Bipolar Querying of Valid-Time Intervals Subject to Uncertainty

Christophe Billiet[1], José Enrique Pons[2], Olga Pons[2], and Guy De Tré[1]

[1] Department of Telecommunications and Information Processing, Ghent University,
Sint-Pietersnieuwstraat 41, B-9000, Ghent, Belgium
Christophe.Billiet@UGent.be, Guy.DeTre@UGent.be

[2] Department of Computer Science and Artificial Intelligence, University of Granada,
C/Periodista Daniel Saucedo Aranda, S/N, E-18071, Granada, Spain
jpons@decsai.ugr.es, opc@decsai.ugr.es

Abstract. Databases model parts of reality by containing data representing properties of real-world objects or concepts. Often, some of these properties are time-related. Thus, databases often contain data representing time-related information. However, as they may be produced by humans, such data or information may contain imperfections like uncertainties. An important purpose of databases is to allow their data to be queried, to allow access to the information these data represent. Users may do this using queries, in which they describe their preferences concerning the data they are (not) interested in. Because users may have both positive and negative such preferences, they may want to query databases in a bipolar way. Such preferences may also have a temporal nature, but, traditionally, temporal query conditions are handled specifically. In this paper, a novel technique is presented to query a valid-time relation containing uncertain valid-time data in a bipolar way, which allows the query to have a single bipolar temporal query condition.

Keywords: Bipolar Querying, Valid-time Relation, Valid Time, Temporal Databases, Uncertainty, Possibility Theory, Ill-known Intervals

1 Introduction

Generally, database systems model (parts of) reality. For this, their databases contain data representing properties of real-world objects or concepts. Some essential properties of real-world objects or concepts are time-related. Thus, databases often contain data representing temporal values [1], which are basically indications of time and describe such properties. These temporal values are usually either time intervals [1] or instants [1], which may informally be seen as infinitesimally short 'periods' or 'points' in time. Based on their interpretation and purpose, temporal values can be classified into several categories, but the presented work will only consider valid-time indications, which indicate when corresponding data is a valid or true representation of the reality modelled by its database [1–4].

A lot of database data are produced by humans, but human-made data are prone to imperfections, as some of these data may be vague or imprecise [5], contradictory, incomplete or uncertain [3], [4], [6]. Of course, data representing temporal values may contain such imperfections too [2–4], [7]. The work presented in this paper will consider databases containing data representing valid-time intervals subject to uncertainty and will assume all non-temporal data in these databases to contain no imperfections.

One of the most important purposes of a database is to allow its data to be queried, to allow the information or knowledge represented by this data to be retrieved. A user may query a database in a 'regular' way: the user describes the data which he or she finds desired or satisfactory and thus wants to retrieve, by perfectly describing the allowed values of these data. A user may also query a database in a 'fuzzy' way: the user describes the data which he or she finds desired or satisfactory by imperfectly describing the allowed values of these data [8]. These imperfect descriptions may contain vagueness or imprecision, often through the use of linguistic terms [9], [10]. A user may also query a database in a 'bipolar' way. Generally, two main approaches to this exist. One is for the user to describe the data which he or she finds acceptable and to describe the data among this acceptable data, which he or she finds really desired, both by describing the allowed values of these data [11]. The other is for the user to independently describe both the data which he or she finds desired or satisfactory and the data which he or she finds undesired or unsatisfactory, both by describing the allowed values of these data [12], [13]. The descriptions used in bipolar querying may contain imperfections. The presented work will only consider the latter approach to bipolar querying and will allow a simple form of imprecision in non-temporal elementary query conditions.

Compared to non-temporal user preferences, temporal user preferences usually have an uncommon nature and interpretation and thus expressing them relies on uncommon mechanics in querying: users usually prefer using specific temporal operators to express temporal preferences. Hence, several proposals have considered specific sets of temporal operators, often based on the possible temporal relationships between two time indications [2], [14], [15]. Such temporal relationships define semantically meaningful relationships with a temporal nature, between two time indications. In [16], a collection of temporal relationships between two time intervals (and as a special case instants) is introduced and this collection is considered groundbreaking. Of course, to query temporal data in a fuzzy way, fuzzy variants of such temporal operators are necessary and several proposals have thus introduced such variants [4], [14], often based on fuzzy variants of such temporal relationships [15], [17].

