Towards flexible querying of XML imprecise data in a dataware house opened on the Web

Patrice Buche[1] , Juliette Dibie-Barthélemy[1] , Ollivier Haemmerlé[12] and Mounir Houhou[1]

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

Abstract. This paper describes a new subsystem of the Sym'Previus knowledge base. This knowledge base contains information useful to help experts in the field of predictive microbiology. Information has several specific properties: it is incomplete, imprecise and heterogeneous. In the pre-existing Sym'Previus knowledge base, stable data are stored in a relational database and data which do not fit the relational structure are stored in a conceptual graph knowledge base. The MIEL language permits to scan simultaneously both bases in a transparent way for the user, using fuzzy queries. The new subsystem described in the paper contains information found on the Web to complete the knowledge base. This information is stored in XML format. Firstly, we extend the XML model of the knowledge base to represent imprecise data as possibility distributions. Secondly, we present the mapping process used to translate a MIEL query into an XML query to scan the XML knowledge base.

1 Introduction

Our team is involved in a national project, called Sym'Previus, which aims at building a tool to help experts in the field of predictive microbiology. This tool is composed of two subsystems: (1) a knowledge base which gathers knowledge about the behaviour of pathogenic germs in food products, (2) a simulation tool. This paper is dedicated to the knowledge base subsystem. The information stored in this knowledge base is characterized by several specific properties: it is incomplete, imprecise and weakly-structured. This information is incomplete by nature because it is not possible to store information about all existing food products and all existing germs. It is imprecise because of the complexity of the underlying biological processes. It is weakly-structured because the information comes from heterogeneous sources (bibliographical literature, industrial partners, ...) and from knowledge about predictive microbiology, which is still a field of research, and as such is then evolving rapidly. To take those properties into account, we have made choices in the design of the Sym'Previus knowledge base and its querying system, called MIEL (see [7, 8]). The more stable information (in terms of structure) is stored in a relational database. The information which does not fit the relational structure is stored in a more flexible representation model: the conceptual graph model. The imprecise information stored in the Sym'Previus knowledge base is represented by possibility distributions in both bases.

The MIEL querying system has several properties: (1) it scans both bases simultaneously in a transparent way (the final user does not know that the knowledge is stored in two different formats); (2) by using MIEL, the final user queries the bases through views which hide the complexity of bases schemas; (3) MIEL executes a fuzzy query: predefined selection attributes are available in the views and the final user has only to specify his/her preferences expressed as fuzzy sets; (4) answers are returned ordered according to their relevance to the selection criteria. In the framework of another national project, called E.dot, our aim is to complete the Sym'Previus knowledge base with information, semi-automatically found on the Web. Information retrieved from the Web in different formats (html, pdf, excel, ...) is stored in XML format in an additional local knowledge base especially adapted to store XML documents: the Xyleme Zone server (see [16]).

Our paper may be related to three types of works. In a first category of study, the fuzzy set framework has been shown to be a sound scientific choice to model flexibles queries (see [5]). It is a natural way of representing the notion of preference using a gradual scale. In a second category of works, the fuzzy set framework has also been proposed to represent imprecise values by means of possibility distributions (see [18]). Several authors have developed this approach in the context of databases (see [3, 6, 10]). In a third category of paper, flexible query languages have been proposed to scan XML data (see [4, 9, 11, 13]). Some flexibility is provided by using similarity relationships [13], partial structural match techniques [2, 9, 11, 12], and flexible constraints to determine significance degrees of indexed terms in the data [4].

Our paper is a very first step to adapt the MIEL system in order to be able to query an XML knowledge base. As we already did for the relational database and the conceptual graph knowledge base, we make the assumption that XML data fit predefined XML schemas. The aims of this paper are: (1) to propose a way of representing imprecise information expressed as possibility distribution in an XML knowledge base, (2) to extend MIEL to be able to scan the additional XML knowledge base. We recall briefly in section 2: firstly, the fuzzy set framework we have used to represent preferences in the queries and imprecise data and secondly, the MIEL language. In section 3 we propose an extension of the XML model to represent imprecise information. In section 4 we present the translation process of a MIEL query to scan the XML knowledge base.

2 Background

2.1 Fuzzy set theory

We use, in this article, the representation of fuzzy sets proposed in [17, 18].

