Fuzzy semantic tagging and flexible querying of XML documents extracted from the Web

Patrice Buche, Juliette Dibie-Barthelemy, Ollivier Haemmerlé, Gaëlle Hignette

Fuzzy semantic tagging and flexible querying of XML documents extracted from the Web

Patrice Buche (patrice.buche@inapg.fr)
INRA, Département Mathématiques et Informatique Appliquées, Unité Mét@risk, 16 rue Claude Bernard, F-75231 Paris Cedex 5

Juliette Dibie-Barthélemy (juliette.dibie@inapg.fr)
UMR INA P-G/INRA MIA, 16 rue Claude Bernard, F-75231 Paris Cedex 05

Ollivier Haemmerlé (ollivier.haemmerle@inapg.fr)
UMR INA P-G/INRA MIA, 16 rue Claude Bernard, F-75231 Paris Cedex 05
LRI (UMR CNRS 8623 - Université Paris-Sud) / INRIA (Futurs), Bâtiment 490, F-91405 Orsay Cedex

Gaëlle Hignette (gaelle.hignette@inapg.fr)
UMR INA P-G/INRA MIA, 16 rue Claude Bernard, F-75231 Paris Cedex 05

Abstract.
The relational database model is widely used in real applications. We propose a way of complementing such a database with an XML data warehouse. The approach we propose is generic, and driven by a domain ontology. The XML data warehouse is built from data extracted from the Web, which are semantically tagged using terms belonging to the domain ontology. The semantic tagging is fuzzy, since, instead of tagging the values of the Web document with one value of the domain ontology, we propose to use tags expressed in terms of a possibility distribution representing a set of possible terms, each term being weighted by a possibility degree. The querying of the XML data warehouse is also fuzzy: the end-users can express their preferences by means of fuzzy selection criteria. We present our approach on a first application domain: predictive microbiology.

Keywords: Flexible querying, semantic tagging, fuzzy data

1. Introduction

The relational database model has been widely studied since the 80's and it is now the most popular database model used in real applications because of its efficiency. In a large area of application domains, thematic relational databases have been developed and they often contain a great deal of reference data. A lot of those databases are built on the Open World Assumption, which means that the lack of answer does not imply that the answer is negative but rather unknown. The corollary of the Open World Assumption is the incompleteness issue, which has been widely studied. Palliating the incompleteness issue can be achieved in two main ways. The first one consists in enlarging the answers of a

query, for example by generalizing the query in order to give relevant answers when there is no exact answer to the query. The second one consists in complementing the database with data provided by external data sources. That solution can be particularly relevant when the incompleteness is due to the fact that the domain is quickly evolving, for example in a context of technological intelligence. In order to palliate such incompleteness by automatically complementing databases, the World Wide Web appears to be an interesting data source.

The work we present in this article takes place in the context of the construction of thematic data warehouses. More precisely, we are interested in integrating data extracted from the Web into an existing thematic relational database, that integration being guided by an existing domain ontology. That ontology is composed of (i) a taxonomy of terms hierarchized by the *kind-of* relation, (ii) a set of semantic relations which correspond to the relations of the relational database schema. Our approach is generic, since changing the domain ontology is sufficient to change the application domain.

The data which are meant to automatically feed the data warehouse are extracted from the Web; the variety of such data leads to a problem of heterogeneity of these data. The choice we made in our approach was to build a data warehouse expressed in XML and to integrate it into the pre-existing relational database by means of the query processing. That query processing is done through a uniform query language, which is very simple and allows the end-users to ask their queries on both bases in a completely transparent way. We chose to use an XML data warehouse since we think that in the near future, a lot of Web documents will be expressed in XML.

Nevertheless, for the moment, in most application domains, very few Web XML documents are available. Thus we chose to translate the Web documents found in different formats into XML. Our process of an automatic construction of an XML thematic data warehouse is implemented and is called AQWEB. It relies on the following three steps which are schematized in figure 1. (i) A crawler and a filter, using the domain ontology, return relevant documents for the application domain. We focus on documents which contain data tables, and the most important part of our treatments concerns those data tables. This may be seen as a restriction of our approach. But, in a lot of application domains, especially in the scientific field, data tables are often a source of relevant, reliable and synthetic data. Moreover, their tabular structure is obviously easier to automatically parse than rough natural language. (ii) The Web documents in html, doc, or most usually in pdf, are translated into a generic XML format we call XTab, which allows the representation of data tables in a classical and generic way

– a table is a set of lines, each line being a set of cells. (iii) The data tables are fuzzy semantically tagged during the "XTab2SML" process. That process consists in associating several potential terms from the domain ontology, weighted by a possibility degree, with the terms used in the Web data tables, then identifying semantic relations between the columns of the tables. The use of terms and relations belonging to the domain ontology during the fuzzy semantic tagging is motivated by the fact that the thematic data warehouse and the relational database are queried through a single query processor, the MIEL++ system, which relies on the domain ontology. Another specificity of the MIEL++ system is that it allows the end-users to express fuzzy selection criteria in their queries. Since the XML data generated by AQWEB are tagged semantically by means of fuzzy sets, we introduce a way of querying fuzzy data with fuzzy selection criteria.