Techniques for the regular or fuzzy querying of valid-time databases containing valid-time data subject to uncertainty are considered by several existing proposals [2], [4], [17]. However, to the knowledge of the authors, only one proposal has considered the bipolar querying of valid-time databases (in [18], a technique is proposed to query a valid-time database containing temporal data subject to imprecision in a bipolar way) and none have considered the bipolar querying of valid-time databases containing temporal data subject to uncertainty. Thus, this paper presents a novel technique to query a valid-time relation containing valid-time data subject to uncertainty in a bipolar way, which allows the user to specify a single bipolar temporal query condition. This paper is structured as follows: in section 2, some preliminary concepts and techniques are described, in section 3, the novel technique which is the main contribution of the work presented in this paper, is explained and in section 4, the conclusions of this paper and some directions for future research are given.

2 Preliminaries

2.1 General Preliminaries, Notations and Nomenclature

Databases may contain data representing temporal values. Based on their purpose and interpretation, such time indications can be classified into different categories [1], [3]. The work presented in this paper only considers temporal values of the category *valid time*. Their purpose or interpretation is for every valid-time indication to correspond to a collection of data and to indicate a period of time during which this data is a valid or true representation of reality.

The work presented in this paper concerns time indications subject to uncertainty. This uncertainty is always assumed to be caused by a (partial) lack of knowledge: the exact, intended time indication is not known, eventhough there is only one time indication intended and as such no variability. Confidence about exactly which time indication is the intended one in the context of such uncertainty is modelled using possibility theory [17], [19]. In the presented work, 'possibility' and 'necessity' are always interpreted as measures of plausibility, respectively necessity, given all available knowledge. Time intervals not subject to any imperfection are called *crisp time intervals* (CTI) in this paper.

2.2 Valid-Time Relations

A *valid-time relation* (VTR) always has *valid-time attributes*. These are attributes describing a single valid time [1] for the objects or concepts represented by the VTR's tuples. A VTR may contain different tuples corresponding to the same real-world concept or object. The non-valid-time attribute values of such a tuple represent the capacities of the properties described by their corresponding attributes which were, are or will be true or valid for the object or concept corresponding to the tuple during the period in time indicated by the valid-time indication represented by the tuple's valid-time attribute values. Thus, such a tuple represents the 'version' of the object or concept corresponding to this tuple which was, is or will be real or valid or the 'state' this object or concept was, is or will be in, during the period in time indicated by the valid-time indication represented by the tuple's valid-time attribute values. In the presented work, such valid-time indications will always be time intervals [1] and will always be referred to as *valid-time intervals* (VTI).

2.3 Uncertainty in Valid-Time Intervals

The presented work allows VTI to be subject to uncertainty, by allowing them to be *ill-known valid-time intervals* (IKVTI). The concept of IKVTI is based on the concepts of *possibilistic variables* (PV) and *ill-known intervals* (IKI) [3], [17], [20].

Definition 1. *A* possibilistic variable *(PV) X on a universe U is a variable taking exactly one value in U, but for which this value is (partially) unknown. The possibility distribution π_X on U models the available knowledge about the value that X takes: for each $u \in U$, $\pi_X(u)$ represents the possibility that X takes the value u.*

Now consider a set U containing single values (and not collections of values). When a PV X_v is defined on such a set U, the unique value X_v takes, which is (partially) unknown, will be a single value in U and is called an *ill-known value* (IKV) in U [3], [17], [20]. In this paper, IKV will be denoted using lower-case letters. The work presented in this paper uses a specific kind of IKI, defined as follows, although other definitions exist [3], [4].

Definition 2. *Consider an ordered set U. An* ill-known interval *(IKI) in U is an interval in U of which both boundary values are IKV in U.*

Specifically concerning valid time, an IKI in a time domain represented by the domains of a VTR's valid-time attributes is called an *ill-known valid-time interval* (IKVTI). The work presented in this paper requires the possibility distributions defining an IKVTI's IKV to be convex [17]. In this paper, an IKVTI with boundary IKV s and e will be noted $[s, e]$.

2.4 Evaluation of Temporal Relationships

To express temporal elementary query conditions, operators based on temporal relationships are necessary. In the presented work, only Allen relationships [16] between a CTI and a IKVTI are considered. To evaluate such relationships, the *ill-known constraints* (IKC) framework presented in [17] is used. It relies on the concept of IKC.