Definition 1 A **fuzzy set** f on a definition domain $Dom(f)$ is defined by a membership function μ_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 μ_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 knowledge base, 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 final user. In order to answer queries in a knowledge base involving fuzzy sets, we must be able to compare fuzzy sets. We studied several comparison degrees in [14]. 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 discrete definition domain.

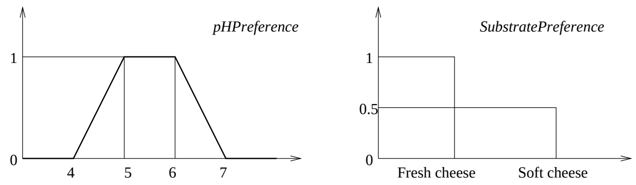

Fig. 1. The fuzzy set pHPreference noted [4,5,6,7] is a continuous fuzzy set and the fuzzy set SubstratePreference noted (1/Fresh cheese + 0.5/Soft cheese) is a discrete one.

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 μ_f and μ_g being their respective membership functions. The **possibility degree of matching** between f and g is $\Pi(f, g) = sup_{x \in Dom}(min(\mu_f(x), \mu_g(x)))$.

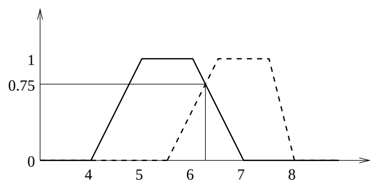

Fig. 2. The possibility degree of matching between the two fuzzy sets is 0.75.

Remark 1 *The evaluation of the possibility degree between a fuzzy selection criterion and a crisp datum is a particular case of definition 3.*

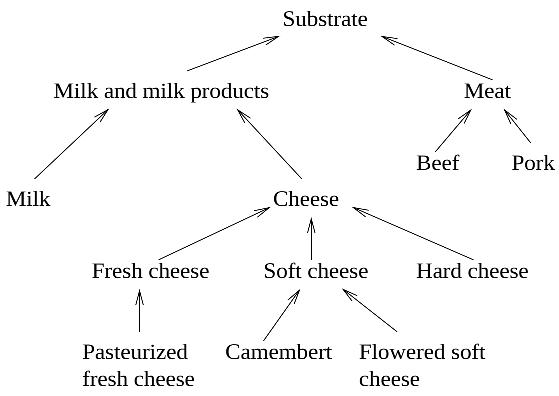

Fig. 3. A part of the ontology composed of the attribute Substrate and its hierarchized symbolic definition domain.

2.2 The MIEL query language

In the MIEL system, the interrogation of a knowledge base is made through the MIEL query language. This interrogation relies on a set of pre-written queries, called views, which are given to help the final user to express his/her queries and an ontology which contains the vocabulary used to express the queries. We introduce the MIEL query language by presenting successively the ontology, the views, the queries and the answers to a query.

The ontology The ontology of a knowledge base in the MIEL system is composed of the set of attributes which can be queried by the final user and their corresponding definition domain. As a matter of fact, each attribute is defined on a definition domain which can be of three kinds: (1) numeric, it is then completely ordered (for example the real numbers), (2) "flat" symbolic, it is then composed of unordered constants (for example a set of authors) or (3) hierarchized symbolic, it is then partially ordered by the "kind-of" relation.

The views Final users are not experts and cannot write queries from scratch. The views are the way given to the final user to query the knowledge base, without having to know the complexity of the schema of the base. In the MIEL system, a view is composed of a visible part which is the set of "queryable" attributes and a hidden part which is the description of the structure of the view (the way the attributes are linked together in the knowledge base).

The queries A query is an instanciation of a given view by the final user, by specifying, among the set of queryable attributes of the view, which are the selection attributes and the projection attributes of the query.

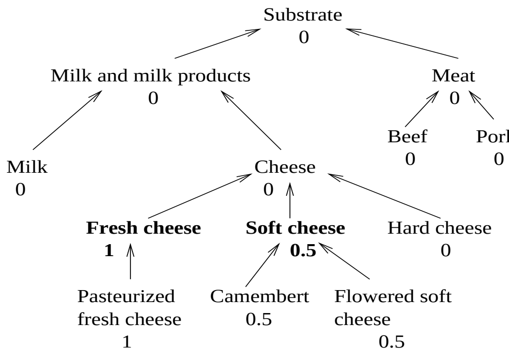

Fig. 4. Fuzzy set in extension corresponding to the fuzzy set SubstratePreference.

