ontology of the MIEL++

data model. SML process [Saïs *et al.*, 2005] achieves two kinds of semantic enrichment : (i) it associates terms of a Web table with their corresponding terms in the Sym'Previus ontology (for example, the term *Stewed exotic fruit* of a Web table is associated with the term *Sweet fresh fruit* belonging to the ontology), (ii) it instantiates semantic relations of the ontology which appear in the Web table schema (for example, the relation *FoodProductPH* is instantiated in a Web table that contains a column composed of food product names and another column with pH values).

Moreover, in [Buche *et al.*, 2004], we propose a fuzzy semantic tagging of the terms of a Web table : each association between a term of a Web table and a term belonging to the ontology is weighted by a degree of possibility depending on their syntactic closeness (for example, the association between the term *Stewed exotic fruit* of a Web table and the term *Sweet fresh fruit* belonging to the ontology is weighted by the degree of possibility 0.33 computed thanks to the words common to both terms). The SML documents thus contain fuzzy data : for a given term of a Web table, its associated terms belonging to the ontology are represented by a discrete fuzzy set.

The SML documents are modeled as fuzzy data trees [Buche *et al.*, 2004] which allow one to represent fuzzy values. According to the definition of [Aguiléra *et al.*, 2000, Xyleme, 2001], a data tree is a triple $(t, l, v)$ where $(t, l)$ is a labelled tree and $v$ is a partial value function that assigns a value to nodes of $t$. The schema of a data tree is defined by a type tree which is a labelled tree such that no node has two children labelled the same. The representation of fuzzy values relies on the fuzzy set formalism.

**Definition 7** A continuous fuzzy set $f$ is represented by a data tree which is composed of a root labelled *CFS* and of four leaves labelled *minSup*, *minKer*, *maxKer*, *maxSup* of respective values $\min(\text{support}(f))$, $\min(\text{kernel}(f))$, $\max(\text{kernel}(f))$ and $\max(\text{support}(f))$.

**Definition 8** A discrete fuzzy set $f$ is represented by a data tree which is composed of a root labelled *DFS* and such that for each element $x$ of $Dom(f)$, there exists a node labelled *ValF* that has two children labelled *Item* and *MD* (for Membership Degree) of respective values $x$ and $\mu_f(x)$.

In a fuzzy data tree, the function $v$ can assign a crisp value, a continuous or a discrete fuzzy value to a node, which is then called **crisp** or **fuzzy value node**.

**Definition 9** A **fuzzy data tree** is a triple $(t, l, v)$ where $(t, l)$ is a labelled tree and $v$ is a partial value function that assigns a value to crisp and fuzzy value nodes of $t$. The value assigned to a crisp value node is an atomic value and the one assigned to a fuzzy value node is a data tree with a root labelled CFS or DFS which respectively conforms to definitions 7 and 8.

The Sym'Previus ontology is represented in the XML subsystem as a data tree stored in an XML document.

## 4.2 Query processing in the XML subsystem

### 4.2.1 The views

The XML subsystem relies on a set of views, which allow one to query the base.

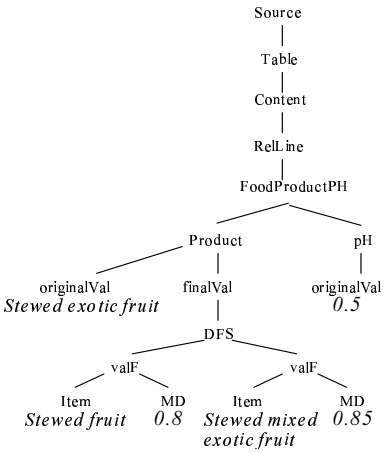


FIG. 8 – An example of fuzzy data tree obtained by fuzzy semantic tagging. The original value *Stewed exotic fruit* is a term of a Web table and the final value represents the terms belonging to the ontology that are associated with the original value.

**Definition 10** A **view that conforms to a type tree** $(t_T, l_T)$ is a triple $V=(t_V, l_V, w_V)$ where $(t_V, l_V)$ is an instance of $(t_T, l_T)$ and $w_V$ is a partial function that assigns the value *ql* (for queryable leaf) to crisp and fuzzy value nodes of $t_V$ which are queryable (i.e. that can be used as selection or projection attributes).

Depending on the kind of information we want to retrieve from the XML base, the type tree associated with the view is composed of a subset of relations that belong to the relational schema of the ontology. A query is an instanciation of a given view, where the end user specifies, among the set of queryable value nodes of the view, the selection and the projection value nodes of the query.

### 4.2.2 The query processing

In the XML subsystem, the query processing consists in searching for data trees which exactly fit the structure of the XML query and satisfy its selection criteria. As described in [Buche *et al.*, 2004], the following steps are processed : (i) selecting the view, (ii) building the query, (iii) doing a valuation of that query into all the data trees of the XML base. Every time a valuation of the query with respect to a data tree $D$ is found, the possibility degree of matching between $D$ and the query is computed and the XML query processor builds an answer, composed of a set of tuples by extracting the values of the projection attributes from the data tree $D$.