Definition 3. *Given an ordered set U, an* ill-known constraint *(IKC) $C = (R, v)$ on U is specified by means of a binary relation $R \subseteq U^2$ and a fixed IKV v in U. Any set $A \subseteq U$ now satisfies IKC $C = (R, v)$ if and only if:*

$$\forall a \in A : (a, v) \in R$$

The satisfaction of an IKC C by a set A will be noted $C(A)$ in this paper. Consider an ordered set U, an IKC $C = (R, v)$ on U and a set $A \subseteq U$. Due to the uncertainty inherent to v, it is uncertain whether A satisfies C or not. The degree of possibility $\text{Pos}(C(A))$ that A satisfies C and the degree of necessity $\text{Nec}(C(A))$ that A satisfies C, can be calculated as follows [17]:

$$\text{Pos}(C(A)) = \min_{a \in A} \left(\sup_{(a,w) \in R} \pi_{X_v}(w) \right) \tag{1}$$

$$\text{Nec}(C(A)) = \min_{a \in A} \left(\inf_{(a,w) \notin R} 1 - \pi_{X_v}(w) \right) \tag{2}$$

Given an ordered set U, degrees of possibility and necessity that a set $A \subseteq U$ satisfies a boolean combination of IKC on U can be found by using the possibilistic extensions of boolean operators 'and' ($\wedge$), 'or' ($\vee$) and 'not' ($\neg$), as described in [3], [4], [17].

The IKC framework now allows evaluating a given Allen relationship AR between a given CTI I and a given IKVTI $J = [s, e]$ by allowing the calculation of the degrees of possibility and necessity that $I\ AR\ J$ holds. For this, the combination of AR and J is translated to a specific boolean combination of specific IKC. These translations are shown in table 1. Every row of this table corresponds to a given Allen relationship between I and J, indicated by the row's value in the 'Allen Relationship' column. The collections of specific IKC for given Allen relationships are shown in the 'Constraints' column (every $C_i, i \in \{1, 2, 3, 4\}$ denotes an IKC) and the specific combination of these IKC used for evaluation of the Allen relationships are shown in the 'Combination' column. Finally, the degrees of possibility and necessity that $I\ AR\ J$ holds are then the degrees of possibility, respectively necessity that I satisfies the specific aggregation of specific IKC found as translation of the combination of AR and J. Using the formulas shown above, the requested possibility and necessity degrees can be calculated from these.

2.5 Bipolar Querying

As mentioned before, humans may express their query preferences using both positive and negative query conditions [12], [13]. If the semantics of these con-

Table 1. The translations of Allen relationships to the IKC framework

Allen Relationship	**Constraints**	**Combination**
I before J	$C_1 \triangleq (<, s)$	$C_1(I)$
I equal J	$C_1 \triangleq (\geq, s)$, $C_2 \triangleq (\neq, s)$ $C_3 \triangleq (\leq, e)$, $C_4 \triangleq (\neq, e)$	$C_1(I) \wedge \neg C_2(I) \wedge$ $C_3(I) \wedge \neg C_4(I)$
I meets J	$C_1 \triangleq (\leq, s)$ $C_2 \triangleq (\neq, s)$	$C_1(I) \wedge \neg C_2(I)$
I overlaps J	$C_1 \triangleq (<, e)$, $C_2 \triangleq (\leq, s)$, $C_3 \triangleq (\geq, s)$	$C_1(I) \wedge \neg C_2(I) \wedge \neg C_3(I)$
I during J	$C_1 \triangleq (>, s)$, $C_2 \triangleq (\leq, e)$ $C_3 \triangleq (\geq, s)$, $C_4 \triangleq (<, e)$	$\big(C_1(I) \wedge C_2(I)\big) \vee$ $\big(C_3(I) \wedge C_4(I)\big)$
I starts J	$C_1 \triangleq (\geq, s)$, $C_2 \triangleq (\neq, s)$	$C_1(I) \wedge, \neg C_2(I)$
I finishes J	$C_1 \triangleq (\leq, e)$, $C_2 \triangleq (\neq, e)$	$C_1(I) \wedge \neg C_2(I)$

ditions are non-symmetric, meaning that the positive preferences can not be derived from the negative or vice versa, the bipolarity in this query is called *heterogenous* [13]. The presented work will concern only such heterogenous query bipolarity.