Definition 4 A **query** Q asked on a view V defined on n attributes $\{a_1, \ldots, a_n\}$ 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 conjonction 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 definition domain of the attributes a_i.

We are interested in the fuzzy value of a selection attribute which has a hierarchized symbolic definition domain. In that case, the fuzzy set used to represent the fuzzy value can be defined on a subset of the hierarchized symbolic definition domain of the selection attribute. Such a fuzzy set is called a **fuzzy set in intention** since it is considered to define degrees implicitly on the whole definition domain of the selection attribute. In order to take those implicit degrees into account, a **fuzzy set in extension** is associated with every fuzzy set in intention (see [14, 15] for more details).

Example 1 *Let us consider the fuzzy value (1/Fresh cheese + 0.5/Soft cheese) (see figure 1) assigned by the final user to the selection criterion Substrate. It can be interpreted as "the final user wants fresh cheese as a substrate and he/she also accepts soft cheese but with a lower interest". Since the selection criterion Substrate has a hierarchized symbolic definition domain (see figure 3), we can consider that the final user who is interested in fresh cheese is also interested in all the kinds of fresh cheese such as pasteurized fresh cheese. The fuzzy set in extension given in figure 4 takes the "kind-of" relation into account by propagating the degree associated with a value to all the specializations of that value.*

When a fuzzy set is used to represent a selection criterion defined on a hierarchized symbolic definition domain, the fuzzy set in extension is computed and used to search for satisfying answers in the knowledge base. Note that this mechanism of transformation of a fuzzy set in intention into a fuzzy set in extension can also be used when the fuzzy set is used to represent a fuzzy datum.

The answers An answer to a query Q must (1) satisfy all the selection criteria of Q in the meaning of definition 5 given below and (2) associate a constant value with each projection attribute of Q.

Definition 5 Let $<a, v>$ be a selection criterion and a value v' of the attribute a stored in the knowledge base. The selection criterion $<a, v>$ is satisfied iff $\Pi(v, v') > 0$ in the meaning of definition 3.

As the selection criteria of a query are conjonctive, 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 degree $\Pi_{l+1}, \ldots, \Pi_m$ 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 Sym'Previus XML knowledge base

In order to store the information retrieved from the Web in different formats (html, pdf, xml, excel, ...), we propose to add a third knowledge base in the Sym'Previus knowledge base, represented with the XML format. We propose to use the tree-based model as the one proposed in the Xyleme Project [1, 16] in order to represent the XML knowledge base with a clear formalization. One of the specificities of the data stored in the Sym'Previus knowledge base is their imprecision. We propose an extension of the tree-based model in order to represent such fuzzy data. First, we briefly give the definitions of the tree-based model used to represent the XML knowledge base. Second, we describe how to represent fuzzy values in the XML knowledge base. Third, we present the Sym'Previus XML knowledge base.

3.1 Preliminaries notions: the tree-based model

In the tree-based model, an XML knowledge base is a set of data trees, each of them representing an XML document.

Definition 7 A **data tree** is a triple (t, l, v) where t is a finite tree, l a labelling function that assigns a label to each node in t and v a partial value function that assigns a value to nodes of t. The pair (t, l) is called a **labelled tree**.

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. A data tree (t, l, v) is said to be an **instance** of a type tree (t_T, l_T) if there exists a strict type homomorphism from (t, l) to (t_T, l_T) as defined below.

Definition 8 Let (t, l) and (t', l') be two labelled trees. The mapping h from nodes of t into nodes of t' is a **strict structural homomorphism** if and only if (1) h preserves the root of t: $\text{root}(t') = h(\text{root}(t))$ and (2) h preserves the structure of t: whenever node m is a child of node n, $h(m)$ is a child of $h(n)$. The mapping h is a **strict type homomorphism** if and only if h is a strict structural homomorphism which preserves the labels of t: for each node n of t, $l(n)=l'(h(n))$.

The **schema** of an XML knowledge base is defined by the set of type trees which are associated with the data trees that it contains.

3.2 The representation of fuzzy values in the tree-based model

We propose to use the fuzzy set formalism in order to represent imprecise data expressed in terms of possibility distributions. In the tree-based model, a fuzzy set is represented by a data tree. We give the definition of a continuous fuzzy set and then of a discrete one.

Definition 9 Let f be a continuous fuzzy set. f is represented by a data tree which is composed of a root labelled *CFS* and composed of four leaves labelled *minSup*, *minKer*, *maxKer*, *maxSup* of respective values min(support(f)), min(kernel(f)), max(kernel(f)) and max(support(f)).

