A Flexible Approach to the Multidimensional Model: The Fuzzy Datacube

Miguel Delgado[a], Carlos Molina[a], Daniel Sánchez[a], Lázaro Rodriguez Ariza[b], M. Amparo Vila[a]

[a]Universidad de Granada, Dpto. de Ciencias de la Computación e I.A., E.T.S.I. de Informática
C/ Periodista Daniel Saucedo Aranda s/n
18071 Granada, Spain
{mdelgado, carlosmo, daniel, vila}@decsai.ugr.es
[b]Universidad de Granada, Depto. Economía Financiera y Contabilidad,
Faculty of Economics and Business Sciences
Campus Universitario de la Cartuja, 18071 Granada, Spain
lazaro@ugr.es

Abstract. As a result of the use of OLAP technology in new fields of knowledge and the merge of data from different sources, it has become necessary for models to support this technology. In this paper, we propose a new multidimensional model that can manage imprecision both in dimensions and facts. Consequently, the multidimensional structure is able to model data imprecision resulting from the integration of data from different sources or even information from experts, which it does by means of fuzzy logic.

1 Introduction

Ever since the appearance of the OLAP technology ([5]), there have been various proposals to support its special needs, and in particular, two different approaches have been documented. The first of these extends the relational model to support the structures and operations which are typical of OLAP, and the first proposal of such a type can be found in [9]. Since then, there have been other proposals (e.g. [10]), and most of the present relational systems include extensions to represent datacubes and operate on them. The second approach is to develop new models using a multidimensional view of the data. Many authors have proposed models in this way ([1, 3, 4, 12]).

In the early 70s, the need for flexible models and query languages to manage the ill-defined nature of information in DSS was identified ([8]). Nowadays, the application of the OLAP technology to other knowledge fields (e.g. medical data) and the use of semi-structured sources (e.g. XML) and non-structured sources (e.g. plain text) has made these requirements on the models even more important. The systems now need to manage imprecision in the data, and more flexible structures are needed to represent the analysis domain. New models have appeared to manage incomplete datacubes ([7]), imprecision in the facts ([11]), and the definition of facts using different levels in the dimensions ([13]). In addition, these models continue to use rigid hierarchies and this makes it extremely difficult for certain domains to be modelled. Consequently, this could result in the loss of information when we need to merge data from different sources with incompatibilities in their schemata.

In this paper, we propose a new multidimensional model which is able to handle imprecision in hierarchies and facts by using fuzzy logic. The use of fuzzy hierarchies enables the structures of the dimensions to be defined to the final user more intuitively, thereby allowing a more intuitive use of the system. Furthermore, this allows information to be merged from different sources with incompatibilities in their structures, or even information given by experts to be used in order to improve the multidimensional schema. In the next section, we shall introduce classical multidimensional models as an introduction to presenting our approach. Then, in the third section we shall include an example of the structure proposed to show how to apply the operations on the multidimensional structure. The final section presents the main conclusions and future work

2 Multidimensional Model

In this section, we shall present our proposed multidimensional model. Firstly, we shall introduce what we have called the classical models (these being the first documented models). Secondly, we shall define the multidimensional structure for managing imprecision. We shall then include the basic operations on the multidimensional models (roll-up, drill-down, dice, slice and pivot), and show how these are applied on the fuzzy structure.

2.1 Classical Multidimensional Models

In classical multidimensional models, we can distinguish two different types of data: on one hand, we have the facts being analysed, and on the other, the dimensions are the context for the facts. Hierarchies may be defined in the dimensions. The different levels of the dimensions allow us to access the facts at different levels of granularity. In order to do so, classical aggregation operators are needed (maximum, minimum, average, etc).

The defined hierarchies use many-to-one relations, so one element in a level can only be grouped by a single value of each upper level in the hierarchy. This makes the final structure of a datacube rigid and well defined in the sense that given two values of the same level in a dimension, the set of facts relating to these values have empty intersection.

The normal operations (roll-up, drill-down, dice, slice and pivot) are defined on this structure.

2.2 Multidimensional Structure

Definition 1. A dimension is a tuple $d=(l,\leq_d,l_\perp,l_\top)$ where $l=\{l_i, i=1,...,n\}$ such that each l_i is a set of values and $l_i \cap l_j=\emptyset$ if $i\neq j$, and $\leq_d$ is a partial order relation between the elements of l. $l_\perp$ and $l_\top$ are two elements in l such that $\forall l_i \in l \quad l_\perp \leq_d l_i$ and $l_i \leq_d l_\top$.