A query usually takes the form of a boolean combination of elementary query conditions. Every elementary query condition then expresses the user's demands concerning a single attribute. Bipolarity in a query can either be specified between or inside the elementary query conditions. In [13], it is shown that combining both approaches makes no sense and that the approach where bipolarity is specified inside elementary query conditions, using intuitionistic fuzzy sets [21], is a more intuitive one. In the presented work, only the latter approach is used. In this approach, elementary query conditions express both what is accepted and what is not accepted by the query, at once, and are called *bipolar query conditions* (BQC) [13].

Consider a relation attribute A. Let dom_A be the domain of A's data type, let μ_{c_A} and ν_{c_A} be membership functions from dom_A to the unit interval $[0,1]$, where $\mu_{c_A}(x)$ represents to what extent $x \in dom_A$ is satisfactory and $\nu_{c_A}(x)$ to what extent x is unsatisfactory to a user, then a BQC c_A expressing this user's preferences concerning A can be modelled by an Intuitionistic Fuzzy Set (IFS) [21] as [12], [13]:

$$c_A = \{(v, \mu_{c_A}(v), \nu_{c_A}(v)) : v \in dom_A\} \tag{3}$$

Note that to allow overspecification of the user's preferences, the IFS's consistency condition can be relaxed, which means that there may exist values $v \in dom_A$ for which $\mu_{c_A}(v) + \nu_{c_A}(v) > 1$ [12], [13].

If the user explicitely defines μ_{c_A}, but doesn't define ν_{c_A}, then ν_{c_A} will be assumed to be the inverse of μ_{c_A} [13]. If the user explicitely defines ν_{c_A}, but doesn't define μ_{c_A}, then μ_{c_A} will be assumed to be the inverse of ν_{c_A} [13]. Thus, in the absence of clear heterogenousness of the bipolarity in a query condition, the bipolarity will be assumed homogenous [13].

The evaluation of a BQC results in a so-called *bipolar satisfaction degree* (BSD) [13], which is a pair

$$(s, d),\ s, d \in [0,1]$$

where s is called the *satisfaction degree* and d is called the *dissatisfaction degree* [13]. Here, s and d are independent from each other and express to which extent the BSD respectively represents 'satisfied' and 'dissatisfied' [13]. Extreme values for s and d are 0 ('not at all') and 1 ('fully'). For example: the BSD $(1,0)$ represents 'fully satisfied, not dissatisfied at all' [13].

As explained in [13], there is no consistency condition for BSD's and for a BSD (s, d), $s + d > 1$ is allowed. The motivation is that BSD's try to reflect heterogenous bipolarity in human reasoning, which can sometimes be inconsistent.

In general, the evaluation of a BQC c_A on relation attribute A for a tuple r will result in a BSD $(s^r_{c_A}, d^r_{c_A})$, which is calculated as follows. Let $r[A]$ denote the value of tuple r for attribute A, then [13]:

$$(s^r_{c_A}, d^r_{c_A}) = (\mu_{c_A}(r[A]), \nu_{c_A}(r[A])) \tag{4}$$

Remark that the traditional approach to fuzzy querying using regular fuzzy sets can be obtained from this as a special case, where the bipolarity involved is homogenous. In that case, a user only specifies positive query preferences [13].

3 A Novel Querying Approach

3.1 Valid-Time Relations Subject to Uncertainty

The presented proposal will concern VTR where the VTI are IKVTI. Generally, such a VTR R can be seen as constructed in the following way. Let R have n non-temporal attributes $A_i, 1 \leq i \leq n, i \in \mathbb{N}$ and two valid-time attributes VST and VET. Every tuple T of R represents an object or concept version or state which is valid during the time period indicated by the tuple's IKVTI I_T. This IKVTI I_T is now defined by two IKV, which respectively describe the starting and ending instants of I_T and are represented by the tuple's VST, respectively VET values. The interpretation is that the version or state corresponding to a tuple was, is or will be valid during a period of time, but exactly which period this is intended to be, is unknown. Confidence about exactly which period is intended, is modelled by the tuple's IKVTI [2], [4].

