HAL
archives-ouvertes.fr

Bipolarity in Flexible Querying of Information Systems Dedicated to Multimodal Transport Networks

Nouredine Tamani, Ludovic Lietard, Daniel Rocacher

To cite this version:

Nouredine Tamani, Ludovic Lietard, Daniel Rocacher. Bipolarity in Flexible Querying of Information Systems Dedicated to Multimodal Transport Networks. The 10th International Symposium on Programming and Systems (ISPS'11), Apr 2011, Algeria. pp.108-115, 2011.

HAL Id: hal-00657295

https://hal.archives-ouvertes.fr/hal-00657295

Submitted on 6 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bipolarity in Flexible Querying of Information Systems Dedicated to Multimodal Transport Networks

Nouredine Tamani
IRISA/ENSSAT/Univ. Rennes 1
6, rue de Kerampont, BP 80518,
22305 Lannion Cedex, France
Email: tamani@enssat.fr

Ludovic Liétard
IRISA/IUT/Univ. Rennes 1
rue Edouard Branly, BP 30219
22302 Lannion Cedex, France
Email: ludovic.lietard@univ-rennes1.fr

Daniel Rocacher
IRISA/ENSSAT/Univ. Rennes 1
6, rue de Kerampont, BP 80518,
22305 Lannion Cedex, France
Email: rocacher@enssat.fr

***Abstract*—Flexible querying of information systems dedicated to multimodal transport networks aims to help user organizing his/her trips by promoting public transport networks (bus, subway, train, boat, plane, etc.). In this context, it is necessary to provide an integrated environment in which it is possible for the user to express queries with complex preferences so as to meet his/her expectations. Complex preferences are modeled by fuzzy bipolar conditions which associate negative and positive conditions. Queries involving such conditions are called bipolar queries. In our case, such queries are addressed to multimodal transport information systems, which are often made of several distributed and heterogeneous databases. Therefore, semantic aspects have to be taken into consideration in the querying process so that only the most relevant data is targeted to evaluate queries. We introduce then in this paper a new approach for flexible querying of information systems that combines a reasoning mechanism (fuzzy bipolar DLR-Lite) with a relational language of a high expressiveness (bipolar SQLf language).**

***Index Terms*—Flexible querying, complex preferences, fuzzy bipolar conditions, bipolar SQLf, fuzzy sets theory, fuzzy description logics, multimodal transportation.**

I. Introduction and Motivation

Multimodal transport information system is intended to provide information called multimodal transport information (denoted MTI) which is about various modes and available transport networks and services to help travelers to plan and to organize their trips [17]. It covers information about transport modes, timetables, pricing, traffic analysis, etc.

User oriented services are developed to promote public transport networks for the planning of user's daily journeys. To fulfill user requirements, multimodal transport information systems must be endowed with powerful and complex functionalities. The issues of the heterogeneity and distribution of data in the transport field have also to be taken into consideration. Ontologies could be used to manage these aspects because they not only unify the domain vocabulary but they also provide reasoning mechanisms capable to extract implicit data from explicit one. Several ontologies have been developed for the multimodal transport domain (see [7], [11], [15], [16], [18]). The more complete one is Ontology Transportation Network (OTN) [14], which allows the modeling of multimodal transport networks from different points of view such as spatial, temporal, services, etc.

Flexible querying systems allow users expressing preferences in their queries. These queries are addressed to regular relational databases and deliver a set of discriminated answers, which are ranked from the most to the least preferred element. Within the scope of flexible querying, the fuzzy set theory provides a general model for the interpretation of queries involving fuzzy predicates. It is also possible to consider fuzzy bipolar conditions to model preferences. Several interpretations are introduced for the evaluation of queries involving fuzzy bipolar conditions (see [10], [9], [21], [23], [24]), and in this paper, we rely on the interpretation introduced by Dubois and Prade [10], [9], in which a bipolar condition is made of two components: a constraint that is a mandatory condition and a wish that is an optional condition. More precisely, for the expression of user preferences, we rely on fuzzy bipolar conditions in which the constraint and the wish are defined by fuzzy sets. We define then a bipolar query as a query that involves bipolar conditions. A bipolar relational algebra is also proposed in [13], [3] and the algebraic operators (selection, projection, join, union, intersection) have been extended to fuzzy bipolar conditions.

We introduce in this paper a flexible querying approach of distributed and heterogeneous information systems, such as those developed for multimodal trip planning systems, which allows users to express complex preferences in their queries. In this context, the system has to deal with a large amount of data while taking into account efficiency in terms of quality of answers. Two aspects have been considered: (i) the expressivity in terms of complex preferences and query language and (ii) the efficiency of the data management. Our approach is then based on the combination of the Bipolar SQLf language (which is an extension of the SQLf language [2], [1] to fuzzy bipolar conditions, suitable to express complex preferences) with a reasoning system, built on a fuzzy bipolar DLR-Lite knowledge base (which is also an extension of the fuzzy DLR-Lite [8], [19] to fuzzy bipolar conditions, used to target the most relevant subset of data to answer user queries).

The remainder of the paper is organized as follows. Section II reminds both fuzzy sets theory (which is the main model for fuzzy conditions) and the SQLf language (which is an extension of the SQL language to fuzzy conditions). In section III, fuzzy bipolar conditions and Bipolar SQLf language are respectively described. Section IV recalls the fuzzy DLR-Lite and introduces our contribution which consists in the extension of the fuzzy DLR-Lite to fuzzy bipolar conditions and Bipolar SQLf language. In section V, the fuzzy bipolar DLR-Lite is used to develop a sample application in the field of multimodal transportation, in order to show the improvement that could be gained from integrating fuzzy bipolar conditions into both fuzzy DLR-Lite and SQLf language. Section VI summarizes our contribution and draws some lines for future works.

II. Flexible Querying Within the SQLf Language

In this section, we briefly introduce fuzzy sets theory, used to define fuzzy predicates such as *fast, high, large, expensive,* ..., and the SQLf language which is an extension of the SQL language to fuzzy conditions.

A. Fuzzy Sets Theory

The fuzzy sets theory is introduced by Zadeh [22] to express the gradual membership of an element to a set. Formally, a fuzzy set F is defined on a referential U by a membership function $\mu_F : U \mapsto [0,1]$ such that $\mu_F(x)$ denotes the membership grade of x in F.

In particular, $\mu_F(x) = 1$ denotes the full membership of x in F, $\mu_F(x) = 0$ expresses the absolute non-membership and when $0 < \mu_F(x) < 1$, it reflects a partial membership (the closer to 1 $\mu_F(x)$, the more x belongs to F).

