Fuzzy and Uncertain Spatio-Temporal Database Models: A Constraint-Based Approach

G. de Tré, R. de Caluwe, A. Hallez, J. Verstraete
Ghent University, Dept. of Telecommunications and Information Processing
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
Guy.DETRE@rug.ac.be

Abstract

In this paper a constraint-based generalised object-oriented database model is adapted to manage spatio-temporal information. This adaptation is based on the definition of a new data type, which is suited to handle both temporal and spatial information. Generalised constraints are used to describe spatio-temporal data, to enforce integrity rules on databases, to specify the formal semantics of a database scheme and to impose selection criteria for information retrieval.

Keywords: Spatio-temporal information modelling, object-oriented database model, constraints.

1 Introduction

A constraint can formally be seen as a relationship, which has to be satisfied. With respect to information and knowledge-based systems, constraints are considered to be an important and adequate means to define the semantics and the integrity of the data [1, 2, 3, 4, 5]. This is especially true for spatial data and for temporal data. An instance then belongs to the information or knowledge base in as far as it satisfies all of its defining constraints.

In practice, spatial data usually consist of line segments, and therefore linear arithmetic constraints are particularly appropriate for representing such data [5]. For example, if spatial geographical information is handled, constraints can be used to define the borders of a country, a city, a region, to define a river, a highway, etc. This is illustrated in Figure 1 and in Table 1, which respectively represent a map of France (a real map will be defined by many more constraints, but the basic ideas are the same) and some geometrical descriptions.

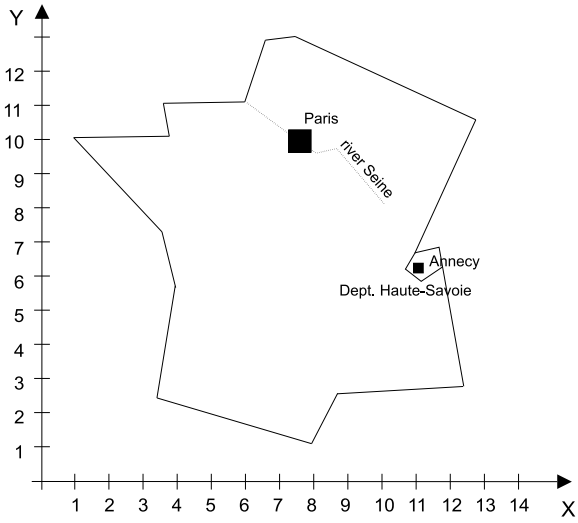

Figure 1: Spatial information: map of France

Constraints can also be used to impose selection criteria for information retrieval. In this case, each constraint defines a condition for the instances to belong to the result of the retrieval [6, 7]. Every instance belongs to the result in as far as it satisfies all the imposed criteria. For example, if someone wants to retrieve all the young persons who live in

Table 1: Geometrical descriptions.

Annecy: $(x \geq 10.9) \wedge (x \leq 11.1) \wedge (y \geq 6)$ $\wedge (y \leq 6.2)$
Seine: $((y \leq 11) \wedge (y + 0.7x = 15.2) \wedge (y \geq 9.6))$ $\vee((y \leq 9.7) \wedge (y - 0.2x = 8) \wedge (y \geq 9.6))$ $\vee((y \leq 8) \wedge (y + 1.2x = -20) \wedge (y \geq 9.7))$
Haute-Savoie: $(y - 0.1x \leq 8) \wedge (y + 7x \leq 89.6) \wedge$ $(y - 0.4x \leq 1.5) \wedge (y + 0.2x \geq 8.2) \wedge$ $(y - 4x \geq 36.7)$

Annecy, two constraints can be imposed: one that selects all the young persons and another that selects all persons living in Annecy.

Spatio-temporal information can be fuzzy and/or uncertain [8, 9]. There has been a considerable amount of research regarding fuzziness in spatio-temporal databases [8, 10, 11, 12]. In this paper an extension of a constraint-based fuzzy object-oriented database model [3] is presented. This extension is based on the introduction of a new data type and on the generalisation of linear arithmetic constraints.

In the following section, the main concepts of the fuzzy object-oriented database model are introduced. The modelling of fuzzy spatio-temporal information, by means of generalised linear arithmetic constraints, is discussed in Section 3. Finally, the achieved results and future developments are summarized in the concluding section.