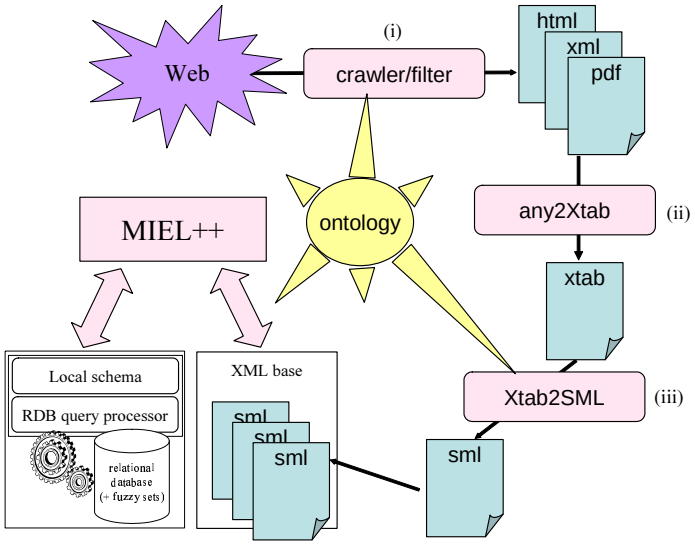

Figure 1. Architecture of the AQWEB system.

The work presented in this article is applied to a scientific domain: predictive microbiology. Since 1999, our team has been working on several projects which concern that field. In the Sym'Previus project, we propose techniques allowing data concerning the behaviour of pathogenic germs in food products to be represented. Those data are designed to be used in a tool for microbiologists in order to help them prevent the risk of food product contamination. In Sym'Previus, we essentially developed a relational database, which allows the representation of fuzzy data and proposes a flexible query processing. Since 2003, our team has been involved in a national project called

e.dot[1]. That project aims at building a thematic data warehouse on the microbial risk in food products. This data warehouse is filled with XML documents automatically extracted from the Web and that XML data warehouse is integrated with the existing relational database.

This article is structured as follows. Section 2 presents the background we need in order to integrate an existing relational database and an XML data warehouse. Section 3 presents fuzzy semantic tagging. Section 4 introduces our XML data warehouse. Section 5 presents the query processing of the data warehouse. Section 6 compares our work to related works.

2. Background

In this section, we briefly present the ontology used by the AQWEB system, the fuzzy set framework and the MIEL query language.

2.1. The ontology

The ontology contains the terminological knowledge of the application domain. It is notably composed of a taxonomy of terms and a relational schema. The taxonomy of terms is composed of the set of attributes which can be queried on by the end-user and their corresponding definition domains. Each attribute is defined on a definition domain which can be numeric (it is then completely ordered) or hierarchized symbolic (it is then partially ordered by the *kind-of* relation).

EXAMPLE 1. *Figure 2 is a part of a taxonomy composed of the attribute Product and its hierarchized symbolic definition domain.*

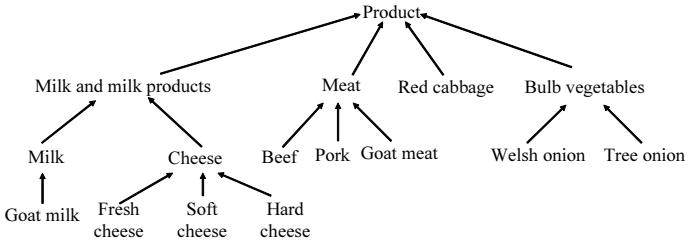

Figure 2. A part of the taxonomy corresponding to the attribute Product.

[1] entrepôts de données ouverts sur la toile, i.e. data warehouses opened on the Web

The relational schema of the ontology corresponds to the relational schema of the relational database of the MIEL++ system. It is composed of a set of signatures (i.e. the types of their attributes) of the possible relations between terms of the taxonomy.

EXAMPLE 2. *The relation FoodProductpH is used to link a food product and its associated pH value.*

2.2. Fuzzy set theory

In this article we use the representation of fuzzy sets proposed in (Zadeh, 1965; Zadeh, 1978).

DEFINITION 1. A *fuzzy set* f on a definition domain $\mathcal{D}(f)$ is defined by a membership function μ_f from $\mathcal{D}(f)$ to $[0,1]$ that associates the degree to which x belongs to f with each element x of $\mathcal{D}(f)$. We call *support* of f the subset of $\mathcal{D}(f)$ such that $support(f) = \{a \in \mathcal{D}(f) \mid \mu_f(a) > 0\}$. We call *kernel* of f the subset of $\mathcal{D}(f)$ such that $kernel(f) = \{a \in \mathcal{D}(f) \mid \mu_f(a) = 1\}$.

The fuzzy set formalism can be used in two different ways: (i) in the database, in order to represent imprecise data expressed in terms of possibility distributions or (ii) in the queries, in order to represent fuzzy selection criteria which express the preferences of the end-user. A fuzzy set can be defined on a continuous or discrete definition domain. For fuzzy sets defined on a continuous domain, we make the assumption that they are always represented as a trapezoidal form.

EXAMPLE 3. *The fuzzy set pHPreference of figure 3 is a continuous fuzzy set noted [4,5,6,7] and the fuzzy set ProductPreference is a discrete one noted (1/Bulb vegetable + 0.5/Red cabbage).*

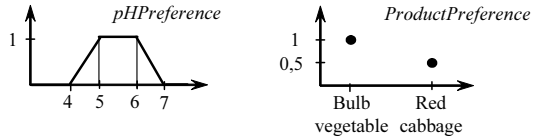

Figure 3. Two fuzzy sets.