A fuzzy set generalizes a crisp set in which membership grades are in $\{0,1\}$. If a fuzzy set is a discrete set then it is denoted $F = \{\mu_F(x_1)/x_1 + ... + \mu_F(x_n)/x_n\}$, else it is characterized by its membership function, generally a trapezoidal function.

The union $\cup$ and the intersection $\cap$ operators are defined with a couple of (t-norm, t-conorm) such as (min, max). Let F, G be two fuzzy sets, $\mu_{F\cup G}(x) = max(\mu_F(x), \mu_G(x))$, $\mu_{F\cap G}(x) = min(\mu_F(x), \mu_G(x))$, and the complement of F, noted F^c, is defined by $\mu_{F^c}(x) = 1 - \mu_F(x)$.

The logical counterparts of $\cap$, $\cup$ and the complement are resp. $\wedge, \vee$ and $\neg$. Other operators have also been defined such as fuzzy implications [4]. We consider particularly R-implications, which are based on the residual principle:

$$I_R(a,b) = sup\{z \in [0,1] : T(a,z) \leq b\}, \quad (1)$$

where $I_R(a,b)$ means $a \rightarrow_R b$ and T is a triangular norm.

R-implications satisfy properties of monotonicity, neutrality, exchange, identity and of ordering [4]. For example, the Gödel fuzzy implication is an R-implication which is defined by the formula (2), in which the used t-norm is min:

$$I_{Gö}(a,b) = sup\{z \in [0,1] : min(a,z) \leq b\}, \quad (2)$$

The Gödel fuzzy implication is also defined as follows: $I_{Gö}(a,b) = 1$ *if* $a \leq b, b$ *otherwise.*

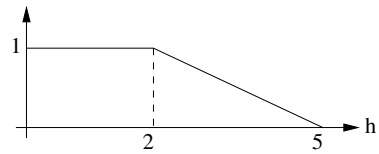

Fig. 1: The membership function of the fuzzy predicate *fast*.

TABLE I: Extension of the fuzzy relation *fastJourney*.

#Journey	cost ($)	duration (h)	...	$\mu_{FastJourney}$
12	80	2	...	1
13	75	3	...	0.66
10	65	4	...	0.33

B. The SQLf Language

The SQLf language [2], [1] is an extension of the SQL language to flexible querying by extending its statements to fuzzy conditions. When querying a regular relation R with such conditions, a fuzzy relation r (fuzzy set) is delivered. Each tuple t from r is attached with a degree of membership that expresses its level of satisfaction. The closer to 1 $\mu_r(t)$, the more t is preferred.

Example 1: Let *fast* be a fuzzy predicate as defined in Fig. 1. A fuzzy relation *fastJourney* can be defined from the relation *Journey (#Journey, source, destination, cost, duration)* s.t. for each tuple t: $\mu_{FastJourney}(t) = \mu_{Fast}(t.duration)$. Table I is an example of extension of *fastJourney*. ■

The SQLf language is based on the extension of the relational algebra operators to fuzzy conditions, defined as follows [2]: let r, s be two fuzzy relations such that the schema of r (resp. s) is X (resp. Y):

- *Fuzzy projection*: $\mu_{\pi(r,V)}(v) = max_w \mu_r(vw)$, where $V \subseteq X$, $v \in V$ and $w \in (X - V)$.
- *Fuzzy restriction*: $\mu_{\sigma(r,p)}(x) = min(\mu_r(x), \mu_p(x))$, where p is a fuzzy predicate.
- *Fuzzy join*: $\mu_{\bowtie}(r, s, \theta, A, B) = min(\mu_r(x), \mu_s(y), \mu_\theta(x.A, y.B))$, where A and B are compatible sets of attributes such that A (resp. B) is a subset of X (resp. Y) and $x.A$ (resp. $y.B$) is the value of A in x (resp. B in y), and θ is either a crisp or a fuzzy binary operator ($\theta \in \{=, \approx, <, >,$ *much larger than*, ...$\}$).

Syntactic elements of the SQLf language:

The main form of an SQLf query (fuzzy restriction) is:
Select [distinct] *[n|t|n,t] attributes* ***From*** *relations*
Where *fuzzy_cond;*

The above query returns a set of ranked tuples with their degree of satisfaction, where n is a limitation of the size of the returned set and $t \in]0,1]$ is a minimal threshold of satisfaction.

Example 2: We consider a sample database which is consistent with the Transmodel standard [6]. It contains the following relations: *Network*, *Line*, *Station*, *Link* and *LinkLine*. We define *Journey* relation as *Journey (#journey, #link, de-*

parture, arrival, source, destination, cost, duration, DepLink, ArrLink, modeLink, waitLink, durLink, costLink, comfLink).

For example, the query *"find the 2 fastest journeys from Lannion to Paris"* could be expressed in SQLf by:

***Select** 2 #journey **From** Journey **Where** source='Lannion' **and** destination='Paris' **and** fast (duration);*

The fuzzy condition *fast*$(duration)$ delivers the fuzzy relation *FastJourney* (see table I) and the query returns the 2 best answers : journeys $\#12$ and $\#13$, attached respectively with the satisfaction degrees 1 and 0.66. ■

The SQLf language allows more complex statements such as partitioning, nesting and division involving fuzzy relations.

III. Flexible Querying Within Bipolar SQLf Language

In this section, we briefly introduce fuzzy bipolar conditions and Bipolar SQLf language which is an extension of the SQLf language to fuzzy bipolar conditions.

A. Fuzzy Bipolar Conditions

A bipolar condition is an association of both negative condition (pole) and positive condition (pole). In this paper, a bipolar condition is made of two conditions defined on the same universe: i) a constraint c, which describes the set of acceptable elements, and ii) a wish w which defines the set of desired or wished elements. The negation of c is the set of rejected elements (non-acceptable elements). Since it is not coherent to wish a rejected element, the following property of coherence holds: $w \subseteq c$.

In addition, condition c is mandatory since an element which does not satisfy c is rejected ($\neg c$ is the negative condition), whereas w is optional because its non-satisfaction by an element does not mean its rejection (w is the positive condition). In this paper, a bipolar condition is noted (c, w) and means: *"satisfy c and if possible satisfy w"* [10], [9].

If c and w are boolean conditions, the satisfaction with respect to (c, w) is a couple from $\{0, 1\}^2$. When querying a database with such a condition, tuples satisfying the constraint and the wish are returned in priority. If such answers do not exist, the tuples satisfying only the constraint are delivered.