**Example 1** *The answer to the query Q of Fig. 9 using the data tree of Fig. 8 is :* { *Stewed exotic fruit, (0.8/Stewed fruit, 0.85/Stewed mixed red fruit), 5.0, ad=0.85* }.

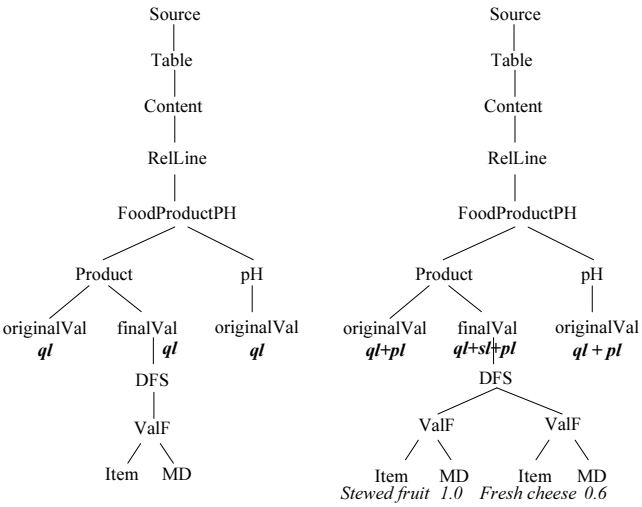


FIG. 9 – An example of query (right tree) expressed in a view (left tree) using the relation *FoodProductPH* with three queryable attributes.

# 5 Conclusion and perspectives

The MIEL++ system has been implemented. It conforms to the J2EE standard (HTML client and servlet/JSP server). The relational database and XML subsystems have been developed in Java. The conceptual graph subsystem has been developed in C++ using the CoGITanT platform [Genest, 2003]. At the moment, the relational database contains about 10.000 data. The MIEL++ relational database subsystem is used by our industrial partners of the Sym'Previus project. The conceptual graph database contains about 150 conceptual graphs manually built by analyzing the relevant sentences of scientific publications which do not fit the relational database subsystem schema. The XML base contains about 200 scientific papers retrieved from the Web. Both conceptual graph and XML subsystems are currently under testing in collaboration with our partners of the Sym'Previus project.

As mentioned in section 4, in the current version of SML, fuzzy data stored in SML documents represent the mapping between terms found in Web tables and their corresponding terms in the data warehouse taxonomy. As we accept partial instanciations of the semantic relations of the ontology in a Web table, in a future work, we will also introduce fuzziness in the representation of semantic relation instanciation, and we will have also to adapt the algorithm we have proposed in this paper.

## Références

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

[Buche *et al.*, 2004] P. Buche, J. Dibie-Barthélemy, O. Haemmerlé, et M. Houhou. Towards flexible querying of xml imprecise data in a data warehouse opened on the web. In *Proceedings of the 6th International Conference On Flexible Query Answering Systems (FQAS'04), Lecture Notes in AI #3055*, pages 28–40, Lyon, France, June 2004. Springer.

[Buche *et al.*, 2005] P. Buche, C. Dervin, O. Haemmerlé, et R. Thomopoulos. Fuzzy querying on incomplete, imprecise and heterogeneously structured data in the relational model using ontologies and rules. *To appear in IEEE Transactions on Fuzzy Systems*, 2005.

[Genest, 2003] D. Genest. Cogitant v-5.1 - manuel de référence. Web site, 2003. http ://cogitant.sourceforge.net.

[Jarke *et al.*, 2000] M. Jarke, M. Lenzerini, Y. Vassiliou, et P. Vassiliadis, editors. *Fundamentals of Data Warehouses*. Springer-Verlag, January 2000.

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

[Saïs *et al.*, 2005] F. Saïs, H. Gagliardi, O. Haemmerlé, et N. Pernelle. Enrichissement sémantique de documents sml représentant des tableaux. In *Actes des 5èmes journées Extraction et Gestion des Connaissances, EGC'2005*, Paris, France, janvier 2005.

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

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

[Thomopoulos *et al.*, 2003b] R. Thomopoulos, P. Buche, et O. Haemmerlé. Representation of weakly structured imprecise data for fuzzy querying. *Fuzzy Sets and Systems*, 140-1 :111–128, 2003.

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

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

[Zadeh, 1965] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8 :338–353, 1965.

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

## Summary

We present the design of a data warehouse in the field of risk analysis in food. Experimental data stored in the data warehouse have three characteristics which have guided us in the design of the warehouse. They are heterogeneous, they may be imprecise, and the data warehouse is incomplete by nature. Three data models have been used to represent the data : the relational model, the conceptual graph model and the XML model. Those models have been extended to be able to represent imprecise data as possibility distributions. The databases built on those models 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.