Two scalar measures are classically used in the fuzzy set theory to evaluate the compatibility between a fuzzy selection criterion and an imprecise datum: (i) a possibility degree of matching (Zadeh, 1978) and (ii) a necessity degree of matching (Dubois and Prade, 1988).

DEFINITION 2. Let f and g be two fuzzy sets defined on the same definition domain $\mathcal{D}$, 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 $\Pi(f,g) = sup_{x\in\mathcal{D}}(min(\mu_f(x),\ \mu_g(x)))$ and the *necessity degree of matching* is $N(f,g) = 1 - sup_{x\in\mathcal{D}}(min(1-\mu_f(x),\ \mu_g(x)))$.

2.3. The MIEL query language

In the MIEL++ system, the query processing is done through the MIEL query language. This query processing relies on a set of pre-written queries, called *views*, which are given to help the end-users to express their queries. We introduce the MIEL query language by presenting successively the views, the queries and the answers to a query.

2.3.1. *The views*

The views are the way given to the end-user to query the bases integrated by the MIEL++ system, without having to know the complexity of the schemas of the bases. In the MIEL query language, 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.

EXAMPLE 4. *The view FoodProductpHView is defined as follows:* $\{$ *Product, pH* $\mid P_{FoodP.pH}$*(Product, pH)* $\}$ *where the attributes Product and pH are the queryable attributes and the predicate* $P_{FoodP.pH}$ *describes the way the attributes of the view are linked together. This view allows one to know the pH value of a food product.*

2.3.2. *The queries*

A query is built by the end-user by specifying among the set of queryable attributes of a given view which are the selection attributes and their corresponding searched values and which are the projection attributes.

DEFINITION 3. A *query* Q asked on a view V defined on n attributes $\{a_1,\ldots,a_n\}$ is defined by $Q = \{V, S, C\}$ where $S \subseteq \{a_1,\ldots,a_n\}$ represents the set of the projection attributes and where $C = \{c_1,\ldots,c_m\}$ is the set of selection criteria. Each selection criterion c_i is restricted to an equality $< a_i = v_i >$ between an attribute $a_i \in \{a_1,\ldots,a_n\}$ and its corresponding searched value v_i which can be crisp or fuzzy and must be defined on a subset of the definition domain of a_i.

EXAMPLE 5. *One can build the following query from the view FoodProductpHView of example 4:* { *FoodProductpHView, pH , <Product = Bulb vegetable>* }. *That query means that the end-user wants to get the pH value of the food product "bulb vegetable".*

When the fuzzy value of a selection attribute has a hierarchized symbolic definition domain, the fuzzy set used to represent the fuzzy value can be defined on a subset of this definition domain. We consider that such a fuzzy set defines degrees implicitly on the whole definition domain of the selection attribute. In order to take those implicit degrees into account, the *fuzzy set closure* has been defined in (Buche et al., 2005). The fuzzy set closure is systematically used when a comparison involves two fuzzy sets (an expression of end-users' preferences and an imprecise datum) defined on a hierarchical definition domain.

EXAMPLE 6. *Let us consider the discrete fuzzy value (1/Bulb vegetable + 0.5/Red cabbage) assigned by the end-user to the selection criterion Product. It can be interpreted as "the end-user wants a bulb vegetable as a Product, but he/she also accepts red cabbage with a lower interest". Since the selection criterion Product has a hierarchized symbolic definition domain (see figure 2), we consider that the end-user who is interested in a bulb vegetable is also interested in all kinds of bulb vegetables. The fuzzy set closure given in figure 4 results from the propagation of the degree associated with a value to its specializations.*

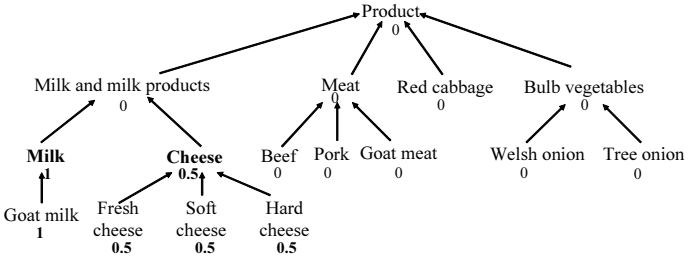

Figure 4. Fuzzy set closure.

2.3.3. *The answers*

An answer to a query Q (i) satisfies all the selection criteria of Q in the meaning of definition 4 given below and (ii) associates a constant value with each projection attribute of Q.

DEFINITION 4. 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 with the possibility degree $\Pi(cl(v), cl(v'))$ and the necessity degree $N(cl(v), cl(v'))$ in the meaning of definition 2 where the cl function corresponds to the fuzzy set closure.

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

DEFINITION 5. An *answer* A to a query $Q = \{V, S, C\}$ is a set of tuples, each of the form $\{v_1, \ldots, v_l, ad_\Pi, ad_N\}$, where $v_1, \ldots, v_l$ correspond to the crisp or fuzzy values associated with each projection attribute $a_i \in S$ of Q, where all the selection criteria $c_1, \ldots, c_m$ of Q are satisfied with the possibility degrees (resp. necessity degrees) $\Pi_1, \ldots, \Pi_m$ (resp. $N_1, \ldots, N_m$), and where ad_Π is the possibility degree (resp. ad_N is the necessity degree) of the answer A to the query Q defined as follows: $ad_\Pi = min_{i=1}^{m}(\Pi_i)$ (resp. $ad_N = min_{i=1}^{m}(N_i)$).