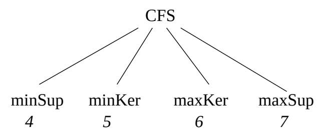

Fig. 5. Data tree representing the continuous fuzzy set pHPreference of figure 1.

Definition 10 Let f be 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 $\mu(x)$.

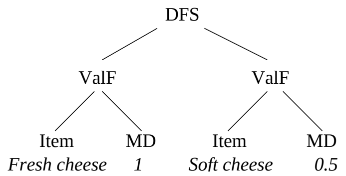

Fig. 6. Data tree representing the discrete fuzzy set SubstratePreference of figure 1

3.3 The Sym'Previus XML knowledge base

The Sym'Previus XML knowledge base is a set of data trees, which are instance of type trees, and allow one to represent fuzzy values. According to definition 7, 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. In a fuzzy data tree, the nodes to which a value can be assigned by the function v are of two types. Firstly, they can be nodes having a (continuous or discrete) fuzzy value, that is to say nodes having only one child labelled CFS or DFS according to definitions 9 and 10. Secondly, they can be leaves that are not descendants of nodes labelled CFS or DFS (those leaves have then a crisp value). These two types of nodes are called respectively **fuzzy and crisp leaves**.

Definition 11 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 leaves of t. The value assigned to a crisp leaf is an atomic value and the one assigned to a fuzzy leaf is a data tree with a root labelled CFS or DFS respectively conform to definitions 9 and 10.

Example 2 *The fuzzy data tree $\mathcal{D}$ of figure 7 talks about experimental data retrieved from the bibliographical source Partner1_03. It concerns the behaviour of Bacillus Cereus in a substrate which may be Fresh Cheese (but it may be also camembert) whose pH value is between 6 and 6.8. This experience studies the influence of conservation temperature on the level of contamination of the substrate by Bacillus Cereus. Two temperature have been tested (8 and 25 celsius degrees). At 8C, the initial contamination was 1950 CFU/g. After 3 days, it was 1000 CFU/g...*

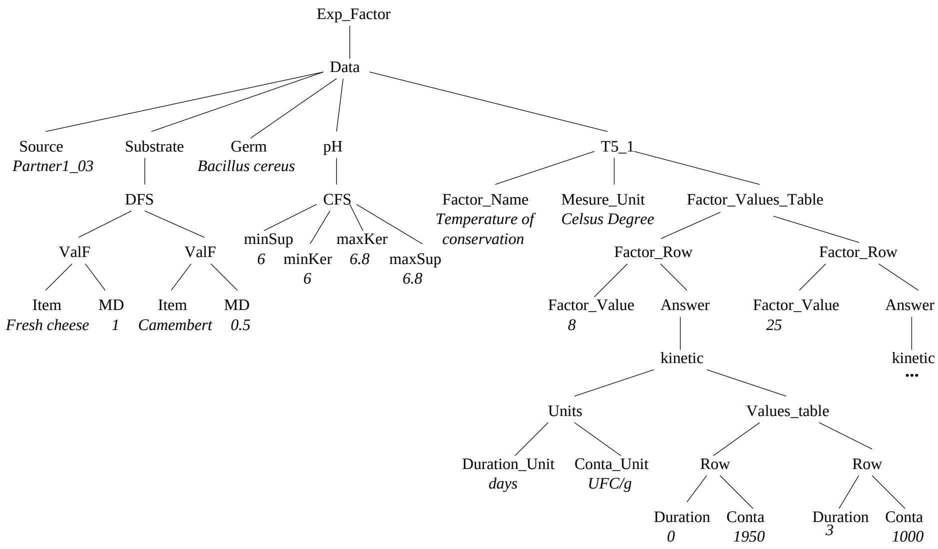

Fig. 7. Fuzzy data tree $\mathcal{D}$ representing one experimental result belonging to the bibliographical source Partner1_03.

Remark 2 Assigning a value to a continuous (resp. discrete) fuzzy leaf means assigning a value to each of its leaves labelled *minSup*, *minKer*, *maxKer* and *maxSup* (resp. to each of its leaves labelled *Item* and *MD*).