If c and w are fuzzy conditions (defined on the universe U), the property of coherence becomes: $\forall u \in U, \mu_w(u) \leq \mu_c(u)$ and the satisfaction with respect to (c, w) is a couple of degrees from $[0, 1]^2$. Each element u from U is then attached with a couple of grades $(\mu_c(u), \mu_w(u))$ that expresses the degree of its satisfaction respectively to the constraint c and the wish w.

When querying a relation R with a fuzzy bipolar condition, each tuple t from R is then attached with a couple of grades $(\mu_c(t), \mu_w(t))$ that expresses the degree of its satisfaction respectively to the constraint c and the wish w; and a so-called fuzzy bipolar relation is obtained. A tuple t is then denoted $(\mu_c, \mu_w)/t$. We assume that any tuple u such that $\mu_c(u) = 0$ does not belong to the fuzzy bipolar relation. In such a context, tuples cannot be ranked using an aggregation of μ_c and μ_w because the constraint and the wish are not commensurable. However, they can be ranked using the lexicographical order: t_1 is preferred to t_2, denoted $t_1 \succ t_2$, if and only if $\mu_c(t_1) > \mu_c(t_2)$ or $(\mu_c(t_1) = \mu_c(t_2) \wedge \mu_w(t_1) > \mu_w(t_2))$.

In this case, the satisfaction with respect to the constraint is firstly used to discriminate among answers (the constraint being mandatory). The satisfaction with respect to the wish being optional, it can only be used to discriminate among answers having the same evaluation with respect to the constraint. A total order is then obtained on μ_c and μ_w (with $(1, 1)$ as the greatest element and $(0, 0)$ as the least element).

Based on the lexicographical order, the $lmin$ and $lmax$ operators [12], [3] are introduced in order to define the conjunction (resp. intersection) and the disjunction (resp. union) of bipolar conditions (resp. relations). They are defined as follows:

$$\begin{array}{lcl} ([0,1]\times[0,1])^2 & \rightarrow & [0,1]\times[0,1] \\ ((\mu,\eta),(\mu',\eta')) & \mapsto & lmin((\mu,\eta),(\mu',\eta')) = \\ & & \begin{cases} (\mu,\eta) & if \mu < \mu' \vee (\mu = \mu' \wedge \eta < \eta'), \\ (\mu',\eta') & else. \end{cases} \end{array}$$

$$\begin{array}{lcl} ([0,1]\times[0,1])^2 & \rightarrow & [0,1]\times[0,1] \\ ((\mu,\eta),(\mu',\eta')) & \mapsto & lmax((\mu,\eta),(\mu',\eta')) = \\ & & \begin{cases} (\mu,\eta) & if \mu > \mu' \vee (\mu = \mu' \wedge \eta > \eta'), \\ (\mu',\eta') & else. \end{cases} \end{array}$$

The $lmin$ (resp. $lmax$) operator is commutative, associative, idempotent and monotonic. The couple of grades $(1, 1)$ is the neutral (resp. absorbing) element of the operator $lmin$ (resp. $lmax$) and the couple $(0, 0)$ is the absorbing (resp. neutral) element of the operator $lmin$ (resp. $lmax$).

Remark 1: Fuzzy bipolar conditions generalize fuzzy conditions since a fuzzy condition c can be rewritten (c, c) to express *" satisfy c and if possible satisfy c"*. So, a fuzzy relation R is a particular case of a fuzzy bipolar relation such that $\forall t \in R, \mu_R(t) = \mu_c(t) = \mu_w(t)$. It has been proven [12], [3] that the $lmin$ (resp. $lmax$) extends the t-norm min (resp. the t-co-norm max) to bipolarity.

B. Basis of Bipolar SQLf Language

Bipolar SQLf language is an extension to fuzzy bipolar conditions of the SQLf language. We introduce in this subsection the fuzzy bipolar extension proposed in [12], [3] of the relational algebraic operators (the union, the intersection, the cartesian product, the projection, the selection and the join) of fuzzy bipolar relations. This extension is based on the couple of the extended t-norm and t-co-norm $(lmin, lmax)$.

Let r and s be two fuzzy bipolar relations defined respectively by fuzzy bipolar conditions (c_1, w_1) and (c_2, w_2).

1) The intersection operator: The intersection of r and s is a fuzzy bipolar relation, in which each tuple t is attached with a couple of degrees $(\mu_c(t), \mu_w(t))$. It is defined as follows:

$$r \cap s = \{(\mu_c, \mu_w)/t | (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_{c_2}, \mu_{w_2})/t \in s \wedge (\mu_c, \mu_w) = lmin((\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{c_2}(t), \mu_{w_2}(t)))\}.$$

2) The union operator: The union of r and s is a fuzzy bipolar relation, in which each tuple t is attached with a couple of degrees $(\mu_c(t), \mu_w(t))$. It is defined as follows:

$$r \cup s = \{(\mu_c, \mu_w)/t | (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_{c_2}, \mu_{w_2})/t \in s \wedge (\mu_c, \mu_w) = lmax((\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{c_2}(t), \mu_{w_2}(t)))\}.$$

3) The cartesian product operator: The cartesian product of r and s is a fuzzy bipolar relation, in which each tuple t is attached with a couple $(\mu_c(t), \mu_w(t))$. It is defined as follows:

$$r \otimes s = \{(\mu_c, \mu_w)/t \oplus t' | (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_{c_2}, \mu_{w_2})/t' \in s \wedge (\mu_c, \mu_w) = lmin((\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{c_2}(t'), \mu_{w_2}(t')))\},$$

where $\oplus$ is the operator of concatenation of tuples.

4) The projection operator: The projection of distinct tuples on attributes $a_1, ..., a_k$ from r is a fuzzy bipolar relation of elements $\langle a_1, ..., a_k \rangle$, defined as follows:

$$\pi_{a_1,...,a_k}(r) = \{(\mu'_{c_1}, \mu'_{w_1})/\langle a_1, ..., a_k \rangle \mid (\mu'_{c_1}, \mu'_{w_1}) = lmax_{t \in r \wedge t[a_1,...,a_k] = \langle a_1,...,a_k \rangle}((\mu_{c_1}(t), \mu_{w_1}(t))),$$

where $t[a_1, ..., a_k]$ is the value of the tuple t on $a_1, ..., a_k$.

5) The selection operator: The selection of tuples from r, based on the fuzzy bipolar condition (c', w') is defined as:

$$\sigma(r, (c', w')) = \{(\mu_c, \mu_w)/t | (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_c, \mu_w) = lmin((\mu_{c'}(t), \mu_{w'}(t)), (\mu_{c_1}(t), \mu_{w_1}(t)))\},$$

where $\mu_{c'}(t)$ (resp. $\mu_{w'}(t)$) is the degree of satisfaction of t with respect to the constraint c' (resp. the wish w').