3. The fuzzy semantic tagging

This section deals with the fuzzy semantic tagging of the data tables extracted from the Web according to a given domain ontology.

3.1. The Semantic Markup Language (SML)

We propose to represent the semantically tagged data tables by means of the SML (Semantic Markup Language) format as defined in (Saïs et al., 2005). Each data table extracted from the Web is enriched with terms of the ontology and stored in an SML document. The semantic tagging consists in: (i) finding in the ontology the *final values* which are the terms that are close to the ones in the Web data table; when no exact identification is possible, inclusion or intersection of words is used; (ii) finding which relations of the ontology are represented in the table, using comparison between the signature of the relations and the types of the columns of the table (identified through the values they contain). Thus, in an SML document, the lines are structured in relations, each relation involving the set of identified columns of the corresponding Web data table. Each cell of the Web data table is represented in the SML document by its original value and its corresponding final values.

EXAMPLE 7. *Figure 5 presents a Web data table and the data table obtained by semantic tagging. The first column has been identified as having the type Product and the second one as the type pH, which correspond to the signature of the relation FoodProductpH (see example 2). Figure 6 gives the SML representation of the Web data table of figure 5.*

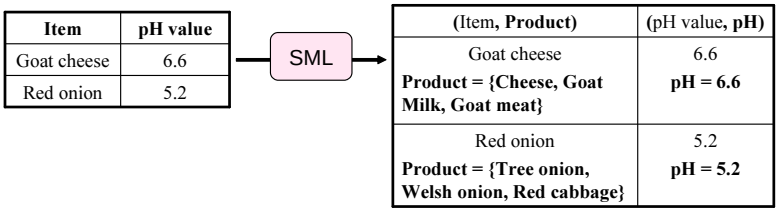

Figure 5. A Web data table and its semantic tagging.

```
<table> <title><table-title> </table-title>
<title-col> Item </title-col> <title-col> pH value </title-col>... </title>
<content>
...
<relLine> <FoodProductPH>
<Product><originalVal>Red onion</originalVal>
<finalVal>Tree onion</finalVal>
<finalVal>Welsh onion</finalVal>
<finalVal>Red cabbage</finalVal>
</Product>
<ph> <originalVal>5.2</originalVal> <finalVal/></ph>
</FoodProductPH> </relLine>
</content> </origine> </table>
```

Figure 6. Simplified representation in SML of the Web data table of figure 5

3.2. A relevance score for fuzzy annotations

In an SML document, several terms can be found to be close to an original value, but not all these terms represent the meaning of the original value equally well. We propose a fuzzy semantic tagging of the data table, which maps each original value with terms belonging to the ontology, ordered according to their relevance to the original value.

After some tests with classic similarity measures (Egghe and Michel, 2002), we propose to use the *cosine* similarity measure in order to compute the relevance score between an original value and a term from the ontology, based on the distinction between important and secondary words in the terms of the ontology (Hignette et al., 2005).

Terms of the ontology are usually composed of several words, such as *Welsh onion*. But all the words do not have the same semantic power in a term (*onion* contains more semantics than *Welsh* to identify the food product). We propose to give a weight between 0 and 1 to each word in each term of the ontology. Non-meaning words (such as *and*, *with*, *the*...) have a weight of 0, the most meaningful words are given a weight of 1, and words carrying less semantics are given a weight in-between. We represent each term of the ontology as a weighted vector over all possible words (limited to the words of the terms of the ontology

Table I. Relevance score between a term from the Web and terms from the ontology

terms and word weights		relevance score	
from the Web	from the ontology	computing	value
$\underbrace{\text{red}}_{1}$ $\underbrace{\text{onion}}_{1}$	$\underbrace{\text{Welsh}}_{0.2}$ $\underbrace{\text{onion}}_{1}$	$\frac{1\times1}{\sqrt{(1^2+1^2)\times(0.2^2+1^2)}}$	0.69
	$\underbrace{\text{tree}}_{0.2}$ $\underbrace{\text{onion}}_{1}$	$\frac{1\times1}{\sqrt{(1^2+1^2)\times(0.2^2+1^2)}}$	0.69
	$\underbrace{\text{red}}_{0.2}$ $\underbrace{\text{cabbage}}_{1}$	$\frac{1\times0.2}{\sqrt{(1^2+1^2)\times(0.2^2+1^2)}}$	0.14

and the words of the studied term found on the Web), a weight of 0 being given to words that do not appear in the term.

Terms found on the Web are also represented as weighted vectors over all possible words: a weight of 1 is given to all words that appear in the term. For each term of the Web, we can now compute its cosine similarity measure with each term of the ontology.