Each element l_i is called a level. In order to identify level l of dimension d, we shall use $d.l$. The two special levels $l_\perp$ and $l_\top$ shall be called the *base level* and *top level*, respectively. The partial order relation in a dimension gives the hierarchical relation between the levels.

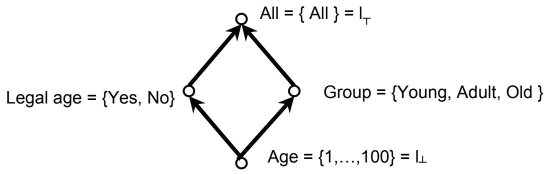

Fig. 1. Example of an age hierarchy

In Figure 1, you can see a definition of an age hierarchy. The definition of the dimension as we have presented it would be *Age = ({Age, Group, legal age, All},*$\leq_{Age}$*, Age, All),* and the relation *Age* $\leq_{Age}$ *Age, Group* $\leq_{Age}$ *Group, Legal age* $\leq_{Age}$ *Legal age, All* $\leq_{Age}$ *All, Age* $\leq_{Age}$ *Group, Age* $\leq_{Age}$ *Legal age, Age* $\leq_{Age}$ *All, Group* $\leq_{Age}$ *All* and *Legal age* $\leq_{Age}$ *All.*

Definition 2. For each dimension d, the domain is $dom(d)=\bigcup l_i$.

In the above example, the domain of the dimension *Age* is *dom(Age)={1,...,100, Young, Adult, Old, Yes, No, All}*.

Definition 3. For each l_i, the set

$$H_{l_i} = \{l_j / l_j \neq l_i \wedge l_j \leq_d l_i \wedge \neg\exists l_k \quad l_j \leq_d l_k \leq_d l_i\} , \tag{1}$$

and we call this the *set of children of level* l_i.

Using the same example of the dimension on the ages, the set of children of the level *All* is H_{All}*={Group, Legal age}*. In all the dimensions we define, for the *base level,* this set will be always the empty set, as you can see from the definition.

Definition 4. For each l_i, the set

$$P_{l_i} = \{l_j / l_i \neq l_j \wedge l_i \leq_d l_j \wedge \neg\exists l_k \quad l_i \leq_d l_k \leq_d l_j\} , \tag{2}$$

and we call this the *set of parents of level* l_i.

On the hierarchy we have defined, the set of parents of level *Age* is P_{Age}*={Legal age, Group}*. In the case of the *top level* of a dimension, this set will always be the empty set.

Definition 5. For each pair of levels l_i and l_j such that $l_j \in H_{l_i}$, we have the relation $\mu_{ij} : l_i \times l_j \rightarrow [0,1]$, and we call this the *kinship relation*.

The degree of inclusion of the elements of a level in the elements of their parent levels can be defined using this relation. If we only use the values 0 and 1 and one element is only included with degree 1 for a single element of its parent levels, this relation represents a crisp hierarchy. Following the example, the relation between the levels *Legal age* and *Age* is of this type. The parent relation in this situation is

$$\mu_{LegalAge,Age}(Yes,x) = \begin{cases} 1 \text{ if x} \in [18,100] \\ 0 \text{ in other case} \end{cases} \quad \mu_{LegalAge,Age}(No,x) = \begin{cases} 1 \text{ if x} \in [1,17] \\ 0 \text{ in other case} \end{cases} . \quad \textbf{(3)}$$

If we relax these conditions and allow values to be used in the interval [0,1] without any other limitation, we have a fuzzy hierarchical relation. This allows several hierarchical relations to be represented more intuitively. An example can be seen in Figure 2 where we present the group of ages according to linguistic labels. Furthermore, this fuzzy relation allows hierarchies to be defined in which there is imprecision in the relationship between elements in different levels. In this situation, the value in the interval shows the degree of confidence in the relation.

Definition 6. For each pair of levels l_i y l_j of the dimension d such that $l_j \le_d l_i \wedge l_j \neq l_i$, the relation $\eta_{ij} : l_i \times l_j \rightarrow [0,1]$ is defined as

$$\eta_{ij}(a,b) = \begin{cases} \mu_{ij}(a,b) \quad if \quad l_j \in H_{l_i} \\ \underset{l_k \in H_{l_i}}{\oplus} \underset{c \in l_k}{\oplus} (\mu_{ik}(a,c) \otimes \eta_{kj}(c,b)) \quad in \quad other \quad case \end{cases} , \quad \textbf{(4)}$$

where ⊕ y ⊗ are a t-conorm and a t-norm, respectively, or operators from the families MOM or MAM defined by Yager ([15]), which include the t-conorms and t-norms, respectively. This relation is called the *extended kinship relation*.

