

A UML Profile for Fuzzy Multidimensional Data Models

Iván Rodríguez¹, José Luis Martí²

¹Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso,
Av. Brasil 2147 – Valparaíso, Chile. Email: ivanrodriguezcontreras@gmail.com

²Departamento de Informática, Universidad Técnica Federico Santa María,
Av. Vicuña Mackenna 3939 – San Joaquín Vitacura, Chile. Email: jmarti@inf.utfsm.cl

Abstract. Over the last several years, multidimensional data modeling has had several proposals for its formalisation; on the other hand, the incorporation of fuzzy logic in databases has increased the need to represent uncertainty. However, to our knowledge, so far projects in both areas have not been developed.

This paper suggests joining those two needs to create a solution; proposing a UML profile oriented to design multidimensional data models with the presence of fuzzy elements.

Keywords: Multidimensional Database, Fuzzy Logic, UML Profile.

1 Introduction

A multidimensional database (MDB) manipulates its data as a (hyper)cube, which is composed of dimensions and facts. A dimension represents a variable of interest to the analysis, while the facts represent the subject (interesting patron, event) that must be analyzed to understand its behavior. For example, for a sale (the fact), it will be important to analyze place, client, time, and products (the dimensions) involved in it, and also to try to understand its relationships. The facts are composed of measures that describe the properties the user wants to optimize. Measures are normally shown quantitatively (additive). Generally, a measure has two parts: a numerical property of the fact (i.e.: price of sale, return) and a formula associated to a simple aggregation function (i.e.: sum, count). For a formal definition, see [1].

For the generation of any data models, there are several graphical notations. UML has become the standard language for this type of task [2]. The fact that UML is a general-purpose language provides great flexibility and expressivity when modeling systems. Nevertheless, there are numerous situations in which it is better to count on more specific language to model and to represent the concepts of certain particular domains. This happens, for example, when the syntax or the UML semantics do not allow expression of the specific domain concepts, or when the author wants to restrict or specialize the UML constructors themselves, which are rather generic.

OMG defines two possibilities for achieving a specific domain language: develop a new language (alternative to the UML); or extend the UML itself specializing some of

its concepts and limiting others, but respecting the original semantics of the basic elements (class, attributes, relationships, operations, transitions and so on). This second form is called UML profile; a profile is defined through three mechanisms: *stereotypes*, *constraints*, and *tagged values*. Several UML profiles have been proposed (i.e., real-time systems [3], business modeling [4]); to our knowledge, the only profile that exists to construct conceptual multidimensional data models [5] is based on (crisp) traditional data types.

In all the known cases, the type of data considered is precise. However, today Fuzzy Logic has become an important technique for including imprecise and uncertain concepts, in a way similar to how the human brain works: evaluating many options and then weighing them to make a final decision. Boolean Logic assigns a value *true* (1) when something matches a condition (1), or *false* (0) in any other case. On the other hand, Fuzzy Logic permits relative grades of matching the condition in the inclusive interval [0,1]. Thus, fuzzy sets are a convenient way to represent concepts without precise limits (e.g., high/low temperature).

The general objective of this work is to provide a UML profile that, including fuzzy concepts, can be utilized to generate multidimensional data models.

2 State of the Art

This paper uses two fundamental concepts: the UML profiles to multidimensional data models and fuzzy data models used to design databases. What follows is a short description of both of these.

2.1 UML Profile for a Multidimensional Data Conceptual Model

To our knowledge, only one UML profile has been proposed so far for multidimensional conceptual modeling [5] (for simplicity, this profile is called MDC). In general terms, this profile is defined with the following schema:


- Description: short explanation in natural language.
- Extensions (profiles) or prerequisites: establishes if the proposal needs other extensions for its complete definition.
- Stereotypes: the concrete definition of the elements of the profile.
- Rules: the semantics of the classes is given by a set of invariants defined through OCL.
- Comments: any additional comment, note or example serving to best explain the ideas, usually written in natural language.

In the MDC, the information is structured in facts and dimensions. It is allowed the use of “many-to-many” relationships, but specific attributes must be set to allow this association. Attributes created in this way are called *DegeneratedFacts*. There are some cases in which a dimension is not considered explicitly, given that it is assumed that most of its properties are represented by means of other elements (made or

dimensions); nevertheless, some attributes are still needed to identify instances of facts solely. These dimensions are called *degenerated dimensions*.

In this profile, the stereotypes and values tagged are specified by means of a structured scheme; part of the definition of the *Fact* element is given in Table 1 as an example. It is important to mention that the correct use of this profile is assured by the definition of 51 restrictions specified in natural language and expressions OCL [5].