DEFINITION 6. *Let $T = (w_1, \ldots, w_n)$ be a term from the Web and let $T' = (w'_1, \ldots, w'_n)$ be a term from the ontology. The relevance score between T and T' is defined as their cosine similarity measure:*

$$relevance(T, T') = \frac{\sum_{k=1}^{n} w_k w'_k}{\sqrt{\sum_{k=1}^{n} {w_k}^2 \times \sum_{k=1}^{n} {w'_k}^2}}$$

EXAMPLE 8. *Table I shows how to compute the relevance score between the term from the Web,* red onion*, and its corresponding terms found in the ontology.*

Thus, for a given value of a Web data table (which appears as *originalVal* tag), we propose to replace the set of its associated terms belonging to the ontology (which appears as a set of *finalVal* tags with values $\{t_1, \ldots, t_k\}$), by a discrete fuzzy set defined on $\{t_1, \ldots, t_k\}$ and whose membership function μ is the relevance score between the value of the table and each term in $\{t_1, \ldots, t_k\}$; the fuzzy set is proportionally normalized so that the degree 1 is associated with the terms having the best relevance score. The fuzzy set we use can be interpreted in the classical meaning of a normalized possibility distribution, which represents an exclusive weighted disjunction of possible values.

We evaluated our approach by an experiment: 186 terms representing food products were taken from tables in publications on food microbiology, and annotated manually using the Codex Alimentarius, an ontology on food products used by the World Health Organization.

Word weighting on the ontology was performed by a human expert, (weights of 1 for important words and 0.2 for minor words) and automatic annotation was launched on the 186 terms. When ordering the annotations according to the relevance score, we find the *best match* within the first 3 positions for 94 terms. The best match is the term of the ontology chosen by manual annotation; we assume that in the list of terms of the ontology, there exists only one best match to represent the original value. We re-launched the experiment with no distinction between major and minor words (all words having a weight of 1): the best match appeared in a lower rank for 23 terms. So, distinction between major and minor words allows better annotation.

4. The XML data warehouse

The XML data warehouse is composed of SML documents which represent Web data tables fuzzy semantically tagged thanks to a domain ontology. In the following, we use the tree-based model as proposed in (Aguiléra et al., 2000; Xyleme, 2001) in order to represent the XML data warehouse. First, we briefly recall the definitions of the tree-based model. Second, we propose a way of representing imprecise data using the fuzzy set formalism in the tree-based model. Third, we define the XML data warehouse which contains imprecise data.

4.1. Preliminary notions: the tree-based model

In the tree-based model, an XML data warehouse 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 (i) h preserves the root of t: $\text{root}(t') = h(\text{root}(t))$ and (ii) 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 data warehouse is defined by the set of type trees which are associated with the data trees that it contains.

4.2. Representation of fuzzy values

In the tree-based model, we propose to represent continuous and discrete fuzzy sets by means of data trees.

DEFINITION 9. Let f be a continuous fuzzy set. f is represented by a data tree which is composed of (1) a root labelled *CFS* and (2) 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 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 $\mathcal{D}(f)$, there exists a node labelled *ValF* that has two children labelled *Item* and *MD* (for Membership Degree) of respective values x and $\mu(x)$.

EXAMPLE 9. *Figure 7 gives the data trees representing the continuous and discrete fuzzy sets pHPreference and ProductPreference of figure 3.*

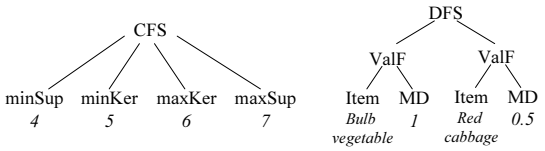

Figure 7. Example of data trees representing continuous and discrete fuzzy sets.

4.3. The XML data warehouse in the tree-based model

The XML data warehouse is a set of SML documents containing fuzzy values. We propose to model SML documents as fuzzy data which are data trees that allow fuzzy values to be represented. In a fuzzy data tree, the partial value function v (Cf. definition 7) can assign a crisp or a fuzzy value to a node, which is then called *crisp* or *fuzzy value node*.

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 the 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 9 and 10.

EXAMPLE 10. *Figure 8 gives an example of a fuzzy data tree corresponding to a part of the SML document of figure 6, the relevance scores being obtained after normalization of those presented in table I.* originalVal *is a crisp value node,* finalVal *is a fuzzy value node.*

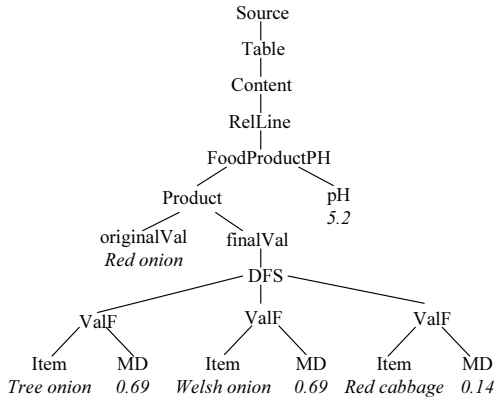

Figure 8. A fuzzy data tree.

5. The query processing of the XML data warehouse

The XML data warehouse presented in the previous section has been built in order to integrate data extracted from the Web with an existing thematic relational database. That integration is done by means of the MIEL++ querying system, which scans both bases by using the same ontology. This uniform querying is possible thanks to the semantic tagging of SML documents with terms and relations of the ontology. This section presents the uniform interrogation of the imprecise XML data warehouse and the relational database by means of the MIEL query language. We define what the notions of ontology, views, queries and answers become in the XML subsystem of the MIEL++ system.

5.1. The ontology

The ontology defined in section 2.1 is represented in the XML subsystem as a tree stored in an XML document. This ontology is a replication of the ontology of the MIEL++ relational subsystem.