This relation gives us information about the degree of relation between two values in different levels in the same dimension. In order to obtain this value, it considers all the possible paths between the elements in the hierarchy. Each one is calculated by aggregating the *kinship relation* between elements in two consecutive levels using a t-norm. The final value is then the aggregation of the result of each path using a t-conorm. By way of example, we will show how to calculate the value of $\eta_{All, Age}(All,25)$. In this situation, we have two different paths. Let us look at each:

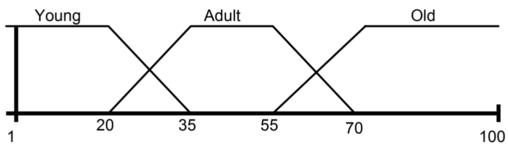

Fig. 2. Kinship relation between levels *Group* and *Age*

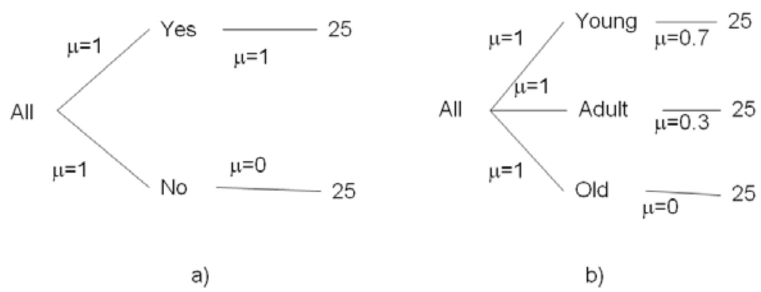

Fig. 3. Example of the calculation of the extended kinship relation. a) path *All – Legal age – Age* b) path *All – Group – Age*

- *All – Legal age – Age*. In Figure 3.a, you can see the two ways to get to 25 from *All* passing the level *legal age*. The result of this path is $(1\otimes 1)\oplus(1\otimes 0)$.
- *All – Group – Age*. This situation is very similar to the previous one. In Figure 3.b, you can see the three different paths going through the level *Group*. The result of this path is $(1\otimes 0.7)\oplus(1\otimes 0.3)\oplus(1\otimes 0)$.

We must now aggregate these two values using a t-conorm in order to obtain the result. If we use the *maximum* as the t-conorm and the *minimum* as the t-norm, the result is $((1\otimes 1)\oplus(1\otimes 0)) \oplus ((1\otimes 0.7)\oplus(1\otimes 0.3)\oplus(1\otimes 0)) = (1\oplus 0)\oplus(0.7\oplus 0.3\oplus 0) = 1\oplus 0.7 = 1$, so the value of $\eta_{All,Age}(All,25)$ is 1, which means that the age *25* is grouped by *All* in the level *All* with grade 1.

Definition 7. We say that any pair (h,α) is a fact when h is an m-tuple on the attributes domain we want to analyze, and $\alpha\in[0,1]$.

The management of uncertainty in the facts is carried out using a degree of certainty with each one. This degree of certainty allows us to use values in analysis that might be interesting to the decisor but which imply imprecision. The value α of each pair controls the influence of the fact in the analysis.

Definition 8. An object of type *history* is the recursive structure

$$H = \begin{cases} \Omega \\ (A, l_b, F, G, H') \end{cases} , \qquad \textbf{(5)}$$

where Ω is the recursivity clause, F is the fact set, l_b is a set of levels $(l_{1b},\ldots,l_{nb})$, A is an application from l_b to F, G is an aggregation operator, and H' is a structure of type *history*.

The role of this structure will be clear after the operations have been defined in the next section.

Definition 9. A datacube is a tuple $C=(D,l_b,F,A,H)$ such that $D=(d_1,\ldots,d_n)$ is a set of dimensions, $l_b=(l_{1b},\ldots,l_{nb})$ is a set of levels such that l_{ib} belongs to d_i, $F = R \cup \emptyset$ where R is the set of facts and $\emptyset$ is a special symbol, H is an object of type *history*, and A is an application defined as $A: l_{1b} \times ... \times l_{nb} \rightarrow F$, giving the relation between the dimensions and the facts defined.

If for $\vec{a} = (a_1,...,a_n)$, $A(\vec{a}) = Ø$, this means that no fact is defined for this combination of values.

Definition 10. We say that a datacube is *basic* if $l_b = (l_{1\perp},...,l_{n\perp})$ and *H= Ω*.