6) The join operator: As in both boolean and fuzzy cases, the join operator is defined in the context of bipolarity by the combination of a cartesian product with a selection operator.

Bipolar SQLf basic statements:

A Bipolar SQLf basic statement is a fuzzy bipolar selection (restriction) of the following main format:

Select [distinct] *[$n|(t_1, t_2)|n, (t_1, t_2)$] attributes* ***From*** *relations [**as** alias]* ***Where*** *$fcond_1$* ***and if possible*** *$fcond_2$;*

The above statement is a simple selection of attributes from fuzzy bipolar relations based on the fuzzy bipolar condition ($fcond_1$,$fcond_2$). It is also possible to express an n-top query by positioning the optional integer value n, which delivers the n best tuples. $(t_1, t_2) \in [0, 1]^2$, such that $t_2 \leq t_1$, is a minimal threshold of satisfaction.

Example 3: We reuse the database defined in the example 2. The query *"Find journeys which are fast and if possible not expensive"*, can be expressed in Bipolar SQLf as:

Select *#Journey* ***From*** *Journey as J* ***Where*** *fast($J.duration$)* ***and if possible*** *not expensive($J.cost$);*

Due to the coherence property of fuzzy bipolar conditions, the above bipolar query is interpreted as *"Find journeys which are fast and if possible (fast and not expensive)"*.

We define, for example, the fuzzy predicate *expensive* as depicted in Fig. 2. Its negation is defined as follows: $\forall x \in \mathbb{R}^+, \mu_{notExpensive}(x) = 1 - \mu_{Expensive}(x)$.

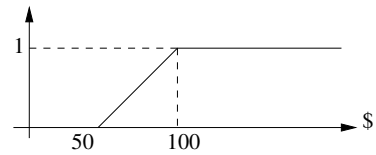

Fig. 2: The membership function of the fuzzy predicate *expensive*.

TABLE II: Fuzzy bipolar relation $Journey_{(Fast,\ notExpensive)}$.

#Journey	cost ($)	duration (h)	μ_{Fast}	$\mu_{FastJourney} \wedge \mu_{notExpensive}$
12	80	2	1	0.40
13	75	3	0.66	0.50
10	65	4	0.33	0.33

Based on the definition of fuzzy predicates *fast* and *not expensive*, the query is evaluated over the relational table *Journey* and delivers the fuzzy bipolar relation $Journey_{(Fast,\ notExpensive)}$. Table II is an example of extension of $Journey_{(Fast,\ notExpensive)}$, and the delivered tuples are ranked, using the lexicographical order, from the best journey to the worst one, depending on their satisfactions to both the constraint *fast* and the wish *not expensive*: $(1, 0.40)/12$, $(0.66, 0.50)/13$, $(0.33, 0.33)/10$. ■

IV. Fuzzy Bipolar DLR-Lite

In this section, we introduce the fuzzy DLR-Lite and our contribution which consists in its extension to bipolarity called fuzzy bipolar DLR-Lite.

A. The Fuzzy DLR-Lite

The flexible querying system developed in [8], [19] is based on a Description Logic extended to m-ary Relations, noted DLR-lite [5]. The knowledge base $\mathcal{K}$ of the fuzzy DLR-Lite consists of three components $(\mathcal{O}, \mathcal{F}, \mathcal{A})$ defined as follows:

1) The facts component $\mathcal{F}$: It contains extensions of relations of the form $R(c_1, ..., c_m)[s]$, where R is an m-ary relation, $c_{i,i=1,...,m}$ are constants and $s \in]0, 1]$ is a degree of membership of tuple $\langle c_1, ..., c_m \rangle$ in the relation R.

2) The ontology component $\mathcal{O}$: the ontology defines the relevant abstract concepts of the application domaine by means of projection and inclusion axioms:

a) The inclusion axiom: The inclusion axiom has the form $(R_1 \sqcap R_2 \sqcap ... \sqcap R_l \sqsubseteq R_r)[s]$, where $l \geq 1$, $R_{i,i=1,...,l}$ are relations of the same arity and $s \in]0, 1]$ is the truth value of the axiom. This axiom states that if c is an instance of $R_{i,i=1,...,l}$ to degree $s_{i,i=1,...,l}$ then c is an instance of R_r to degree at least $s \otimes s_1 \otimes ... \otimes s_l$, where $\otimes$ is a t-norm.

b) The projection axioms: They allow defining new concepts by extracting columns from other relations. Two kinds of projection axioms are defined: *the simple projection* and *the restricted projection*. The former is of the form $\exists[i_1, ..., i_k]R$ that expresses the projection of the relation R on the columns $i_1, ..., i_k$ and the latter is of the form $\exists[i_1, ..., i_k]R.(Cond_1 \sqcap ... \sqcap Cond_h)$ which restricts the projection of $\exists[i_1, ..., i_k]R$

according to conditions $Cond_{i,i=1...h}$ of the form $([i]\theta v)$, where $\theta \in \{<, \leq, =, \neq, >, \geq\}$ and v is a value.

3) The abstraction component $\mathcal{A}$: It forms the set of statements which connect concepts and relations, defined in the ontology component, to physical relational tables from a database. Simple and complex abstraction statements are defined as follows:

a) Simple abstraction statement: It has the form $R_1 \longmapsto R_2(c_1[t_1], ..., c_k[t_k])[c_s]$ which states that the k-ary relation R_1 of the ontology component is mapped into the projection on columns c_1 of type t_1, c_2 of type t_2, ..., and on c_k of type t_k of the m-ary table R_2. The score of these tuples is provided by the column c_s.

b) Complex abstraction statement: It has the form $R_1 \longmapsto (t_1, ..., t_k)[c_s].sql$, where R_1 is a k-ary relation of the ontology component and sql is an SQL query which returns the k-ary tuples $\langle t_1, ..., t_k \rangle$ ranked in decreasing order of scores provided by column c_s.

4) The query language: A query in a fuzzy DLR-Lite knowledge base querying system consists of a conjunctive query with a scoring function to rank between answers. The query has the following form:

$$\underbrace{q(x)[s]}_{head} \leftarrow \underbrace{\exists y R_1(z_1)[s_1], ..., R_l(z_l)[s_l],}_{body} \underbrace{OrderBy(s = f(s_1, ..., s_l)),}_{scoring\,function} \quad (3)$$

where q is an m-ary relation, every R_i is either an m_i-ary relation or a concrete predicate of the form of $(z\theta v)$, where $\theta \in \{<, \leq, =, \neq, >, \geq\}$ and v is a value of a datatype, x (resp. y) is a vector of distinguished (resp. non distinguished) variables, z_i is a tuple of constants or variables in x or y, f is a scoring function which combines scores $s_1, ..., s_l$ and s is a degree of satisfaction of a tuple x to the query q.