5.2. The views

The XML subsystem relies on a set of views, which are built from the terms and the relations of the ontology and allow one to query the XML data warehouse.

DEFINITION 12. 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 mark ql to crisp and fuzzy value nodes of t_V, which are then queryable.

EXAMPLE 11. *Figure 9 (the left side) shows a view using the relation FoodProductPH involving three queryable attributes: the* finalVal *of the product which is a fuzzy value node, the* originalVal *of the product and the* originalVal *of the pH which are both crisp value nodes.*

5.3. The queries

A query is built from 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.

DEFINITION 13. A *query* that conforms 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 that conforms to (t_T, l_T);
- p_Q is a partial function that assigns the mark pl to the queryable value nodes of the view, which are considered as the *projection value nodes*;
- s_Q is a partial function that assigns the mark sl to the queryable value nodes of the view, which are considered as the *selection value nodes*, also called selection criteria;
- ws_Q is a partial value function that assigns a value to the selection value nodes of the query, such that the value assigned to a crisp value node is an atomic value and the value assigned to a fuzzy value node is a data tree with a root labelled CFS or DFS which respectively conforms to definitions 9 and 10.

As defined in definition 3, the value v of a selection criterion $< a = v >$, a being a value node of the query, must be defined on a subset of the definition domain of a. This value is given by the end-user and can be crisp or fuzzy. In the second case, a fuzzy set is used to represent a fuzzy selection criterion which expresses the end-user's preferences.

EXAMPLE 12. *The query Q of figure 9 (right side) expresses that the end-user wants to obtain the product (originalVal and finalVal) and the pH value from the view using the relation FoodProductpH. The fuzzy value assigned to the selection criterion Product can be interpreted as "the user wants a bulb vegetable as a product, but he/she also accepts red cabbage with a lower interest".*

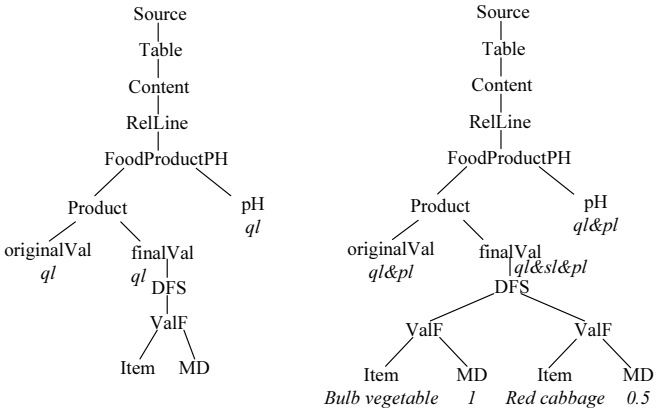

Figure 9. An example of view using the relation *FoodProductPH* and a query expressed in that view.

5.4. The answers

An answer to a query Q (i) satisfies all the selection criteria of Q in the meaning of definition 4 and (ii) associates a constant value with each projection leaf of Q. The search for the answers to a query in an XML data warehouse is done through the valuation of the query on the data trees of the data warehouse as defined below.

DEFINITION 14. Let Q=(t_Q, l_Q, w_Q, p_Q, s_Q, ws_Q) be a query conforming 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 (i) σ_D is a strict type homomorphism from (t_Q, l_Q) into (t_D, l_D) and (ii) σ_D satisfies each selection criterion n_s^i, $i \in [1, m]$, of Q with the possibility degree $\Pi(ws_Q(n_s^i), v_D(\sigma_D(n_s^i)))$ and the necessity degree $N(ws_Q(n_s^i), v_D(\sigma_D(n_s^i)))$. The adequation degrees of the data tree D to the query Q through the valuation σ_D are $ad_{\Pi(D)}$=$min_{i\in[1,m]}(\Pi(ws_Q(n_s^i), v_D(\sigma_D(n_s^i))))$ and $ad_{N(D)}$=$min_{i\in[1,m]}(N(ws_Q(n_s^i), v_D(\sigma_D(n_s^i))))$.

An answer to a query in the XML data warehouse is a set of tuples, each tuple being a set of values given to each projection node.

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 data warehouse $\mathcal{W}$ is a set of tuples, each tuple being defined as follows: $\{ \cup_{i=1}^{m} v_D(\sigma_D(n_p^i)) \cup ad_{\Pi(D)} \cup ad_{N(D)} \mid D$ is a data tree of $\mathcal{W}$ and σ_D is a valuation of Q w.r.t. $D\}$.

REMARK 1. *When a fuzzy set is used to represent a fuzzy selection criterion defined on a hierarchized symbolic definition domain, the fuzzy set closure is computed and used to search for satisfying answers.*

EXAMPLE 13. *The answer to the query Q of figure 9 in the SML document of figure 8 is the following:* { *Red onion, (1.0/Tree onion + 1.0/Welsh onion + 0.20/Red cabbage), 5.2,* ad_{Π}*=1.0,* ad_{N}*=0.8* }. *To compute this answer, we have used the fuzzy set closure (1/Bulb vegetable + 0.5/Red cabbage) given in figure 4 and the fuzzy set closure (1.0/Tree onion + 1.0/Welsh onion + 0.20/Red cabbage).*

6. Related works

In our work, Web tables are indexed thanks to a fuzzy semantic tagging with terms and semantic relations of the ontology which are used in the fuzzy querying. As a consequence, our approach must be compared to two types of works: fuzzy database systems where semantic relations and terms are used in the queries and fuzzy information retrieval systems where only terms are used in the queries.