Table 2. The example relation used in this paper

ID	Author	VST	VET
1	Aloïsius	$[4/4/1208, 6/4/1208, 16/4/1208]$	$[10/12/1208, 1/1/1209, 26/1/1209]$
2	Theofilus	$[2/4/1209, 12/4/1209, 22/4/1209]$	$[21/12/1209, 1/1/1210, 21/1/1210]$
3	Gerardus	$[14/1/1209, 15/1/1209, 16/1/1209]$	$[21/12/1209, 15/1/1210, 25/1/1210]$
4	Euforius	$[21/12/1210, 1/1/1211, 11/1/1211]$	$[21/12/1211, 1/1/1212, 11/1/1212]$
5	Ambrosius	$[11/12/1213, 21/12/1213, 15/1/1214]$	$[9/10/1216, 10/10/1216, 15/10/1216]$
6	Aloïsius	$[21/12/1213, 1/1/1214, 11/1/1214]$	$[9/6/1217, 9/6/1217, 12/6/1217]$
7	Gerardus	$[29/12/1214, 1/1/1215, 8/1/1215]$	$[9/6/1217, 10/6/1217, 12/6/1217]$

In this paper, the relation shown in table 2 will be used as example relation. This relation models the being in effect of medieval legal acts. Since the properties of a legal act cannot change once it has taken effect, every legal act has only one version or state. This was deliberately done to simplify the example. Thus, every tuple of the relation corresponds to a legal act. A tuple's value for attribute 'ID' is a number uniquely identifying the legal act corresponding to the tuple. A tuple's value for attribute 'Author' is a character string representation of the name of the author of the legal act corresponding to the tuple. A tuple's IKVTI represents the time period during which the act corresponding to the tuple was in effect. For this, every value for VST, respectively VET represents an IKV describing the day on which the legal act respectively took effect and stopped taking effect. For this, every value for VST or VET is a triple $[d_1, d_2, d_3]$, where d_1, d_2, d_3 are elements of the ordered set of days in history. Such a triple $[d_1, d_2, d_3]$ now defines a triangular (and thus convex) possibility distribution π which defines the mentioned IKV and which is defined by (differences in dates in this function prescription are expressed in amounts of days):

$$\pi(x) = \begin{cases} \frac{x-d_1}{d_2-d_1}, & \text{if } d_1 \leq x < d_2 \\ \frac{d_3-x}{d_3-d_2}, & \text{if } d_2 \leq x \leq d_3 \\ 0, & \text{else} \end{cases} \tag{5}$$

3.2 Querying Using Bipolar Valid-Time Conditions

The presented work introduces a novel querying technique. The most interesting aspect of this technique is that it allows the user to specify a bipolar valid-time demand. According to this technique, a user query Q consists of two separate parts Q_n and Q_t: $Q = (Q_n, Q_t)$. Here, Q_n is a boolean combination of BQC on non-valid-time attributes, expressing the user's non-temporal demands. Q_t expresses the user's valid-time demands and is a single crisp temporal BQC $((AR_+, I_+), (AR_-, I_-))$, where both AR_+ and AR_- are Allen relationships and both I_+ and I_- are CTI. The interpretation is that the user requires an object or concept that has a version or state that complies with his or her non-temporal demands and was, is or will be valid during a time interval which is in Allen relationship AR_+ with I_+ and wasn't, isn't or won't be valid during a time interval which is in Allen relationship AR_- with I_-.

Consider the example relation shown in table 2. Now assume a user queries the relation to find all legal acts of which the author is preferably named Aloïsius, less preferably Euforius and perhaps Eugenius, and of which the author is preferably not named Ambrosius, rather not Theofilus and perhaps not Antonius and which took effect preferably before 2/1/1210 and preferably not after 1/1/1214. These demands can now be translated to a query Q_{ex} in the following way:

$$\begin{aligned} Q_{ex} &= (Q_{n,ex}, Q_{t,ex}) \;=\; (Q_{n,ex}, ((AR_{+,ex}, I_{+,ex}), (AR_{-,ex}, I_{-,ex}))), \\ Q_{n,ex} &= \{(x, \mu_{ex}(x), \nu_{ex}(x)), \forall x \in \mathbb{S}\} \\ AR_{+,ex} &= \; AR_{-,ex} \; = \; DURING \\ I_{+,ex} &= \;]-infinity, 1/1/1210] \,, \quad I_{-,ex} \; = \; [1/1/1214, +infinity[\end{aligned}$$

and μ_{ex} and ν_{ex} are the membership functions of the fuzzy sets:

$$\{(Aloisius, 1), (Euforius, 0.7), (Eugenius, 0.1)\}, \text{ respectively}$$
$$\{(Ambrosius, 1), (Theofilus, 0.7), (Antonius, 0.1)\}$$

and $\mathbb{S}$ is the set of all author names in the example relation's 'Author' attribute domain.