2 Generalised object-oriented database model

The employed fuzzy object-oriented database model [3, 14] has been obtained as a generalisation of a crisp object-oriented database model that is consistent with the ODMG de facto standard [13]. The model is build upon a generalised algebraic type system and a generalised constraint system, which are both used for the definition of so-called generalised object schemes and generalised database schemes.

2.1 Type system

To support the definition of types, a (generalised) type system GTS has been built [3]. In order to be consistent with the ODMG data model, the type system supports the generalised definitions of literal types, object types and reference types (which enable to refer to the instances of object types and are used to formalize the binary relationships between the object types in the database scheme).

The semantic definitions of a type $\tilde{t}$ are based on domains, sets of domains and sets of operators (cf. [15]) and are determined by:

- a set of domains $D_{\tilde{t}}$
- a designated domain $dom_{\tilde{t}} \in D_{\tilde{t}}$
- a set of operators $O_{\tilde{t}}$ and
- a set of axioms $A_{\tilde{t}}$

The designated domain $dom_{\tilde{t}}$ is called the domain of the type $\tilde{t}$ and consists of the set of all the possible values for $\tilde{t}$. Every domain value is represented by a fuzzy set, which is defined over the domain of the corresponding ordinary type t. In order to deal with "undefined" values, a type specific bottom value $\perp_t$, has been added to the domain of every ordinary type t. The set of operators $O_{\tilde{t}}$ contains all the operators that are defined on the domain $dom_{\tilde{t}}$. The set of domains $D_{\tilde{t}}$ consists of all the domains that are involved in the definition of the operators of $O_{\tilde{t}}$, whereas the set of axioms $A_{\tilde{t}}$ contains all the axioms that are involved in the definition of the semantics of $O_{\tilde{t}}$.

The instances of a literal type, an object type and a reference type are respectively called literals, objects and reference instances. Every instance is characterised by its type and a domain value of this type (also called the state of the instance). Objects either have a transient or a persistent lifetime. Persistent objects are additionally characterised by a unique object identifier and a set of object names.

2.2 Constraint system

Constraints are used to enforce integrity rules on databases (e.g. domain rules, referential integrity rules, etc.) and to specify the formal semantics of the database scheme (e.g. the applicability of null values, the definition of keys, etc.). To support the definition of constraints, a (generalised) constraint system GCS has been built. The set of generalised constraint definitions supported by the constraint system can be partitioned into the subset of constraints which can be applied to objects independent of any existing database (e.g. domain constraints) and the subset of "database" constraints which are defined for database objects (e.g. referential integrity constraints) [3, 4].

The semantics of a constraint are defined by means of a function $\tilde{\rho}$, which associates with every object $\tilde{o}$ a fuzzy set

$$\{(True, \mu_{True}), (False, \mu_{False}), (\perp_{Boolean}, \mu_{\perp_{Boolean}})\}$$

which represents the extended possibilistic truth value [16] of the proposition

"the object $\tilde{o}$ satisfies the constraint $\tilde{\rho}$"

The membership degrees μ_{True} and μ_{False} indicate to which degree this proposition is respectively true and false. The membership degree $\mu_{\perp_{Boolean}}$ denotes to which degree the proposition is not applicable and is used to model those cases where the constraint $\tilde{\rho}$ is (partially) not applicable to $\tilde{o}$.

2.3 Object schemes

The full semantics of an object are described by an object scheme $\tilde{os}$. This scheme "in fine" completely defines the object, now including the definitions of the constraints that apply to it. Each object scheme is defined by an identifier id, an object type $\tilde{t}$, a "meaning" $\tilde{M}$ and a conjunctive fuzzy set of constraints $\tilde{C}_{\tilde{t}}$, which all have to be applied onto the objects of type $\tilde{t}$ independently of any existing database

$$\tilde{os} = [id, \tilde{t}, \tilde{M}, \tilde{C}_{\tilde{t}}]$$

The "meaning" $\tilde{M}$ is provided to add comments and is usually described in a natural language. The membership degree of an element of $\tilde{C}_{\tilde{t}}$ indicates to which degree the constraint applies to the object type $\tilde{t}$.

An instance $\tilde{o}$ of the object type $\tilde{t}$ is defined to be an instance of the object scheme $\tilde{os}$, if it satisfies (with a truth value which differs from $\{(False, 1)\}$) all the constraints of $\tilde{C}_{\tilde{t}}$ and all the constraints of the sets $\tilde{C}_{\hat{\tilde{t}}}$ of the object schemes, which have been defined for the supertypes $\hat{\tilde{t}}$ of $\tilde{t}$.