Table 1. Example of definition of the *Fact* element in the MDC Profile (obtained from [5]).

Name	Fact
Base Class	Class
Description	Classes of this stereotype represent facts in a multidimensional model
Icon	
Constraints	All attributes of a fact must be DegenerateDimension or factAttribute: $\text{self.feature} \rightarrow \text{select}(fe \mid fe.\text{oclIsKindOf}(\text{Attribute})) \rightarrow \text{forAll}(f \mid f.\text{oclIsTypeOf}(\text{DegenerateDimension}) \text{ or } f.\text{oclIsTypeOf}(\text{FactAttribute}))$...
Labels	None

The authors also present a metamodel for their proposal, which is composed of three levels. Level 1 corresponds to the definition of the model; a package represents a star schema of a multidimensional conceptual model. A dependency between two packages of this level indicates that the star schemas share at least one dimension. Level 2 is the definition of the star schemas; a package represents a fact or a dimension of the schema; a dependency between two packages of a dimension indicates that the packages share at least one level in a hierarchy of dimensions (see the graphical representation in figure 1a). Finally, level 3 corresponds to the definition of facts and dimensions; a package is divided into a set of classes that represent the levels of hierarchy in a dimension package, or the complete star schemas in the case of a fact package (see figure 1b).

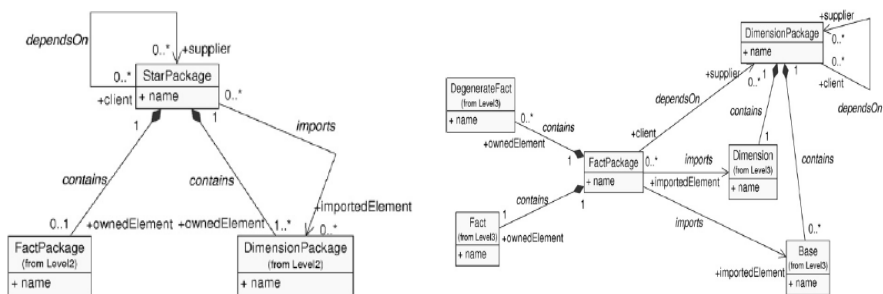


Fig. 1: Level 1 (a) and level 2 (b) of the MDC metamodel (obtained from [5]).

2.2 Fuzziness in the Data Modeling

An extension of the Entity-Relationship Model to incorporate fuzziness was proposed in [6], where fuzzy elements as entities, attributes and associations are represented graphically. This extension defines three levels of fuzziness:

1. Entities, attributes and associations could be fuzzy; in other words, could have membership grades to the E-R model.
2. Fuzzy occurrences of entities and associations.
3. Fuzzy values of the attributes, expressed through a distribution of possibility.

In the case of UML, the fuzzy data modeling corresponds to an extension of the class diagram [6,7]. A class could be extended considering:

1. Some objects are fuzzy and share the same properties; therefore, the class defining them is fuzzy. Then, the objects belong to that class with a grade of membership in the interval [0,1].
2. If the domain of an attribute could be fuzzy; then the class including it is fuzzy, too.
3. The subclasses generated from a fuzzy class by specialization, and the superclass produced by one or more fuzzy subclasses through generalization are fuzzy, too.

Figure 2 shows a fuzzy class, which is distinct of the classic case for its discontinuous line. Its attribute *Age* has a fuzzy domain, and as consequence the class including it is fuzzy. Taking in account that students with a grade of possibility to achieve a job are really postgraduate students, the same class is fuzzy, too.

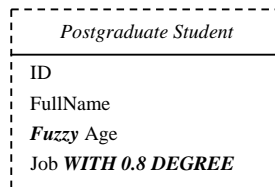


Fig. 2. Examples of a fuzzy UML class (obtained from [6]).

A fuzzy association could be present in two levels of fuzziness; the first one means a fuzzy association can exist between two classes, with a grade of possibility; in the second one, an instance belongs with a membership grade to an association class. It is important to mention that with fuzzy sets, it is possible to represent “imprecise” concepts in a convenient form to core of a database, considering four types de values [6]:

1. Precise values (*crisp*), traditionally used in a database, but with the possibility to be included in fuzzy queries.
2. Fuzzy data represented as distributions of possibility. For example: the quality expressed as (0.4/regular, 0.7/good), where regular “<” good.
3. Attributes over “discreet, no ordered domains with analogy”, with a relation of similarity. For example: hair color expressed as (1/blonde, 0.4/brown, 0.6/red).
4. Attributes defined as the last type, but without the relation of similarity.