B. Extension of the Fuzzy DLR-Lite to Bipolarity

The bipolarity is integrated into the fuzzy DLR-Lite knowledge base in order to provide means to define and to handle fuzzy bipolar concepts and relations. For that purpose, we propose the extension of the fuzzy DLR-Lite to fuzzy bipolar conditions as described in what follows.

1) Extension of the facts component $\mathcal{F}$: Facts in our case are delivered from fuzzy bipolar relations. A fact has the following form: $R(c_1, ..., c_n)[s_c, s_w]$, where $c_{i,i=1,...,n}$ are constants which form the tuple $u = \langle c_1, ..., c_n \rangle$, R is a fuzzy bipolar relation defined by the fuzzy bipolar condition (c, w) and s_c (resp. s_w) is the grade of satisfaction of the tuple $u = \langle c_1, ..., c_n \rangle$ with respect to the constraint c (resp. to the wish w). The consistency condition holds for each element u ($s_w \leq s_c$) and every tuple u s.t. $s_c(u) = 0$ is discarded.

2) Extension of the abstraction component $\mathcal{A}$: Simple and complex abstraction rules are both extended to fuzzy bipolar conditions as follows:

a) The simple abstraction rule: It has the form $R_1 \longmapsto R_2(c_1, ..., c_n)[s_c, s_w]$, where R_1 is a relation or concept (in the sense of the fuzzy DLR-Lite); R_2 is an m-ary fuzzy bipolar relation from a database, $c_1, ..., c_n$, with $n \leq m$, are columns from R_2 and (s_c, s_w) is the couple of grades of satisfaction attached to the tuple $\langle c_1, ..., c_n \rangle$ with respect to the fuzzy bipolar condition (c, w) which is used to define R_2.

b) The complex abstraction rule: It has the form $R_1 \longmapsto (c_1, ..., c_n)[s_c, s_w].$*sqlfb*, where *sqlfb* is a Bipolar SQLf query which delivers n-ary tuples $\langle c_1, ..., c_n \rangle$ with their corresponding couples of grades (s_c, s_w).

3) Extension of the ontology component $\mathcal{O}$: The ontology component defines intersection axioms ($\sqcap$), simple and conditional projection axioms and subsumption (inclusion) axioms ($\sqsubseteq$). They are extended to bipolarity as follows:

a) The intersection axiom: It is denoted $R_1 \sqcap R_2$, where R_1 and R_2 are fuzzy bipolar relations defined resp. by fuzzy bipolar conditions (c_1, w_1) and (c_2, w_2). It is interpreted as a fuzzy bipolar relation R in which each tuple u is attached with a couple of degrees $(s_c(u), s_w(u))$, defined as follows:

$$(R_1 \sqcap R_2)^{\mathcal{I}} \Leftrightarrow \forall u, (s_c(u), s_w(u)) = lmin((\mu_{c_1}(u), \mu_{w_1}(u)), (\mu_{c_2}(u), \mu_{w_2}(u))) \quad (4)$$

b) The simple projection axiom: It is denoted $\exists [i_1, ..., i_k] R$. It corresponds to the projection of columns $i_1, ..., i_k$ from the fuzzy bipolar relation R of arity n such that $n \geq k$. This projection extracts new facts from a database or from facts already defined in $\mathcal{F}$.

This rule is interpreted as a bipolar projection of distinct elements of fields $i_1, ..., i_k$ from the fuzzy bipolar relation R defined by the fuzzy bipolar condition (c, w). This axiom delivers a fuzzy bipolar relation made of the projected columns and its interpretation attaches a couple of grades for each retrieved tuples $u = \langle i_1, ..., i_k \rangle$ as follows:

$$\forall u, (\exists [i_1, ..., i_k] R)(u)^{\mathcal{I}} = (s_c(u), s_w(u)) = \underset{t \in R \wedge t[i_1, ..., i_k] = u}{lmax} ((\mu_c(t), \mu_w(t))) \quad (5)$$

c) The restricted projection axiom: It is denoted $\exists [i_1, ..., i_k] R.(Cond_1 \sqcap ... \sqcap Cond_h)$, where $Cond_{i,i=1...h} = ([i]\theta v)$ are conditions of selection of fields with $\theta \in \{<, \leq, =, \neq, >, \geq\}$, $[i]$ is a column number corresponding to a field from the fuzzy bipolar relation R, defined by the fuzzy bipolar condition (c, w), v is a value and $\sqcap$ is the operator and. This rule corresponds to a combination of a bipolar projection and a bipolar selection of distinct tuples from R. It delivers a fuzzy bipolar relation formed by the retrieved tuples $u = \langle i_1, ..., i_k \rangle$. The attached couples of grades are defined by the formula (5) applied on the retrieved tuples limited to those satisfying the boolean condition $Cond = (Cond_1 \sqcap ... \sqcap Cond_h)$.

$$\forall u, (\exists [i_1, ..., i_k] R.(Cond_1 \sqcap ... \sqcap Cond_h))(u)^{\mathcal{I}} = (s_c(u), s_w(u)) = \underset{t \in R \wedge t[i_1, ..., i_k] = u}{lmax} ((\mu_c(t), \mu_w(t))), \; if \; Cond^{\mathcal{I}}(u) = 1 \quad (6)$$

d) The subsumption (inclusion) axiom: This axiom is denoted $(R_l \sqsubseteq R_r)[n_1, n_2]$, where R_l and R_r are fuzzy bipolar concepts (relations) of the same arity defined respectively by fuzzy bipolar conditions (c_l, w_l) and (c_r, w_r).

The interpretation of this axiom is based on the strengthened membership-based interpretation of fuzzy bipolar conditions. Let R be a fuzzy bipolar relation defined by a fuzzy bipolar condition (c, w). R is seen as a particular fuzzy set of desired elements such that each element u holds two grades:

- $\mu_c(u)$ is a basic grade of membership to the fuzzy set,
- $\mu_w(u)$ is a strengthening grade of membership that make distinguishable tuples having the same basic grade. In other terms, for tuples having the same basic grade, more a tuple satisfies the wish more it belongs to the fuzzy set.