2.4 Database schemes

A database scheme $\tilde{ds}$ describes the full semantics of the objects which are stored in a generalised database and is defined by the quadruple

$$\tilde{ds} = [id, \tilde{D}, \tilde{M}, \tilde{C}_{\tilde{D}}]$$

in which id is the identifier of the database scheme,

$$\tilde{D} = \{\tilde{os}_i | 1 \leq i \leq n, i, n \in \mathbb{N}_0\}$$

is a finite set of object schemes, $\tilde{M}$ is provided to add comments, and $\tilde{C}_{\tilde{D}}$ is a conjunctive fuzzy set of "database" constraints, which imposes extra conditions on the instances of the object schemes of $\tilde{D}$ (e.g. referential constraints between two object schemes). Again, the membership degrees denote the relevance of the constraints. Every generalized object scheme in $\tilde{D}$ has a different object type. If an object scheme $\tilde{os} \in \tilde{D}$ is defined for an object type $\tilde{t}$ and $\tilde{t}'$ is a supertype of $\tilde{t}$, or $\tilde{t}'$ is an object type for which a binary relationship with $\tilde{t}$ is defined, then an object scheme $\tilde{os}' \in \tilde{D}$ has to be defined for $\tilde{t}'$.

Every persistent instance $\tilde{o}$ of an object scheme $\tilde{os} \in \tilde{D}$ of a database scheme $\tilde{ds}$ has to satisfy all the constraints of $\tilde{C}_{\tilde{D}}$, with a truth value which differs from $\{(False, 1)\}$.

2.5 Database Model

The generalised database model is finally obtained by extending the formalism with data definition (DDL) and data manipulation operators (DML) [14] (see Figure 2).

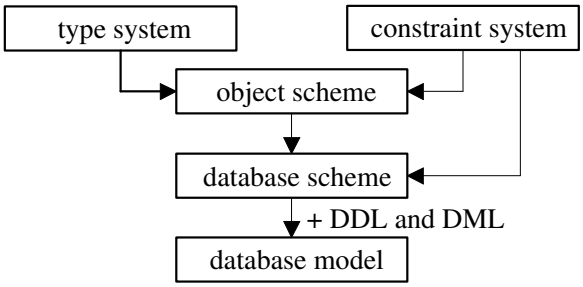

Figure 2: Generalised object-oriented database model: an overview

3 Modelling of spatio-temporal information

The generalised object-oriented database model presented in the previous section is extended in order to support the modelling of both temporal and spatial information. This is done by adding a new generic literal type *SpaceTime* to the type system. The domain of this new type consists of all the fuzzy sets which are defined over the points of a given geometrical space, which on its turn is defined by a finite number of axes, which all have only one point in common. Each axis either represents a time dimension or a spatial dimension. Generalised linear arithmetic constraints are defined and are used to describe the domain values of the *SpaceTime* type.

In the next subsection the focus is on the modelling of one-dimensional temporal information. In this special case, the literal type *SpaceTime* has to describe a temporal space, which is defined by one time axis. The modelling of spatial information is discussed in Subsection 3.2. The cases of one-dimensional, two-dimensional and n-dimensional spaces are handled. The formal definition of the *SpaceTime* type is given in Subsection 3.3.

3.1 Modelling of temporal information

In order to model temporal information, a new data type *TimeDim* is defined. This data type will not be included directly in the database model, but is necessary for the definition of the type *SpaceTime*.

The domain of *TimeDim* is defined by

$$dom_{TimeDim} = \mathbb{R} \cup \{\perp_{TimeDim}\}$$

where $\mathbb{R}$ denotes the set of real numbers and $\perp_{TimeDim}$ represents an "undefined" domain value.

The considered operators are the binary operators $=$, $\neq$, $<$, $>$, $\leq$, $\geq$, $+$, $-$, $*$ and $/$ and a null-ary operator $\perp$, which always results in an "undefined" domain value. When restricted to the set $dom_{TimeDim} \setminus \{\perp_{TimeDim}\}$, all binary operators have the same semantics as their counterparts within $\mathbb{R}^2$. For the bottom value $\perp_{TimeDim}$, the semantics are:

$$\forall\, x \in dom_{TimeDim} : op\,(x, \perp_{TimeDim}) = op\,(\perp_{TimeDim}, x) = \perp_{TimeDim}$$

where "*op*" is a variable copula whose successive values are respectively the symbols $=$, $\neq$, $<$, $>$, $\leq$, $\geq$, $+$, $-$, $*$ and $/$.