Example 3 *The data tree $\mathcal{D}$=($t_{\mathcal{D}}$, $l_{\mathcal{D}}$, $v_{\mathcal{D}}$) of figure 7 is an instance of the type tree $\mathcal{T}$=($t_{\mathcal{T}}$, $l_{\mathcal{T}}$) of figure 8: there exists a strict type homomorphism from ($t_{\mathcal{D}}$, $l_{\mathcal{D}}$) to $\mathcal{T}$ (see definition 8). Note that the mapping between the nodes of $t_{\mathcal{D}}$ and the nodes of $t_{\mathcal{T}}$ only concerns a subset of the nodes of $t_{\mathcal{T}}$.*

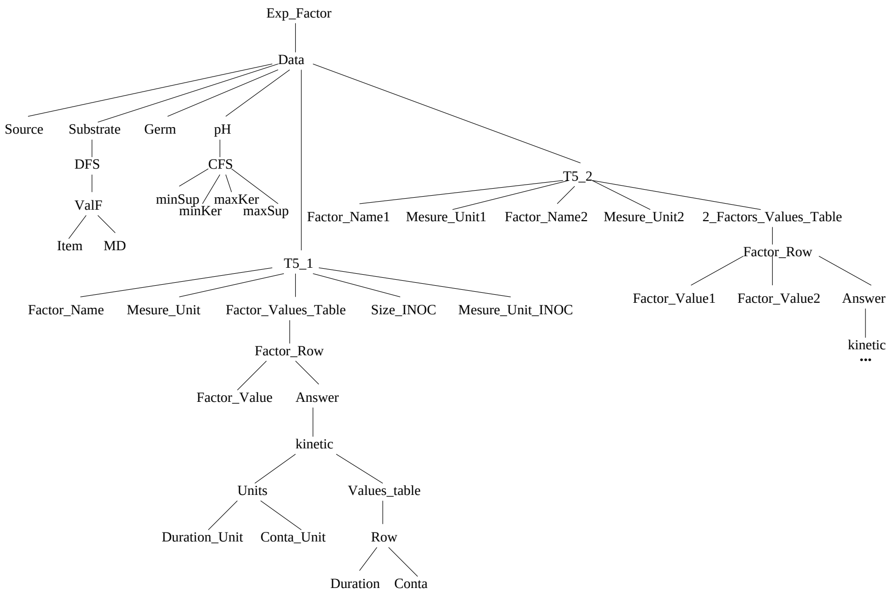

Fig. 8. Type tree $\mathcal{T}$ representing one experimental result belonging to a bibliographical source.

4 The interrogation of the XML knowledge base in the MIEL system

In order to ensure a uniform interrogation of Sym'Previus knowledge base, we propose to query the XML knowledge base, represented in the tree-based model, by means of the MIEL query language. In order to ensure a mapping between the MIEL query language and the XML subsystem of the MIEL system, we define successively the ontology, the views, the queries and the answers in this subsystem.

4.1 The ontology

In the XML knowledge base, the ontology defined in section 2.2 is represented as a set of trees stored in a XML document. Each tree materializes the definition domain and the partially ordered "kind-of" relation associated with a queriable leaf of the XML knowledge base (queriable leaf Substrate by example shown in figure 3).

4.2 The views

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

Definition 12 A **view conform 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 to crisp and fuzzy leaves of t_V, indicating that such leaves are queryable ones.

4.3 The queries

A query is an instanciation of a given view by the final user, by specifying, in the set of queryable leaves of the view, which are the selection leaves and which are the projection leaves of the query.

Definition 13 A **query** conform to a type tree (t_T, l_T) is a 6-tuple Q=(t_Q, l_Q, w_Q, p_Q, s_Q, ws_Q) where:

- (t_Q, l_Q, w_Q) is a view conform to (t_T, l_T);
- p_Q is a partial function that assigns the value pl to queryable leaves of the view, allowing one to identify the **projection leaves** of the query;
- s_Q is a partial function that assigns the value sl to queryable leaves of the view, allowing one to identify the **selection leaves** of the query, also called selection criteria;
- ws_Q is a partial value function that assigns a value to the selection leaves of the query, such that the value assigned to a crisp leaf is an atomic value and the one assigned to a fuzzy leaf is a data tree with a root labelled CFS or DFS respectively conform to definitions 9 and 10.

The value of a selection criterion given by the final user can be crisp as well as fuzzy. In the second case, the fuzzy set formalism is used to represent a fuzzy selection criterion which expresses the preferences of the final user.