3 Proposal of a UML Profile for a Fuzzy Multidimensional Data Model

The explanation of the profile will be made in an inverse order with respect the structure developed to the MDC that is *Attributes*, *Classes and Packages*. This way it begins with the smaller elements and finishes with the greater ones, which are composed of the first.

In addition to the previously mentioned, there is a fourth element, an *AssociationClass*, which could be of the *DegeneratedFact* type. This element is manipulated as a relationship between the fact table of the star schema and some dimension. Note that this is because there is an association justifying it, but it is not the association itself.

3.1 Attributes

There are five types of attributes: *Degenerate Dimension*, *FactAttribute*, *OID*, *Descriptor* and *DimensionAttribute*, which can only be present in the classes *Fact*, *Base* and *DegenerateFact*. The possibility to include them in the profile is indicated below.

- **OID (OID)**: corresponds to the identifier of the *Base* classes. This element is necessary to a process of automatic export in OLAP tools, because these store it in the metadata. For this reason, it was not considered as a fuzzy element.
- **Descriptor (D)**: attribute derivable from another attributes, and it can be quantitative or qualitative. Without taking this into account, it can be any fuzzy type.
- **DimensionAttribute (DA)**: gives descriptive information about an instance of *Dimension*. It can be optional and doesn't need to be specified in every element (if it is necessary, it can contain the "null" value). For its role in a dimension, it could be any fuzzy type.
- **DegenerateDimension (DD)**: can be considered as a dimension included in the composition of the fact table. For example, the attribute "Address" as geographical feature in a fact table, could be disaggregated into sector, street, and so on. Its presence can not be fuzzy, but its value could. For the same example, the DD *Address* could be type 3 to establish relationships of proximity between the cities and permit fuzzy queries like "close to..." or "far away from...".
- **FactAttribute (FA)**: attribute of the *Fact* o *DegenerateFact* classes. Consider the case of an attribute *city*; taken into account as FA not as DA, could be classified as type 1, since it is a precise data but can be used in fuzzy queries. However, if the case is such that the borders of the cities are not well defined but it is required to register such data for a house that is "within the city limits", it could be necessary to specify with what degree it belongs to that city and what degree to another. If this second situation were true, the attribute *city* would become type 3, which could be represented through distributions of possibility.

As it has been indicated already, the absence of some of the attributes that support fuzziness (*DD*, *FA*, *DA* and *D*) would repel in the data model (cube), with a degree between [0,1] in the requirements satisfied by it. For example, suppose a cube that includes the *Customer* dimension, in which the author wants incorporate fuzziness in the property of the *FullName* attribute of the *Base* class. Consider a case in which a given attribute is not present in that *Base*; transitively then, it will not be present in the *Dimension*, either. Then, to respond to queries about sales, purchases or credit notes (any be the domain) of a person, this requirement could not be solved given the absence of essential data for the problem.

The above is explained in terms of having fuzzy attributes, not fuzzy values -- since the classes cannot own values but rather attributes -- nor in terms of its presence in the model, because it has been decided to consider that aspect as fixed. In addition, the absence of the attribute can be replaced by the value "null" in this attribute.

3.2 Classes

The elements of the Class type belonging to the profile MDC are *Fact*, *Dimension* and *Base*. In these cases it was considered that evaluation about the presence of fuzziness must be focused on the nature of the attributes that compose them. Like in the previous section, the presentation will be made beginning with the parts -- *Base*, which through inheritance form the *Dimension*, and these with the *Fact* composition structure.

- **Base:** it is a level of a hierarchy of a classification, by means of which the *Dimension* classes are defined. As an example, a *Customer* Dimension can have several Bases; one of these called *Region* Base, which owns the attribute *Location*, which can be considered of type 3 when incorporating the degree of proximity with other regions. So, that Base can be considered fuzzy while it has those fuzzy attributes.
It is possible to consider to any *Base* as fuzzy based on the roll that their attributes play in the queries. For example, it can be more advisable to define type 3 to the bases relative to the *Dealership* dimension (concessionary), to analyze the sales grouped by region, that of the dealership not of the clients, where these last ones are associated to the dealership where they bought the vehicle.
- **Dimension:** by itself does not own attributes, but an instance of the *Base* class does. In addition, only one *Base* is associated to the instance of *Dimension*, therefore it will be considered as a fuzzy *Dimension* to those associated with an instance of the class *Fuzzy Base*.
- **Fact:** composed by *FactAttributes*, owned by the *Fact* class, and by *DegenerateDimension* attributes. Both support fuzzy data so that this class could be considered fuzzy in the way that some of their attributes are.