The type *TimeDim* can be employed to model time, hereby using the set $\mathbb{R}$ of real numbers as a representation of the continuum of physical time points [8]. However, in prospect of the generalisation discussed in Subsections 3.2 and 3.3, the type *SpaceTime* is introduced.

The type *SpaceTime* is structured and consists of a finite number of components. Each component either represents a temporal dimension or a spatial dimension. In this subsection only one (temporal) component is considered, so that the specification of *SpaceTime* is defined as:

$$SpaceTime\ id(id_1 : TimeDim)$$

where id is the identifier of the type and id_1 is the identifier of the component.

The domain of type

$$SpaceTime\ id(id_1 : TimeDim)$$

(shortly written as dom_{id}) is defined by

$$dom_{id} = \tilde{\wp}(\{(x) | x \in dom_{TimeDim}\}) \cup \{\perp_{SpaceTime}\}$$

where $\tilde{\wp}(U)$ denotes the set of all fuzzy sets, which can be defined over the universe U and

$\perp_{SpaceTime}$ represents an "undefined" domain value.

With the previous definition, every regular value of dom_{id} is a fuzzy set, which is defined over the continuum of physical time points. In order to describe the values of dom_{id}, linear arithmetic constraints are generalised. This is done by generalising the comparison operators $=$, $\leq$ and $\geq$.

Traditionally, these operators allow to describe crisp subsets of dom_{id}, e.g. $\forall\, t \in dom_{id}$, $x = t$ is a description of the fuzzy set $\{(t, 1)\}$, $x \leq t$ represents the set $\{(x, 1)|x \leq t\}$ and $x \geq t$ describes the set $\{(x, 1)|x \geq t\}$.

For the generalisation, a normalised fuzzy set $\tilde{V}$ has been associated with each operator. This fuzzy set is defined over the universe of valid distances —the set $\mathbb{R}^*$ of positive real numbers— and the boundary condition $\mu_{\tilde{V}}(0) = 1$ must hold for it.

If $d(x, x')$ denotes the Euclidean distance between the defined elements x and x' of $dom_{TimeDim}$ —i.e. $d(x, x') = |x - x'|$— then the membership functions of the fuzzy sets described by the generalised operators $=_{\tilde{V}}$, $\leq_{\tilde{V}}$ and $\geq_{\tilde{V}}$ are defined as follows:

$\forall\, x, t \in dom_{TimeDim} \setminus \{\perp_{TimeDim}\}$:

- $\mu_{x=_{\tilde{V}}t}((x)) = \mu_{\tilde{V}}(d')$, with

$$d' = \min\{d(x, x')|x' \in dom_{TimeDim} \wedge\ x' = t\}$$

- $\mu_{x\leq_{\tilde{V}}t}((x)) = \mu_{\tilde{V}}(d')$, with

$$d' = \min\{d(x, x')|x' \in dom_{TimeDim} \wedge\ x' \leq t\}$$

- $\mu_{x\geq_{\tilde{V}}t}((x)) = \mu_{\tilde{V}}(d')$, with

$$d' = \min\{d(x, x')|x' \in dom_{TimeDim} \wedge\ x' \geq t\}$$

Figure 3 illustrates the membership functions that result from the application of the generalised comparison operators $\leq_{\tilde{V}}$, $\geq_{\tilde{V}}$ and $=_{\tilde{V}}$ to a given fuzzy set $\tilde{V}$.

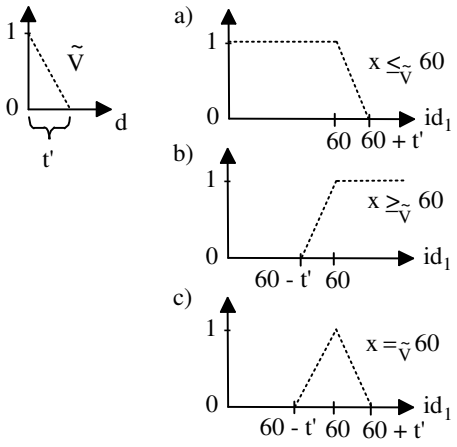

Figure 3: Application of the generalised comparison operators

Linear arithmetic constraints have been generalised by replacing all regular comparison operators by (adequate) generalised comparison operators and by replacing the regular logical operators $\wedge$, $\vee$ and $\neg$ by their fuzzy counterparts $\tilde{\wedge}$, $\tilde{\vee}$ and $\tilde{\neg}$, which semantics have been defined as follows:

- the impact of the $\tilde{\wedge}$ operator is reflected by applying Zadeh's (standard) intersection operator [17] onto the fuzzy sets described by the arguments of the operator, i.e. with arguments $\tilde{U}$ and $\tilde{V}$, the membership degree of (x), $x \in dom_{TimeDim}$ in the resulting fuzzy set equals

$$\min(\mu_{\tilde{U}}(x), \mu_{\tilde{V}}(x))$$

- the impact of the $\tilde{\vee}$ operator is reflected by applying Zadeh's (standard) union operator [17], i.e. with arguments $\tilde{U}$ and $\tilde{V}$, the membership degree of (x), $x \in dom_{TimeDim}$ in the resulting fuzzy set equals

$$\max(\mu_{\tilde{U}}(x), \mu_{\tilde{V}}(x))$$

- the impact of the $\tilde{\neg}$ operator is reflected by applying Zadeh's (standard) complement operator [17], i.e. with argument $\tilde{U}$, the membership degree of (x), $x \in dom_{TimeDim}$ in the resulting fuzzy set equals

$$1 - \mu_{\tilde{U}}(x)$$

For example, with appropriate fuzzy sets $\tilde{U}$, $\tilde{V}$ and $\tilde{W}$, the fuzzy temporal information "*around* time point 60 and from time point *about* 100 until time point *about* 120" can be described as:

$$(x =_{\tilde{U}} 60) \,\tilde{\vee}\, ((x \geq_{\tilde{V}} 100) \,\tilde{\wedge}\, (x \leq_{\tilde{W}} 120))$$

3.2 Modelling of spatial information

The data type *SpaceTime* can be adapted to model spatial information. A distinction is made between one-dimensional, two-dimensional and n-dimensional data.

In order to model spatial information a new data type *SpaceDim* is defined. The definition of this type is similar to the definition of the type *TimeDim*, introduced in the previous section: the domain of *SpaceDim* is defined by

$$dom_{SpaceDim} = \mathbb{R} \cup \{\perp_{SpaceDim}\}$$

where $\perp_{SpaceDim}$ represents an "undefined" domain value; furthermore, the same operators $=$, $\neq$, $<$, $>$, $\leq$, $\geq$, $+$, $-$, $*$, / and $\perp$ have been defined.

3.2.1 One-dimensional spatial data

The type *SpaceTime* is also suited for the modelling of spatial data. One-dimensional spatial data can be handled by considering one (spatial) component. The specification of *SpaceTime* then becomes:

$$SpaceTime\ id(id_1 : SpaceDim)$$

where *id* remains the identifier of the type and id_1 remains the identifier of the component. In the one-dimensional case, the modelling of spatial information is then completely analogous to the temporal case discussed in the previous subsection.

3.2.2 Two-dimensional spatial data

Two-dimensional spatial data can be modelled by considering two (spatial) components for the type *SpaceTime*, i.e. by considering the specification:

$$SpaceTime\ id(id_1 : SpaceDim, id_2 : SpaceDim)$$

In this case, the domain dom_{id} is defined by

$$dom_{id} = \tilde{\wp}(\{(x, y) | x, y \in dom_{SpaceDim}\}) \cup \{\perp_{SpaceTime}\}$$

With this definition, each regular value of dom_{id} is a fuzzy set, which is defined over the continuum of points in the plane defined by the two spatial axes with identifiers id_1 and id_2.

The generalisation of the comparison operators $=$, $\leq$ and $\geq$ is obtained analogously as in the one-dimensional case. A normalised fuzzy set $\tilde{V}$, which is defined over the universe of valid distances and for which the boundary condition $\mu_{\tilde{V}}(0) = 1$ holds, is associated with each operator.