3.3 Elementary Query Condition Evaluation

Generally, a first step in determining which objects or concepts corresponding to tuples to present to a user as answer to his or her query, is evaluating the query's elementary conditions for every tuple. In the presented work, given a user query Q constructed as proposed in section 3.2 and using the same notations, the following is done separately for every tuple T of VTR R:

- every non-temporal BQC in Q_n is evaluated as described in [13], resulting in a BSD for each. The interpretation is as described in [13]: the BSD's satisfaction, respectively dissatisfaction degrees express to which extent T satisfies, respectively dissatisfies the user preferences expressed by the BQC.
- the query's temporal BQC Q_t is evaluated as follows. Let I_T be the tuple's IKVTI. Then, independently, the statements 'I_T AR_+ I_+', respectively 'I_T AR_- I_-' are evaluated using the IKC framework as described in [2], [4], [17]. These evaluations result in a possibility degree $Pos_+(I_T)$ and a necessity degree $Nec_+(I_T)$, respectively a possibility degree $Pos_-(I_T)$ and a necessity degree $Nec_-(I_T)$. The interpretation is that $Pos_+(I_T)$ and $Nec_+(I_T)$ express the possibility, respectively necessity, that the time interval during which the version or state represented by T is valid and which is intended by I_T, is in relationship AR_+ with I_+ and thus complies with the user's positive temporal demand. Furthermore, the interpretation is that $Pos_-(I_T)$ and $Nec_-(I_T)$ express the possibility, respectively necessity, that the time interval during which the version or state represented by T is valid and which is intended by I_T, is in relationship AR_- with I_- and thus complies with the user's negative temporal demand.

Table 3 shows the results of the evaluation of the example query's elementary query conditions for the tuples of the example relation.

Table 3. Elementary query condition evaluation results for the example relation

ID	$BSD_{Q_{N,ex}}(T)$	$(Pos_+(I_T), Nec_+(I_T))$	$(Pos_-(I_T), Nec_-(I_T))$
1	$(1, 0)$	$(1, 1)$	$(0, 0)$
2	$(0, 0.7)$	$(1, 0)$	$(0, 0)$
3	$(0, 0)$	$(11/25, 0)$	$(0, 0)$
4	$(0.7, 0)$	$(0, 0)$	$(0, 0)$
5	$(0, 1)$	$(0, 0)$	$(14/25, 0)$
6	$(1, 0)$	$(0, 0)$	$(1, 0)$
7	$(0, 0)$	$(0, 0)$	$(1, 1)$

3.4 Aggregation and Ranking

Generally, a second step in determining which objects or concepts corresponding to tuples to present to a user as answer to his or her query, is aggregating, for every tuple, the tuple's evaluation results for the query's elementary conditions, in order to determine how well the tuple complies with the entire user request expressed by the combination of the elementary query conditions. Usually, these evaluation results are quantifications of (dis)satisfaction. However, in the presented proposal, two different types of evaluation results can be discerned:

- the BSD's (dis)satisfaction degrees constitute quantifications of (dis)satisfaction: they quantify to which extent a tuple's attribute values (dis)satisfy a user's non-temporal preferences and thus assess an answer to the question: 'To what extent does a version or state represented in the relation (dis)satisfy the user's request and could thus be a (un)wanted result?'

- the possibility and necessity degrees Pos_+, Nec_+, respectively Pos_-, Nec_- constitute quantifications of possibility and necessity: they quantify the possibility that a tuple's intended crisp VTI does (not) comply with the user's temporal demands by quantifying confidence about exactly which crisp VTI is the tuple's intended VTI. Thus, they assess an answer to the question: 'Given all available knowledge, how plausible is it that a version or state represented in the relation (that may or may not (dis)satisfy the user's non-temporal preferences) actually existed, exists or will exist during the time period indicated by the user?'