3.3 Packages

- **DimensionPackage:** can contain instances of the *Dimension* and *Base* classes, where the number of dimensions is 1; this is considered fuzzy if the dimension containing it is fuzzy, meaning *Base* classes of the package can be fuzzy but not necessarily will the last one be. This choice is to avoid all of the instances of the data model being fuzzy.
- **FactPackage:** can only store instances of the *Fact*, *Dimension* or *Base* classes; this package is fuzzy if the only instance of the *Fact* class is also fuzzy.
- **StarPackage:** can only contain instances of the *FactPackages* or *DimensionPackages* classes. It will be considered fuzzy if the unique *FactPackage* is fuzzy as well.

3.4 Association Classes

The *DegeneratedFact* is the one element belonging to this category. The attributes can be of *DegenerateDimension* or *FactAttribute* classes. Then, it is considered if any of its attributes can contain a fuzzy value of some type.

3.5 Summary of the Profile

Considering the above, the identified elements must be associated with the definition of the metamodel of level 3, defined previously by the base profile. Figure 3 shows the relationships between the classic elements of a multidimensional data model, the elements incorporated by MDC and the *fuzzy* elements added by the present work.

To avoid ambiguous use of the profile's elements, OCL Constraints has been incorporated; as example, all of the attributes of a *FuzzyBase* must be *OID*, *Descriptor*, *FuzzyDescriptor*, *DimensionAttribute* or *FuzzyDimensionAttribute*, which OCL expression is:

```
self.feature -> select(fe | fe.oclsKindOf(Attribute)) -> forAll(f | f.oclsTypeOf(OID) or
    f.oclsTypeOf(Descriptor) or f.oclsTypeOf(FuzzyDescriptor) or
    f.oclsTypeOf(DimensionAttribute) or f.oclsTypeOf(FuzzyDimensionAttribute))
```

where *self* references to the instance of the *FuzzyBase* class that is being evaluated.

4 Application of the Proposal in a Practice Case

To demonstrate the use of the profile, it is applied to a problem based in a library, where an analysis of information about the material lent to its different types of users is required. The dimensions of interest are:

1. Material: book, magazine, newspaper.
2. Librarian: staff responsible for giving the service.
3. User: the person who gets a material from the library.
4. Time.

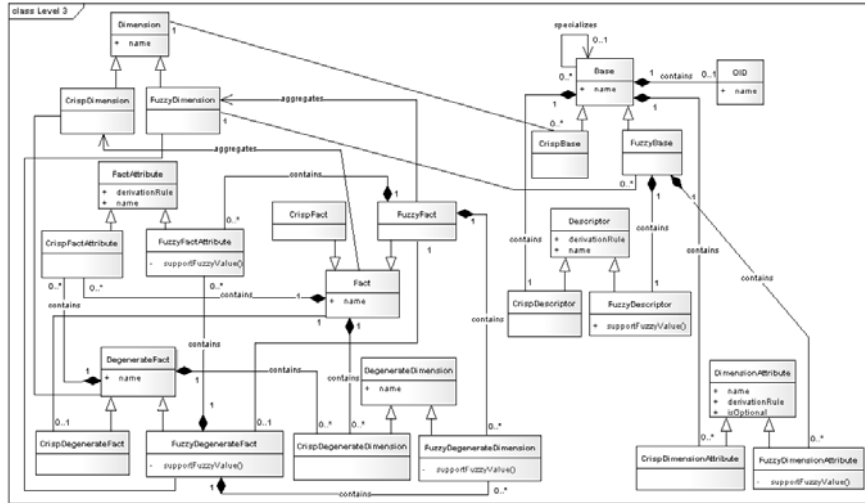


Fig. 3. Level 3 of the metamodel of the proposed profile.

A fact table will register the daily loans, associated with the previous dimensions as figures 4 and 5 show. In the first, the definition of level 2 of the *Material* dimension is given. Note that fuzzy class *Material* is superclass of the other three, passing on the fuzziness. Moreover, the base class *Magazine* is fuzzy too, because its attribute *tendency* could be of *vogue*, *entertainment*, *sport*, and so on, in different grades and without an order between them (type 4).

Figure 4b gives the level 2 of the *Librarian* dimension. The base classes have been considered as fuzzy because the descriptors, which mean the type of position the person has and the function they fulfill can be imprecise or overlap.