If $d((x, y), (x', y'))$ denotes the Euclidean distance between the defined elements (x, y) and (x', y') of $dom_{SpaceDim} \times dom_{SpaceDim}$ —i.e. $d((x, y), (x', y')) = \sqrt{(x - x')^2 + (y - y')^2}$— then the membership functions of the fuzzy sets described by the generalised operators $=_{\tilde{V}}$, $\leq_{\tilde{V}}$ and $\geq_{\tilde{V}}$ are defined as follows:

$$\forall\ (x, y) \in (dom_{SpaceDim} \setminus \{\perp_{SpaceDim}\})^2, \quad \forall\ m, l \in \mathbb{R}:$$

- $\mu_{x+my=_{\tilde{V}}l}((x, y)) = \mu_{\tilde{V}}(d')$, with

$$d' = \min\{d((x, y), (x', y')) | (x', y') \in (dom_{SpaceDim})^2 \ \wedge\ x' + my' = l\}$$

- $\mu_{x+my\leq_{\tilde{V}}l}((x, y)) = \mu_{\tilde{V}}(d')$, with

$$d' = \min\{d((x, y), (x', y')) | (x', y') \in (dom_{SpaceDim})^2 \ \wedge\ x' + my' \leq l\}$$

- $\mu_{x+my\geq_{\tilde{V}}l}((x, y)) = \mu_{\tilde{V}}(d')$, with

$$d' = \min\{d((x, y), (x', y')) | (x', y') \in (dom_{SpaceDim})^2 \ \wedge\ x' + my' \geq l\}$$

For example, with the fuzzy set $\tilde{V}$ of Figure 3 and the fuzzy set $\tilde{W} = \{(0, 1)\}$, "the *environment* of Annecy" can be modelled by

$$(x =_{\tilde{V}} 11) \,\tilde{\vee}\, (y =_{\tilde{V}} 6.1)$$

and "the *neighbourhood* of the Lake of Geneva *in* Haute-Savoie" can be modelled by

$$(y \geq_{\tilde{V}} 6.7) \,\tilde{\wedge}\, (y - x \geq_{\tilde{V}} -3.7) \,\tilde{\wedge}\, (y \leq_{\tilde{W}} 6.7) \,\tilde{\wedge}\, (y - x \geq_{\tilde{W}} -3.7) \,\tilde{\wedge}\, (x \leq_{\tilde{W}} 11.9) \,\tilde{\wedge}\, (x \geq_{\tilde{V}} 11)$$

Both examples are illustrated in Figure 4 (drawn to scale).

Figure 4: Illustration of the modelling of two-dimensional spatial data

3.2.3 n-dimensional spatial data

In order to model n-dimensional spatial data, the type *SpaceTime* can be constructed with n spatial components. In this case, the domain dom_{id} is defined by

$$dom_{id} = \tilde{\wp}(\{(x_1, x_2, \ldots, x_n) | x_1, x_2, \ldots, x_n \in dom_{SpaceDim}\}) \cup \{\perp_{SpaceTime}\}$$

The comparison operators $=$, $\leq$ and $\geq$ can be generalised straightforwardly and analogously to previous cases by considering the Euclidean distance

$$d((x_1, x_2, \ldots, x_n), (x'_1, x'_2, \ldots, x'_n)) = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \cdots + (x_n - x'_n)^2}$$

and an associated, normalised fuzzy set $\tilde{V}$, which is defined over the universe of valid distances and for which the boundary condition $\mu_{\tilde{V}}(0) = 1$ holds.

3.3 Literal type *SpaceTime*

In general the literal type *SpaceTime* can have both spatial and temporal components. This allows to model spatio-temporal information in its most general form. The type specification then becomes:

$$SpaceTime\ id(id_1 : t_1, id_2 : t_2, \ldots, id_n : t_n)$$

where id remains the identifier of the type, id_i, $i = 1, 2, \ldots, n$ remain the identifiers of the axes represented by the components and $t_i \in \{TimeDim, SpaceDim\}$, $i = 1, 2, \ldots, n$ denote the nature of the axes.

The domain of *SpaceTime* is defined by

$$dom_{id} = \tilde{\wp}(\{(x_1, x_2, \ldots, x_n) | x_i \in dom_{t_i}, i = 1, 2, \ldots, n\}) \cup \{\perp_{SpaceTime}\}$$

where $\perp_{SpaceTime}$ represents an "undefined" domain value.

Because by definition all the elements of the domain are fuzzy sets, operators have been provided for the handling of fuzzy sets. Among the considered operators are: $\cup$, $\cap$, *co*, *normalise*, *support*, *core*, $\alpha-cut$ and $\bar{\alpha}-cut$. Each operator preserves its usual semantics. Additionally, a null-ary operator $\perp$, which always results in an "undefined" domain value, is added.