In this context, it becomes possible to extend the interpretation of the inclusion axiom to bipolarity as follows:

$$((R_l \sqsubseteq R_r)[n_1, n_2])^{\mathcal{I}} = \forall x \in R_l, \\ lmin((\mu_{c_l}(x), \mu_{w_l}(x)), (n_1, n_2)) \leq_{lex} (\mu_{c_r}(x), \mu_{w_r}(x)) \quad (7)$$

where $\leq_{lex}$ is the operator s.t. $(a, b) \leq_{lex} (a', b') \Leftrightarrow lmin((a, b), (a', b')) = (a, b)$. The couple of degrees (n_1, n_2) is interpreted as the minimal threshold of inclusion of R_l in R_r, and it processed by the following formula:

$$((R_l \sqsubseteq R_r)[n_1, n_2])^{\mathcal{I}} \Leftrightarrow \\ lmin_{x \in R_l}((\textit{if}\, x \in R_l\, \textit{then}\, x \in R_r)^{\mathcal{I}}) = (n_1, n_2) \quad (8)$$

The formula (8) is the extension to fuzzy bipolar conditions of the formula (9), which is introduced in [20] to compute the truth value of inclusion axioms in the case of fuzzy DLR-Lite.

$$((R_l \sqsubseteq R_r)[n])^{\mathcal{I}} \Leftrightarrow min_x(R_l(x) \rightarrow_{G\ddot{o}} R_r(x)) = n, \quad (9)$$

where $\rightarrow_{G\ddot{o}}$ is the Gödel fuzzy implication.

The formula $(\textit{if}\ x \in R_l\, \textit{then}\, x \in R_r)^{\mathcal{I}}$ defines a kind of implication for bipolar elements. We propose the extension of the formula (2) to bipolarity to define an extended bipolar R-implication operator. The formula (10) is, then, proposed:

$$I_R((a_c, a_w), (b_{c'}, b_{w'})) = lmax\{(z, z') \in [0, 1]^2 : z' \leq z \wedge \\ lmin((a_c, a_w), (z, z')) \leq_{lex} (b_{c'}, b_{w'})\}. \quad (10)$$

Finally, we obtain the following interpretation for $(\textit{if}\ (x \in R_l)\, \textit{then}\, (x \in R_r))$ in the bipolar context:

$$(\textit{if}\ (x \in R_l)\, \textit{then}\, (x \in R_r))^{\mathcal{I}} = \\ \begin{cases} (1, 1)\ \textit{if}(\mu_{c_l}(x), \mu_{w_l}(x)) \leq_{lex} (\mu_{c_r}(x), \mu_{w_r}(x)), \\ (\mu_{c_r}(x), \mu_{w_r}(x)), else. \end{cases} \quad (11)$$

It is easy to prove that formula (11) verifies properties of R-implications extended to bipolarity (monotonicity, neutrality, exchange, identity and ordering).

It is also easy to show that formula (11) is compatible with formula (10) and it defines an extension of the Gödel fuzzy implication to fuzzy bipolar conditions. Indeed, in the formula (10), when $(a_c, a_w) \leq (b_{c'}, b_{w'})$, any valid couple of degrees (z, z') from $[0, 1]^2$ satisfies $lmin((a_c, a_w), (z, z')) \leq (b_{c'}, b_{w'})$ and the greatest couple of degrees is $(1, 1)$. When $(a_c, a_w) > (b_{c'}, b_{w'})$, couples of degrees which satisfy $lmin((a_c, a_w), (z, z')) \leq (b_{c'}, b_{w'})$ are (z, z') such that $(z, z') \leq (b_{c'}, b_{w'})$; then, the greatest couple of degrees (z, z') corresponds to $(b_{c'}, b_{w'})$.

4) Extension of the query language to bipolarity: A query addressed to a fuzzy bipolar DLR-Lite knowledge base consists in a conjunctive query which is extended to bipolarity. The main form of such queries is:

$$\underbrace{q(x)[s_c, s_w]}_{head} \leftarrow \underbrace{\exists y R_1(z_1)[s_{c_1}, s_{w_1}], ..., R_l(z_l)[s_{c_l}, s_{w_l}],}_{body} \\ \underbrace{OrderBy((s_c, s_w) = f((s_{c_1}, s_{w_1}), ..., (s_{c_l}, s_{w_l})))}_{scoring\,function}. \quad (12)$$

Symbols q, x, y, R_i and z_i are the same as those used in the fuzzy case (formula (3)). The main form of conjunctive queries are extended to bipolarity to handle couples of degrees expressed at each atom of the body. These couples of degrees are finally aggregated by the bipolar scoring function f, which can be simplified to the $lmin$ operator, due to the conjunctive nature of the query. $[s_c, s_w]$ corresponds to couples of degrees (s_c, s_w) returned by f and attached to the resulting tuples. Syntactically, couples of the form $[1, 1]$ could be omitted.

V. Fuzzy Bipolar DLR-Lite Approach for Flexible Querying: a Sample Application

We show the relevance of the fuzzy bipolar DLR-Lite based approach through the design and the querying of a sample application in the field of multimodal transportation networks.

The relational database previously introduced in the example 2 is reused. For each table from the database, a relation is defined in the ontology component and an abstraction statement is associated to it in the abstraction component. In addition, we define the following relations in the ontology:

$Step \equiv \exists[2, 3, 4, 5, 6]Journey$ is a simple projection of *Journey* relation on columns *#link*, *source*, *destination*, *departure* and *arrival*.

$AreaStation \equiv \exists[1, 2, 3, 4]Station$ defines stations such as *railway station* and *airport*. It is a simple projection of *Station* relation on columns *#station*, *designation*, *city* and *type*.

Let *RailwayStation*, noted *RailStation*, and *Airport* be two concepts, extracted from the *AreaStation* by the following restricted projection axioms:

- $RailStation \equiv \exists[1]AreaStation.(([3] = "Railway"))$
- $Airport \equiv \exists[1]AreaStation.(([3] = "Airport"))$

Let *PreferredRailStation* be a fuzzy bipolar concept defined by the fuzzy bipolar condition *(Large, NearCenter)* that corresponds to railway stations which are *large and if possible (large ∧ located near to the town center)*. This concept could be expressed by the following abstraction rule:

*PreferredRailStation ↦ (#station).**Select** #station **From** RailStation **Where** LargeRail(#station)* and if possible *NearCenter(#station)*;

where *LargeRail* (resp. *NearCenter*) is the membership function of the fuzzy predicate *LargeStation* (resp. *near*

TABLE III: Examples of relations from the facts component.

(a) *PreferredRailStation* relation.

#station	designation	μ_{Large}	$\mu_{Large} \wedge \mu_{NearCenter}$
1	RailStation1	0.5	0.4
2	RailStation2	1	0.7
3	RailStation3	0.6	0.5

(b) *PreferredAirport* relation.