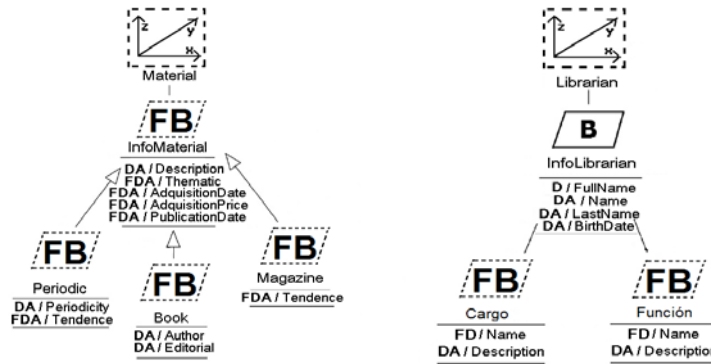


Fig. 4. Level 2 of the metamodel of the proposed profile, by the Material (a) and Librarian (b) dimensions.

In the case of figure 5, the *User* dimension has been modeled as fuzzy, because it has a type 3 attribute, which is the user type (*Student*, *Particular*, *habitualUser*, *SporadicUser*, ..).

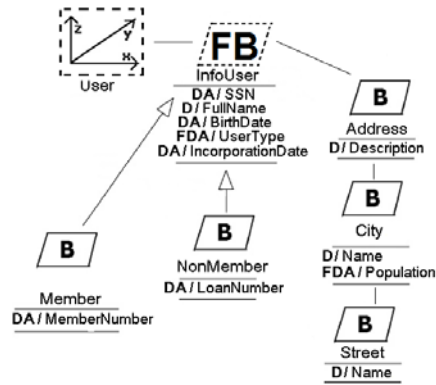


Fig. 5. Level 2 of the metamodel of the proposed profile.

Figure 6 shows the model of level 2 to represent the *Fact* instance with its Dimensions associated. Considering the dimensions *User*, *Material* and *Librarian* are fuzzies, and having attributes that can be used in fuzzy queries, the *Loan* instance of *Fact* is fuzzy too.

The degenerate fact *DateBT* represents the loans start-date and length. This can occur in an association of the type “many-to-many”. However, its type is fuzzy because such a date can be used in fuzzy queries, for example to list the loans out for a long time.

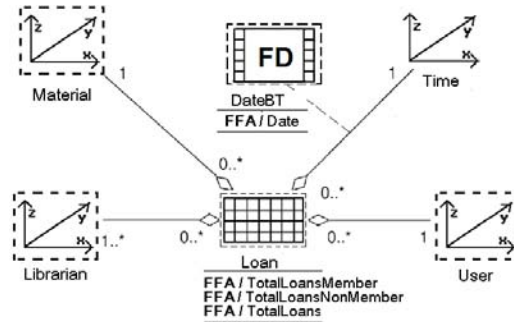


Fig. 6. Example of the level 2 of the *Material* Dimension.

5 Conclusions

The research of the fuzzy logic and its inclusion in the databases, provide the first ideas to formulate a way to get a multidimensional data model with imprecise elements. This task was facilitated by a previous profile, which just contained crisp data in its structure.

Current work is concentrated on adding more precise OCL specification to the elements, to guide towards an automatic process on a CASE tool.

References

1. Molina, C., Rodríguez-Ariza, L., Sánchez D., Vila M. A.: *A New Fuzzy Multidimensional Model*. IEEE Transactions on Fuzzy Systems, vol. 14(6), pp. 897-912 (2006).
2. Object Management Group (OMG). *Unified Modeling Language Specification 1.5*. <http://www.uml.org/>.
3. Ludovic, A., Courtiat, J.-P., Lohr, C., Saqui-Sannes, P. de.: *TURTLE: A Real-Time UML Profile Supported by a Formal Validation Toolkit*. IEEE Transactions on Software Engineering, vol. 30(7), pp. 473-487 (2004).
4. Catalog of UML Profile Specifications. http://www.omg.org/technology/documents/profile_catalog.htm.
5. Lujan-Mora, S., Trujillo, J., Il-Yeong, S.: *A UML Profile for Multidimensional Modeling in Data Warehouses*. ACM Data and Knowledge Engineering, vol. 59(3), pp. 725-769 (2006).
6. Galindo, J., Urrutia, A., M. Piattini, M. : *Fuzzy Databases: Modeling, Design and Implementation*. Idea Group Publishing (2006).
7. Ma. Z.: *Fuzzy Database Modeling with XML*. Springer (2005).