4 Conclusion

A new approach for the handling of spatio-temporal information is presented. The rationale behind this approach is the assumption that linear arithmetic constraints are particularly appropriate for representing such information.

The approach is presented as an extension of a constraint-based fuzzy object-oriented database model, but its application is definitely not restricted to database models. Central to the approach is the introduction of a new generic type *SpaceTime*, which is suited to handle fuzzy multi-dimensional temporal and/or spatial information. The description of domain values of *SpaceTime* by means of so-called generalised linear arithmetic constraints, which have been obtained by generalising the definition of the comparison operators $=$, $\leq$ and $\geq$, is typical. Future work includes the definition of appropriate data definition and data manipulation operators.

References

[1] L.A. Zadeh (1996). Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems, vol. 2, pp. 103–111.

[2] L.A. Zadeh (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems, vol. 90 (2), pp. 111–127.

[3] G. De Tré, R. De Caluwe and B. Van der Cruyssen (2000). A Generalised Object-Oriented Database Model. In: Recent Issues on Fuzzy Databases. G. Bordogna and G. Pasi (eds.). Studies in Fuzziness and Soft Computing, Physica-Verlag, Heidelberg, Germany, pp. 155–182.

[4] G. De Tré and R. De Caluwe (2000). The Application of Generalized Constraints to Object-Oriented Database Models. Mathware and Soft Computing, vol. VII (2–3), pp. 245–255.

[5] G.M. Kuper, L. Libkin and J. Paredaens (eds.) (2000). Constraint Databases. Springer-Verlag, Berlin, Germany.

[6] P.C. Kanellakis, G.M. Kuper and P.Z. Revesz (1995). Constraint Query Languages. Journal of Computer and System Sciences, vol. 51, pp. 26–52.

[7] H. Prade (2001). From flexible queries to case-based evaluation. In: Proc. of ECSQARU-2001 workshop on Management of uncertainty and imprecision in multimedia information systems, Toulouse, France.

[8] R. De Caluwe, G. De Tré, B. Van der Cruyssen, F. Devos and P. Maesfranckx (2000). Time management in Fuzzy and Uncertain Object-Oriented Databases. In: Knowledge Management in Fuzzy Databases. O. Pons, M.A. Vila and J. Kacprzyk (eds.). Studies in Fuzziness and Soft Computing, Physica-Verlag, Heidelberg, Germany, pp. 67–88.

[9] A. Morris (2001). Why Spatial Databases Need Fuzziness. In: Proc. of joint 9th IFSA and 20th NAFIPS International Conference, Vancouver, Canada, pp. 2446–2451.

[10] R. George, F.E. Petry, B.P. Buckles and S. Radharkrishnan (1996). Fuzzy Database Systems – Challenges and Opportunities of a New Era. International Journal of Intelligent Systems, Vol. 11, John Wiley and Sons.

[11] E.L. Usery (1996). A Conceptual Framework an Fuzzy Set Implementation for Geographic Features. In: Geographic Objects with Indeterminate Boundaries. P. Burrough and A. Frank (eds.).GISDATA Series, vol. 2, Taylor and Francis, London, UK, pp. 71–86.

[12] A. Morris, F.E. Petry and M. Cobb (1998). Incorporating Spatial Data into the Fuzzy Object Oriented Data Model. In: Proc. of IPMU98, Paris, France, pp. 604–611.

[13] R.G.G. Cattell et al. (2000). The Object Data Standard: ODMG 3.0. Morgan Kaufmann Publishers Inc., San Francisco, USA.

[14] G. De Tré (2001). An algebra for querying a constraint defined fuzzy and uncertain object-oriented database model. In: Proc. of joint 9th IFSA and 20th NAFIPS International Conference, Vancouver, Canada, pp. 2138–2143.

[15] J. Early (1971). Toward an understanding of data structures. Communications of the ACM, vol. 14 (10), pp. 617–627.

[16] G. De Tré and R. De Caluwe (2001). Application of Extended Possibilistic Truth Values in Multimedia Database Systems. In: Proc. of ECSQARU-2001 workshop on Management of uncertainty and imprecision in multimedia information systems, Toulouse, France.

[17] L.A. Zadeh (1965). Fuzzy sets. Information and Control, vol. 8 (3), pp. 338–353.