Example 4 *The following query Q (see figure 9) expresses that the final user wants to obtain the germ, the factor name and the substrate from the view Exp_Factor. Moreover, the fuzzy value assigned by the final user to the selection criterion Substrate can be interpreted as "he wants fresh cheese as a substrate, but he also accepts soft cheese but with a lower interest".*

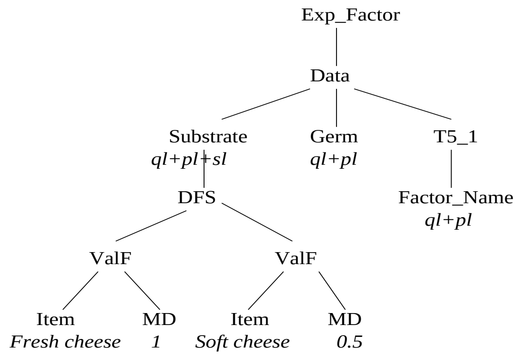

Fig. 9. Query Q in the view Exp_Factor which contains the results of the effect of one factor in biological experiments

Remark 3 *When a fuzzy set is used to represent a fuzzy selection criterion defined on a hierarchized symbolic definition domain, the fuzzy set in extension is computed and used to search for satisfying answers in the knowledge base as seen in subsection 2.2.*

4.4 The answers

An answer to a query Q must (1) satisfy all the selection criteria of Q in the meaning of definition 5 and (2) associate a constant value with each projection leaf of Q. The search of the answers to a query in an XML knowledge base is made through the valuation of the query on the data trees of the knowledge base as defined below.

Definition 14 Let Q=(t_Q, l_Q, w_Q, p_Q, s_Q, ws_Q) be a query conform to a type tree T=(t_T, l_T) and D=(t_D, l_D, v_D) be a data tree instance of the type tree T. A **valuation** of Q with respect to D is a mapping σ_D from the tree t_Q of Q into the tree t_D of D such that (1) σ_D is a strict type homomorphism from (t_Q, l_Q) into (t_D, l_D) and (2) σ_D satisfies each selection criterion n_s^i, $i \in [1, m]$, of Q with the possibility degree $\Pi(ws_Q(n_s^i), v_D(\sigma_D(n_s^i)))$. The adequation degree of the data tree D to the query Q through the valuation σ_D is $ad_D = min_{i \in [1,m]}(\Pi(ws_Q(n_s^i), v_D(\sigma_D(n_s^i))))$.

We can now give the definition of an answer to a query in an XML knowledge base. An answer is a set of tuples, each tuple corresponding to a set of values given to each projection leaf of the query.

Definition 15 An **answer** to a query Q=(t_Q, l_Q, w_Q, p_Q, s_Q, ws_Q) composed of m projection leaves noted n_p^1, ..., n_p^m in an XML knowledge base $\mathcal{KB}$ is a set of tuples, each tuple being defined as follows: $\{ \cup_{i=1}^m v_D(\sigma_D(n_p^i)) \cup ad_D \mid D$ is a data tree of $\mathcal{KB}$ and σ_D is a valuation of Q with respect to $D\}$.

Example 5 *The answer to the query Q of figure 9 in the XML knowledge base of figure 7 is: {(1/Fresh cheese, 0.5/Camembert), Bacillus cereus, Temperature of conservation, ad=1.0}.*

5 Conclusion

In this article, we have defined the basic framework of a new subsystem in the MIEL system, the XML subsystem, in order to store the information retrieved from the Web in the field of predictive microbiology. The XML subsystem is composed of an XML knowledge base which had to take into account the specificities of the knowledge stored in the Sym'Previus knowledge base and had to be able to be queried by means of the MIEL query language, this in order to ensure its integration in the MIEL system. We have chosen to use the tree-based model in order to represent the XML knowledge base. We have proposed an extension of the tree-based model proposed in the Xyleme project (see [16]) in order to represent imprecise data expressed as possibility distribution in the XML knowledge base. We have also adapted the tree-based model in order to allow one to query the XML knowledge base by means of the MIEL query language. Consequently, we have proposed an XML querying language adding flexibility on the searched values in the leaves (selection criteria expressing preferences, preferences in extension for selection criteria defined on symbolic definition domains, fuzzy pattern matching to compare selection criteria to data). At the moment, the XML querying subsystem has been implemented in Java. We have used XML Schema to define the schemas (called type trees in this article) of the XML documents of the knowledge base, the schemas of the ontology and the schemas of the views. We have used XQuery to query the knowledge base and to obtain the answers. The knowledge base can be queried by a final user through an Internet browser.