#airport	designation	μ_{Large}	$\mu_{Large} \wedge \mu_{NearCenter}$
10	Airport1	1	0.8
11	Airport2	0.5	0.3
12	Airport3	0.4	0.4

TABLE IV: *hasPreferredRestaurant* relation.

#place	designation	μ_{Good}	$\mu_{Good} \wedge \mu_{Cheap}$
100	RailStation1	0.6	0.6
102	RailStation2	0.8	0.7
103	RailStation3	0.8	0.3
104	Airport1	0.7	0.6
105	Airport2	0.5	0.4
106	Airport3	0.5	0.3
107	ShoppingCentre1	0.6	0.25
108	Hotel1	0.4	0.3

to the town center). Similarly, let *PreferredAirport* be a fuzzy bipolar concept defined by the fuzzy bipolar condition *(Large, NearCenter)* that corresponds to airports which are *large and if possible (large $\wedge$ not located far from the town center)*. Tables IIIa and IIIb displays examples of facts attached respectively to these concepts.

We can enrich the system with data from web services or other information systems. For example, we query a web service about restaurants, from which we define the fuzzy bipolar relation *hasPreferredRestaurant*, which corresponds to places which have *good and if possible cheap* restaurants as railway stations, hotels, airports, shopping centers (see table IV). Instead of saving the table IV on the facts component, we focus only on data related to our transport system. For efficiency reason, we represent them as data summary using the two following bipolar inclusion axioms:
(*PreferredRailStation* $\sqsubseteq \exists[1]$*hasPreferredRestaurant*)$[n_1, n_2]$
(*PreferredAirport* $\sqsubseteq \exists[1]$*hasPreferredRestaurant*)$[n_3, n_4]$
where (n_1, n_2) (resp. (n_3, n_4)) is the minimal couple of degrees of inclusion of *PreferredRailStation* (resp. *PreferredAirport*) in the set of *hasPreferredRestaurant*. These couples of degrees are processed based on formulas (7), (8) and (11):

$$\begin{aligned}(n_1, n_2) &= lmin((0.5, 0.4) \rightarrow (0.6, 0.6),\\ &\qquad (1, 0.7) \rightarrow (0.8, 0.7), (0.6, 0.5) \rightarrow (0.8, 0.3))\\ &= lmin((1, 1), (0.8, 0.7), (1, 1)) = (0.8, 0.7).\\ (n_3, n_4) &= lmin((1, 0.8) \rightarrow (0.7, 0.6),\\ &\qquad (0.5, 0.3) \rightarrow (0.5, 0.4), (0.4, 0.4) \rightarrow (0.5, 0.3))\\ &= lmin((0.7, 0.6), (1, 1), (1, 1)) = (0.7, 0.6).\end{aligned}$$

Even if we have no restaurant relation in the facts component and without querying any external database, the system can answer queries about journeys which involve preferred restaurants, such as *"go from Lannion to Paris in 2 steps with a preferred restaurant at the midway"*, which is expressed by:

$$\begin{aligned}&q(x_1, x_2)[s_c, s_w] \leftarrow \exists a, b, c, d, e, f, g, h, i.\\ &\quad AreaStation(a, e, "Lannion"), Step(x_1, a, c, d, g),\\ &\quad Step(x_2, c, b, d, h), AreaStation(b, f, "Paris"),\\ &\quad \textit{hasPreferredRestaurant}(c, i)[s_{c_1}, s_{w_1}],\\ &\quad OrderBy((s_c, s_w) = (s_{c_1}, s_{w_1})).\end{aligned}$$

To evaluate this query, the atom *hasPreferredRestaurant* is substituted by the left-hand side of each inclusion axiom in which it appears in its right-hand side. Each substitution generates a new query. Two subqueries are, then, obtained:

$$\begin{aligned}&q_1(x_1, x_2)[s_c, s_w] \leftarrow \exists a, b, c, d, e, f, g, h.\\ &\quad AreaStation(a, e, "Lannion"), Step(x_1, a, c, d, g),\\ &\quad Step(x_2, c, b, d, h), AreaStation(b, f, "Paris"),\\ &\quad \textit{PreferredRailStation}(c)[s_1, s_2],\\ &\quad OrderBy((s_c, s_w) = lmin((s_1, s_2), (0.8, 0.7))).\\ &q_2(x_1, x_2)[s_c, s_w] \leftarrow \exists a, b, c, d, e, f, g, h.\\ &\quad AreaStation(a, e, "Lannion"), Step(x_1, a, c, d, g)\\ &\quad Step(x_2, c, b, d, h), AreaStation(b, f, "Paris"),\\ &\quad \textit{PreferredAirport}(c)[s_3, s_4],\\ &\quad OrderBy((s_c, s_w) = lmin((s_3, s_4), (0.7, 0.6))).\end{aligned}$$

The scoring function in both q_1 and q_2 takes into consideration the inclusion axiom used in the substitution process. Both q_1 and q_2 are to be translated into Bipolar SQLf queries, in order to be evaluated over the facts component.

For example, query $q_1(x_1, x_2)$ is expressed as follows:
***Select** R1.#step, R2.#step **From** Step **as** R1, Step **as** R2 **Where** R1.destination = R2.source **and** R1.source **in (Select** #station **From** Station **Where** city = 'Lannion') **and** R2.destination **in (Select** #station **From** Station **Where** city = 'Paris') **and** R2.Source **in (Select** #station **From** PreferredRailStation);*

The answers are ranked depending on couples of degrees delivered by $lmin((s_1, s_2), (0.8, 0.7))$, where (s_1, s_2) is the couple of degrees attached to each *R2.Station* corresponding to a preferred rail station. The query $q_2(x_1, x_2)$ is similarly processed and the obtained sets of answers are merged, while respecting the lexicographical order.

The Main Architecture of the System:

For implementation purpose, we propose a software architecture which consists of a combination of a reasoning system (based on the fuzzy bipolar DLR-Lite) and a flexible information system (based on Bipolar SQLf).

As depicted in Fig. 3, the system is of two parts. The left side consists of the following modules:

a) Query interpreter: It provides the user with interface that receives queries and returns answers.

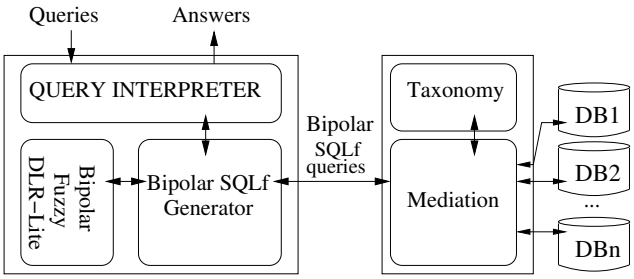

Fig. 3: Software architecture of the flexible querying system.

b) Bipolar SQLf generator: It performs user queries transformation (relevant substitutions based on the inclusion axioms) using the fuzzy bipolar DLR-Lite knowledge base and generates equivalents bipolar SQLf queries to be evaluated by the right side of the system.

c) Fuzzy bipolar DLR-Lite knowledge base: it defines the knowledge base of the system (relations, facts, axioms).