A fundamental question arises now: should one consider combining quantifications of these different categories? On one hand, such quantifications have clearly different semantics and it would not be clear what exactly the meaning would be of the result of such a combination or what the semantically most coherent ways would be to further process such combination results. Thus, it is important, for every query result tuple presented to the user, to certainly keep both different types of evaluation results as separate (meta)data. On the other hand, without an unambiguous and straightforward ranking of the query result tuples, the user cannot clearly discern the result tuples which comply well with his or her demands from those which don't. This would defeat the purpose of querying. Reasonably, such a ranking should be based on the elementary query condition evaluation results. As there cannot exist a ranking between quantifications of categories with different semantics, a combination of quantifications of satisfaction and possibility seems to be required. In most existing proposals requiring a combination of quantifications of satisfaction and possibility, both quantifications are combined as to restrict one another. The result is usually seen as a quantification of possibility. Below, this approach is translated to the specific situation encountered in the presented work.

For every tuple T, the BSD's which are the evaluation results of the non-temporal BQC are combined to a single BSD $(s_n(T), d_n(T))$ as described in [13]. In this combination method, satisfaction degrees are combined with each other and separately, dissatisfaction degrees are combined with each other. This reasoning is now extended to include $Pos_+(I_T)$, $Nec_+(I_T)$, $Pos_-(I_T)$ and $Nec_-(I_T)$: a couple $(Pos_+(T), Nec_+(T))$, respectively $(Pos_-(T), Nec_-(T))$ is calculated, expressing the possibility and necessity that the version or state corresponding to T complies with all of the user's positive, respectively negative demands. This calculation is done as follows:

$$
\begin{aligned}
Pos_+(T) &= \min(s_n(T), Pos_+(I_T))\\
Nec_+(T) &= \begin{cases} 0, & \text{if } Pos_+(T) < 1 \\ \min(s_n(T), Nec_+(I_T)), & \text{else} \end{cases}\\
Pos_-(T) &= \min(d_n(T), Pos_-(I_T))\\
Nec_-(T) &= \begin{cases} 0, & \text{if } Pos_-(T) < 1 \\ \min(d_n(T), Nec_-(I_T)), & \text{else} \end{cases}
\end{aligned}
$$

Next, a tie-break approach is used to rank the versions or states represented in R: ranking is done based on the value of $Pos_+(T)$, with $Nec_+(T)$ as tie-breaker for $Pos_+(T)$, with $Pos_-(T)$ as tie-breaker for $(Pos_+(T), Nec_+(T))$ and finally $Nec_-(T)$ as tiebreaker for $(Pos_+(T), Nec_+(T), Pos_-(T))$. The results of this aggregation and ranking approach for the example relation and query are shown in table 4.

The introduced technique for aggregating elementary query condition evaluation results and determining a ranking does not require the combination of quantifications with different interpretations and still presents the query result tuples in a ordered manner, where this ordering is consistent with the expected extend to which each tuple is usefull to the user. However, the technique still requires the user to decide which objects or concepts constitute the best query answer, although this decision is heavily supported.

Table 4. Aggregation and ranking results for the example relation and query

ID	$Pos_+(T)$	$Nec_+(T)$	$Pos_-(T)$	$Nec_-(T)$	$BSD_{Q_{n,ex}}$	$(Pos_+(I_T), Nec_+(I_T))$	$(Pos_-(I_T), Nec_-(I_T))$
1	1	1	0	0	$(1, 0)$	$(1, 1)$	$(0, 0)$
2	0	0	0	0	$(0, 0.7)$	$(1, 0)$	$(0, 0)$
3	0	0	0	0	$(0, 0)$	$(11/25, 0)$	$(0, 0)$
4	0	0	0	0	$(0.7, 0)$	$(0, 0)$	$(0, 0)$
6	0	0	0	0	$(1, 0)$	$(0, 0)$	$(1, 0)$
7	0	0	0	0	$(0, 0)$	$(0, 0)$	$(1, 1)$
5	0	0	14/25	0	$(0, 1)$	$(0, 0)$	$(14/25, 0)$

4 Conclusions and Future Work

In this paper, a novel technique to query a valid-time relation containing valid-time data subject to uncertainty in a bipolar way, is presented. This technique allows the user to specify a single valid-time bipolar query condition. A major issue concerning the need to combine quantifications of (dis)satisfaction with quantifications of possibility resulting from this technique is presented and shortly discussed, along with a solution for this issue. In the near future, the interactions between these types of quantifications and between uncertainty in valid-time data and bipolar querying will be further studied. Also, an approach to allow the valid-time bipolar query condition to be fuzzy will be considered.