As mentioned in the introduction, we make in this article the assumption that XML data fit XML schemas which are used in the querying process. We have evaluated this first version of our new XML subsystem by creating manually XML documents checking this hypothesis. Our next step will be to study ways to relax this assumption by introducing, for example, flexibility on the tree query structure or similarity relationships on tree labels. The design of this next version will be guided by the properties of the pertinent XML documents which are retrieved semi-automatically from the Web.

References

1. V. Aguiléra, S. Cluet, P. Vetri, D. Vodislav, and F. Wattez, *Querying the xml documents on the web*, Proceedings of the ACMSIGIR Workshop on XML and I.R. (Athens), July 2000.
2. S. Amer-Yahia, SungRan Cho, and Divesh Srivastava, *Tree pattern relaxation*, Proceedings of the 8th International Conference on Extending Database Technology (EDBT 2002), 2002.

3. G. Bordogna and G. Pasi, *A fuzzy object oriented data model managing vague and uncertain information*, International Journal of Intelligent Systems **14** (1999), no. 6, SCI 3495.
4. Gloria Bordogna and Gabriella Pasi, *Flexible querying of web documents*, Proceedings of the ACM Symposium Applied Computing (Madrid, Spain), 2002, pp. 675–680.
5. P. Bosc, L. Lietard, and O. Pivert, *Soft querying, a new feature for database management system*, Proceedings DEXA'94 (Database and EXpert system Application), Lecture Notes in Computer Science #856, Springer-Verlag, 1994, pp. 631–640.
6. Patrick Bosc, L. Lietard, and Olivier Pivert, *Fuzziness in Database Management Systems*, ch. Fuzzy theory techniques and applications in data-base management systems, pp. 666–671, Academic Press, 1999.
7. P. Buche and O. Haemmerlé, *Towards category-based fuzzy querying of both structured and semi-structured imprecise data*, Proceedings of the Fourth International Conference on Flexible Query Answering Systems (FQAS'2000) (Warsaw, Poland), Springer-Verlag, October 2000, pp. 362–375.
8. P. Buche, O. Haemmerlé, and R. Thomopoulos, *Integration of heterogeneous, imprecise and incomplete data: an application to the microbiological risk assessment*, Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems, ISMIS2003 (Maebashi, Japan), Lecture Notes in AI #2871, Springer, October 2003, pp. 98–107.
9. P. Ciaccia and W. Penzo, *Adding flexibility to structure similarity queries on xml data*, Proceedings of the Fifth International Conference on Flexible Query Answering Systems (FQAS'2002) (Copenhagen, Denmark), Lecture Notes in AI #2522, Springer, 2002, pp. 124–139.
10. H. Prade, *Lipski's approach to incomplete information data bases restated and generalized in the setting of Zadeh's possibility theory*, Information Systems **9** (1984), no. 1, 27–42.
11. T. Schlieder, *Similarity search in xml data using cost-based query transformations*, Proceedings of the Fourth International Workshop on the Web and Databases (WebDB 2001), 2001.
12. Torsten Schlieder, *Schema-driven evaluation of approximate tree-pattern queries*, Proceedings of the 8th International Conference on Extending Database Technology (EDBT 2002), 2002.
13. A. Theobald and G. Weikum, *Adding relevance to xml*, Proceedings of the Third International Workshop on the Web and Databases (WebDB 2000), 2000, pp. 35–40.
14. R. Thomopoulos, P. Buche, and O. Haemmerlé, *Different kinds of comparison between fuzzy conceptual graphs*, Proceedings of the 11th International Conference on Conceptual Structures, ICCS'2003, Lecture Notes in Artificial Intelligence 2746 (Dresden, Germany), Springer, July 2003, pp. 54–68.
15. Rallou Thomopoulos, P. Buche, and O. Haemmerlé, *Sous-ensembles flous définis sur une ontologie*, Revue des Nouvelles Technologies de l'Information E-2, Vol 1, EGC'2004, pp 147–158.
16. Lucie Xyleme, *A dynamic warehouse for xml data of the web*, IEEE Data Engineering Bulletin (2001).
17. L. Zadeh, *Fuzzy sets*, Information and control **8** (1965), 338–353.
18. Lofti Zadeh, *Fuzzy sets as a basis for a theory of possibility*, Fuzzy Sets and Systems **1** (1978), 3–28.