The right side deals with the mediation issues for the combination of data fetched from remote databases, and processes bipolar SQLf queries received from the bipolar SQLf generator. It is composed by the following modules:

d) Mediation: It evaluates the received bipolar SQLf queries by retrieving the needed data from relevant remote databases, and by translating queries using the right vocabulary in compliance with a domain ontology (taxonomy).

e) Taxonomy: It is based on the Ontology Transportation Networks (OTN). Depending on the remote information system to query, this module provides the *Mediation* module with the right vocabulary to use in the query translation process.

VI. Conclusion and Future Works

To be able to express complex preferences in user queries, addressed to distributed and heterogeneous information systems, we have proposed a new flexible querying approach in which the fuzzy DLR-Lite is extended to both fuzzy bipolar conditions and Bipolar SQLf language.

We showed through an application the relevance of this approach in the field of the multimodal transportation networks.The bipolar fuzzy DLR-Lite can go beyond relational tables by inferring implicit knowledge and by summarizing data, while improving widely the expressiveness of the system. So, it becomes possible to express complex preferences of the form *"c and if possible w"* and to handle them within a reasoning system to answer user queries.

A software architecture for such a querying system has also been proposed. The common formal framework (fuzzy sets theory) of both Bipolar SQLf language and fuzzy bipolar DLR-Lite allows their combination in a consistent way.

An implementation of the system is in progress. It is based on algorithms for query evaluation which are introduced in [19]. As future works, we plan the study of performances of our approach in terms of response time and quality of answers, i.e. in which extent answers meet user expectations.

Acknowledgments

We warmly thank the Brittany region, the department of Côtes-d'Armor and the National Agency for Research (AOC Ref. ANR-08-CORD- 009) for financing this work.

References

[1] P. Bosc, L. Liétard, O. Pivert, and D. Rocacher, *Base de données - Gradualité et imprécision dans les bases de données Ensembles flous, requêtes flexibles et interrogation de données mal connues*, 1st ed., ser. Technosup, Ellipse, Ed., 2004.

[2] P. Bosc and O. Pivert, "SQLf: A relational database langage for fuzzy querying," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 1, pp. 1–17, 1995.

[3] P. Bosc, O. Pivert, L. Liétard, and A. Mokhtari, "Extending relational algebra to handle bipolarity," in *SAC'10*, 2010, pp. 1717–1721.

[4] B. Bouchon-Meunier, D. Dubois, L. Godo, and H. Prade, *Fuzzy sets in approximate reasoning and information systems*, ser. The Handbook of fuzzy sets. Kluwer Academic Publishers, 1999, ch. 1 : Fuzzy set and possibility theory in approximate and plausible reasoning, pp. 27–31.

[5] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, "Data complexity of query answering in description logics," in *KR*, 2006, pp. 260–270.

[6] O. Carles, "Modèle conceptuel de données générique pour la représentation des réseaux intermodaux," INRETS, Tech. Rep., 2002.

[7] K.-H. Chen, C.-R. Dow, and S.-J. Guan, "Nimbletransit: Public transportation transit planning using semantic service composition schemes," in *11th Inter. IEEE con. on ITS*, 2008, pp. 723–728.

[8] S. Colucci, T. D. Noia, A. Ragone, M. Ruta, U. Straccia, and E. Tinelli, *Semantic Web Information Management*. Springer-Verlag Belin Heidelberg, 2010, ch. 19 : Informative Top-k retrieval for advanced skill management, pp. 449–476.

[9] D. Dubois and H. Prade, "Bipolarité dans un processus d'interrogation flexible," in *LFA*, 2002.

[10] ——, "Bipolarity in flexible querying," *LNAI*, vol. 2522, pp. 174–182, 2002.

[11] R. Lewis, F. Fuchs, M. Pirker, C. Roberts, and G. Langer, "Using ontology to integrate railway condition monitoring data," in *International Conference on Railway Condition Monitoring*, 2006, pp. 149–155.

[12] L. Liétard and D. Rocacher, "On the definition of extended norms and co-norms to aggregate fuzzy bipolar conditions," in *IFSA/EUSFLAT*, 2009, pp. 513–518.

[13] L. Liétard, D. Rocacher, and P. Bosc, "On the extension of sql to fuzzy bipolar conditions," in *The 28th NAFIPS'09*, 2009.

[14] B. Lorenz and M. Rosener, "Ontology of transportation networks," REWERSE: reasoning on the web. European Commission and Swiss Federal Office for Education and Science, Tech. Rep., 2005.

[15] A. S. Niaraki and K. Kim, "Ontology based personalized route planning system using a multi-criteria decision making approach," *Expert Systems with Applications*, vol. 36, pp. 2250–2259, 2009.

[16] M. Obitko and V. Marik, "Integration transportation ontologies using semantic web languages," *HoloMas, LNAI*, vol. 3593, pp. 99–110, 2005.

[17] Predim, *Predim : Plate-forme de recherche et d'expérimentation pour le développement de l'information multimodale*, PREDIM Std., Nov 2007.

[18] S. Saad, H. Zgaya, and S. Hammadi, "The flexible negotiation ontology-based knowledge management system: the transport ontology case study," in *3rd ICTTA From Theory to Applications*, 2008, pp. 1–7.

[19] U. Straccia, "Softfacts : a top-k retrieval engine for a tractable description logic accessing relational databases," ISTI-CNR, Tech. Rep., 2009.

[20] N. Tamani, L. Liétard, and D. Rocacher, "Extension d'une ontologie floue pour l'interrogation flexible d'une base de données embarquée," in *LFA*, 2010.

[21] G. D. Tré, S. Zadrozny, T. Matthé, J. Kacprzyk, and A. Bronselaer, "Dealing with positive and negative query criteria in fuzzy database quering bipolar satisfaction degrees," *LNAI, FQAS*, vol. 5822, pp. 593–604, 2009.

[22] L. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[23] S. Zadrozny and J. Kacprzyk, "Bipolar queries using various interpretations of logical connectives," *LNAI, IFSA*, vol. 4529, pp. 182–190, 2007.

[24] ——, "Bipolar queries: An approach and its various interpretations," in *In Proceedings of IFSA/EUSFLAT*, 2009, pp. 1288–1293.