In the first category of works, the fuzzy set framework has been shown to be a sound scientific choice for modeling flexible queries (Bosc et al., 1994; Bosc and Pivert, 1995). It is a natural way of representing the notion of preference using a gradual scale. The fuzzy set framework has also been proposed to represent imprecise values by means of possibility distributions (Zadeh, 1978). Several authors have developed this approach in the context of databases (Prade, 1984), especially in the framework of the relational database model (Bosc et al., 1999) and object-oriented database model (Bordogna and Pasi, 1999). We have proposed in this paper to adapt this approach to the tree-based model we use (Aguiléra et al., 2000; Xyleme, 2001) to modelize an XML database. First, we define XML queries including fuzzy sets representing end-user's preferences. Second, we define the way an imprecise datum is represented in an XML data tree and how this data tree is compared to an XML fuzzy query. Third, in order to become comparable, fuzzy sets representing end-users' preferences and imprecise data are transformed using the fuzzy set closure. The fuzzy set closure allows the enlargement of the querying to terms of the ontology which are more specific than those specified in the original fuzzy sets. To the best of our knowledge, those contributions are original in the framework of XML databases.

In the second category of works, fuzzy information retrieval techniques have been proposed: (i) to index documents, (ii) to query the

documents using indexed terms, and (iii) to build fuzzy associations between terms (Bordogna and Pasi, 2001). The fuzzy indexation of a document is represented as a fuzzy set defined on the domain of the index terms. The membership degree of a given index term represents the relevance of the term in the document. In general, it is based on the occurrence count of the term in the document and in the whole set of documents (Salton and Buckley, 1988; Spark Jones, 1972). As the granularity level of this indexation is the whole document, it does not take into account the fact that a term can play a different role in different sections of the document. (Bordogna and Pasi, 2001; Bordogna and Pasi, 2002) proposes a more refined fuzzy indexation of a document represented as a fuzzy binary relation on the Cartesian product $T \times S$ (T the set of index terms and S the set of sections of the document). With each pair (term, section), a significance degree of the term in the section is computed. In fuzzy information retrieval querying languages, queries may consist of two types of components. The first one concerns atomic selection conditions which are expressed as pairs (term, weight) in which weight in [0, 1] indicates a soft constraint. The second one is constituted by soft aggregation operators which permit one to obtain a unique relevance score of the document compared to the set of atomic selection conditions expressed in the query. The concept of Ordered Weight Averaging (Yager, 1988) associated with linguistic quantifiers (for example, *most of, all, at least one, ...*) (Zadeh, 1983) provides a suitable framework to build soft aggregation operators. The result of a query is a fuzzy set defined on the whole set of documents where the membership degree corresponds to the relevance score of the document compared to the query. Fuzzy associations between terms may be computed using fuzzy thesaurii and fuzzy pseudo-thesaurii to serve two purposes: (i) to expand the set of index terms of a document with associated terms also present in it, (ii) to expand each of the terms sought in a query to associated terms. (Miyamoto, 1990) proposed to build a similarity relation between terms thanks to a set of concepts C which permits each term to be described. A term is associated with a fuzzy set defined on C in which the membership degree reveals the degree to which the term is related to a given concept $c \in C$. A fuzzy pseudo-thesaurus can be defined by replacing the set of concepts C with a given set of documents. The fuzzy set defined on C for each term t is replaced by the fuzzy set of documents indexed by t (De Cock et al., 2004). Then the computation of the similarity between terms is based on the cooccurrence frequency of the terms in a given set of documents.

A direct use of fuzzy information retrieval techniques is rather difficult in our approach for two reasons: (i) all those techniques are based on the terms which are present in a given set of documents, (ii) the

relevance of a term in a document is based on its occurrence count. In our approach, we want to index the Web tables with terms of the ontology. First, the number of terms in a given Web table is small compared to a whole document. Therefore, a relevance degree based on the occurences count of a given term should not be significant in this context. Second, to apply those techniques, it requires a set of documents in which the Web table terms to be indexed and the terms of the ontology are all present together. This is a strong constraint which is difficult to satisfy in practice. This is the reason why we have based our approach on more classic information retrieval techniques which permit similarity to be built between terms based on the comparison of strings (Salton and Gill, 1987; Boyce et al., 1995; Lin, 1998). Compared to the bibliography, we have proposed one improvement (a syntactic relevance score using a weight function which associates an importance factor to each word of the term) which permits one to obtain more best matches which are well-ranked, as discussed in section 3.2.

7. Conclusion

This paper has dealt with the completion of an existing thematic relational database by means of an XML data warehouse automatically filled with data extracted from the Web. Even if our approach has been applied in a scientific domain, i.e. predictive microbiology, we think that it is domain-independent and generic. As a matter of fact, the main asset of our approach is that it is based on an existing ontology, composed of a term taxonomy and a set of semantic relations defined according to the relational database schema. Those relations are used to automatically build the SML DTD which is used to reformat the documents found on the Web. Moreover, those documents are annotated by means of terms belonging to the ontology in order to allow a uniform query processing of both bases.