References

1. Böhlen, M.H., Clifford, J., Elmasri, R., Gadia, S.K., Grandi, F., Hayes, P., Jajodia, S., Käfer, W., Kline, N., Lorentzos, N., Mitsopoulos, Y., Montanari, A., Nonen, D., Peressi, E., Pernici, B., Roddick, J.F., Sarda, N.L., Scalas, M.R., Segev, A., Snodgrass, R.T., Soo, M.D., Tansel, A.U., Tiberio, P., Wiederhold, G.: The Consensus Glossary of Temporal Database Concepts-February 1998 Version. In: Lecture Notes in Computer Science. Volume 1399/1998. (1998) 367–405
2. Pons, J.E., Marín, N., Pons, O., Billiet, C., De Tré, G.: A Relational Model for the Possibilistic Valid-time Approach. International Journal of Computational Intelligence Systems **5**(6) (2012) 1068–1088

3. Billiet, C., Pons, J.E., Pons Capote, O., De Tré, G.: Evaluating Possibilistic Valid-Time Queries. In: Communications in Computer and Information Science. Volume 297. (2012) 410–419
4. Pons, J.E., Billiet, C., Pons Capote, O., De Tré, G.: A Possibilistic Valid-Time Model. In: Communications in Computer and Information Science. Volume 297. (2012) 420–429
5. Medina, J.M., Pons, O., Amparo Vila, M.: Gefred: A Generalized Model of Fuzzy Relational Databases. Information Sciences **76**(1-2) (1994) 87–109
6. Bosc, P., Pivert, O.: Modeling and Querying Uncertain Relational Databases: A Survey of Approaches Based on the Possible Worlds Semantics. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **18**(5) (2010) 565–603
7. Dyreson, C.E., Snodgrass, R.T.: Supporting Valid-Time Indeterminacy. ACM Transactions on Database Systems **23**(1) (1998) 1–57
8. Zadrozny, S., De Tré, G., De Caluwe, R., Kacprzyk, J.: An overview of fuzzy approaches to flexible database querying. In: Handbook of Research on Fuzzy Information Processing in Databases. IGI Global (2008)
9. Devos, F., Maesfranckx, P., De Tré, G.: Granularity in the Interpretation of Around in Approximative Lexical Time Indications. Journal of Quantitative Linguistics **5**(3) (1998) 167–173
10. Kacprzyk, J., Zadrozny, S.: Computing with Words in Intelligent Database Querying: Standalone and Internet-based Applications. Information Sciences **134**(1-4) (2001) 71–109
11. Dubois, D., Prade, H.: Bipolarity in Flexible Querying. In: LNCS. Volume 2522. (2002) 174–182
12. De Tré, G., Zadrozny, S., Bronselaer, A.: Handling Bipolarity in Elementary Queries to Possibilistic Databases. IEEE Transactions on Fuzzy Systems **18**(3) (2010) 599–612
13. Matthé, T., De Tré, G., Zadrozny, S., Kacprzyk, J., Bronselaer, A.: Bipolar Database Querying Using Bipolar Satisfaction Degrees. International Journal of Intelligent Systems **26**(10) (2011) 890–910
14. Galindo, J., Medina, J.M.: FTSQL2: Fuzzy Time in Relational Databases*. In: Proceedings of the 2nd International Conference in Fuzzy Logic and Technology. (2001) 47–50
15. Schockaert, S., De Cock, M., Kerre, E.E.: Fuzzifying Allen's Temporal Interval Relations. IEEE Transactions on Fuzzy Systems **16**(2) (2008) 517–533
16. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM **26**(11) (1983) 832–843
17. Pons, J.E., Bronselaer, A., De Tré, G., Pons, O.: Possibilistic evaluation of sets. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (2012) Accepted for publication in the International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.
18. Billiet, C., Pons, J.E., Matthé, T., De Tré, G., Pons Capote, O.: Bipolar Fuzzy Querying of Temporal Databases. In: Lecture Notes in Artificial Intelligence. (2011) 60–71
19. Dubois, D., Prade, H.: Possibility Theory. Plenum Press (1988)
20. Dubois, D., Prade, H.: Incomplete Conjunctive Information. Computers & Mathematics with Applications **15**(10) (1988) 797–810
21. Atanassov, K.T.: Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems **20**(1) (1986) 87–96