At the moment, our approach is restricted to documents containing data tables. That seems to be a limitation of our approach but it is an original treatment, and it gives promising results. In the future, we do not exclude the idea of developing methods allowing one to address the whole content of the Web documents. Thus it will no longer be possible to assume that the structure of the document is fixed. It would be interesting to study ways of relaxing this assumption, for example by introducing flexibility on the tree query structure and/or similarity relationships on tree labels.

The second originality of our work is that we propose a fuzzy semantic tagging. That tagging is not limited to associating a term of

the ontology with the values of the data tables we parse, but a set of potential terms weighted with a possibility degree. That fuzzy semantic tagging allows us to keep traceability of the term matchings done by our method. This allows us to propose an enlarged query processing. But, for the moment, the fuzzy tagging we propose only represents the fuzzy correspondance between a value found in the document and the terms of the domain ontology. In this paper, we assume that, in the semantic tagging, the identification of relations belonging to the ontology must be complete. This means that the whole signature of the relation must match the line of the table. This is a strong assumption. In a future work, we will study ways of representing a fuzzy correspondance between one line of a table and several relations of the ontology.

The third originality is the flexible querying system of the XML data warehouse we propose, with selection criteria expressed as fuzzy sets. This provides the end-user with the closest answers to the selection criteria in addition to the exact answers.

Finally, our approach has been fully implemented in Java in the AQWEB system. We have used a subset of XQuery to query the base through an Internet browser using the MIEL++ querying system. The AQWEB system has been validated by experts in microbiology during the French e.dot project.

We now have to prospect other application domains in order to validate the genericity of our approach.

References

Aguiléra, V., S. Cluet, P. Vetri, D. Vodislav, and F. Wattez: 2000, 'Querying the XML Documents on the Web'. In: *Proceedings of the ACMSIGIR Workshop on XML and I.R.* Athens.

Bordogna, G. and G. Pasi: 1999, 'A Fuzzy Object Oriented Data Model Managing Vague and Uncertain Information'. *International Journal of Intelligent Systems* **14**(6), SCI 3495.

Bordogna, G. and G. Pasi: 2001, 'Modeling vagueness in Information Retrieval'. In: *Proceedings of ESSIR 2000, Lecture Notes in Computer Science #1980.* pp. 207–241.

Bordogna, G. and G. Pasi: 2002, 'Flexible querying of WEB documents'. In: *Proceedings of the ACM Symposium Applied Computing.* Madrid, Spain, pp. 675–680.

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

Bosc, P., L. Lietard, and O. Pivert: 1999, *Fuzziness in Database Management Systems*, Chapt. Fuzzy theory techniques and applications in data-base management systems, pp. 666–671. Academic Press.

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

Boyce, B. R., C. T. Meadow, and D. H. Kraft: 1995, *Measurement in information science*. New York: Academic Press.

Buche, P., C. Dervin, O. Haemmerlé, and R. Thomopoulos: 2005, '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.

De Cock, M., G. S., and M. Nikravesh: 2004, 'Fuzzy thesauri for and from the www'. *In Nikravesh, M., Zadeh, L., Kacprzyk, J., eds.: Soft Computing for Information Processing and Analysis* pp. 275–284.

Dubois, D. and H. Prade: 1988, *Possibility theory- An approach to computerized processing of uncertainty*. Plenum Press, New York.

Egghe, L. and C. Michel: 2002, 'Strong similarity measures for ordered sets of documents in information retrieval'. *Information Processing and Management* **38**, 823–848.

Hignette, G., P. Buche, J. Dibie-Barthélemy, and O. Haemmerlé: 2005, 'Fuzzy semantic annotation of XML documents'. In: E. T. Jaelson Castro (ed.): *Proceedings of CAiSE'05 Workshops. The 17th conference on advanced information systems engineering, DisWeb'05*. Porto, Portugal, pp. 319–332, FEUP edicoes.

Lin, D.: 1998, 'An Information-Theoretic Definition of Similarity'. In: *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA, USA, pp. 296–304, Morgan Kaufmann Publishers Inc.

Miyamoto, S.: 1990, 'Information retrieval based on fuzzy associations'. *Fuzzy Sets and Systems* **38**, 191–205.

Prade, H.: 1984, '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.

Salton, G. and C. Buckley: 1988, 'Term weighting approaches in automatic text retrieval'. *Information processing and Management* **24**(5), 513–523.

Salton, G. and M. J. M. Gill: 1987, *Introduction to modern information retrieval*. New York: Mc Graw-Hill.

Saïs, F., H. Gagliardi, O. Haemmerlé, and N. Pernelle: 2005, 'Enrichissement sémantique de documents SML représentant des tableaux'. In: *Actes des 5èmes journées Extraction et Gestion des Connaissances, EGC'2005 (to appear)*. Paris, France.

Spark Jones, K. A.: 1972, 'A statistical interpretation of term specificity and its application in retrieval'. *Journal of documentation* **28**(1), 11–20.

Xyleme, L.: 2001, 'A dynamic warehouse for xml data of the web'. *IEEE Data Engineering Bulletin*.

Yager, R.: 1988, 'On ordered weighted averaging aggregation operators in multicriteria decision making'. *IEEE Transactions on Systems, Man and Cybernetics* **18**(1), 183–190.

Zadeh, L.: 1965, 'Fuzzy sets'. *Information and control* **8**, 338–353.

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

Zadeh, L. A.: 1983, 'A computational approach to fuzzy quantifiers in natural languages'. *Computing and mathematics with applications* **9**, 149–184.