Having defined the structure, we shall now show how to translate a multidimensional schema into our model. An example of a multidimensional model is shown in Figure 4. In this schema, we want to analyze the sales in a company. The broken lines represent the fuzzy relation between the levels, i.e. the relations take values in the entire interval [0,1]. It is possible to see how three dimensions are considered: *Time*, *Product* and *Customer*. This schema translated into our model corresponds to C_{sales}*=({customer, product, time}, {(price, amount)}∪Ø,A,Ω)*. In order to complete the definition, we need the dimension structures: *Customer = ({Age, Legal Age, Group, All}, $\leq_{Customer}$, Age, All), Product = ({Product, Category, Provider, Quality, All}, $\leq_{Product}$, Product, All), Time = ({Date, Month, Holiday, All},$\leq_{Time}$, Date, All)* and the application *A* that gives the relation between the dimensions and the facts: *A: Age x Product x Date → {(price, amount)}∪Ø* .

2.3 Operations

Once we have defined the multidimensional structure, we need the basic operations to work with it. In this section, we shall define the operations to change the level in the hierarchies (roll-up and drill-down) as well as the selection (dice), projection (slice) and pivot. First, two preliminary concepts are needed.

Definition 11. An aggregation operator is a function *G(B)* where $B = \{(h,\alpha)/(h,\alpha) \in F\}$ and the result is a tuple *(h', α')*.

The parameter of an aggregation operator can be seen as a fuzzy bag ([6]) since it concerns a collection of elements (the facts) which can be repeated, with each having a value in the [0,1] interval (the α defined in the tuples).

Definition 12. For each value *a* in a level l_i, we have the set

$$F_a = \begin{cases} \bigcup_{l_j \in H_{l_i}} F_b / b \in l_j \wedge \mu_{ij}(a,b) > 0 & if \quad l_i \neq l_b \\ \{h / h \in H \wedge \exists a_1,...,a_n A(a_1,..,a,...,a_n) = h\} & if \quad l_i = l_b \end{cases} . \quad \textbf{(6)}$$

This set includes all the facts that are in any way related to value *a*, and this is all we need to introduce the operations and to apply them on the fuzzy multidimensional structure proposed.

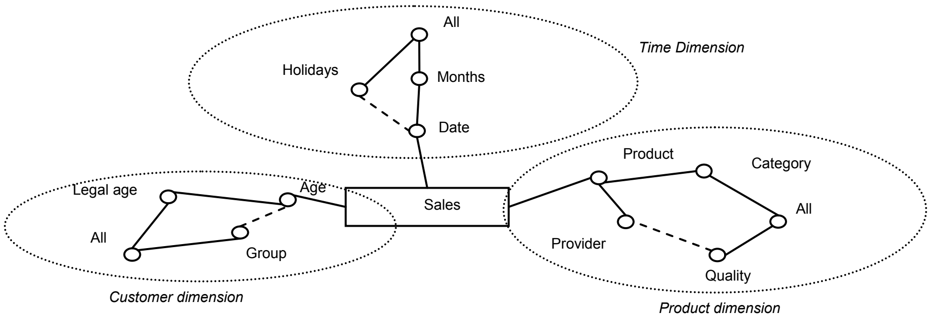

Fig. 4. Example of multidimensional schema

Definition 13. The result of applying *roll-up* on dimension d_i, level l_r ($l_r \neq l\bot$), using the aggregation operator G on a datacube $C=(D,l_b,F,A,H)$ is another datacube $C'=(D,l'_b,F',A',H')$ where $l'_b=(l_{1b},\ldots,l_r,\ldots,l_{nb})$, $A'(a_1,\ldots,a,\ldots,a_n)=G(\{(b,\alpha\otimes\eta_{rb}(a,c))/(b,\alpha)\in F_a \wedge A(a_1,\ldots,c,\ldots,a_n)=(b,\alpha)\})$, F' is the range of A', and $H'=(A, l_b, F, G, H)$.

Definition 14. The result of applying *drill-down* on a datacube $C=(D,l_b,F,A,H)$ having $H=(A',l'_b,F'.H')$ is another datacube $C'=(D,l'_b,F',A',H')$.

After the definition of the *drill-down* operation, we can see the role of the structure *history* inside our proposal. This recursive structure enables us to return at any time to the previous state before the *roll-up* was applied. Consequently, loss of information is prevented as you progress up the hierarchy.

Definition 15. The result of applying *dice* with the condition β on level l_r of dimension d_i in a datacube $C=(D,l_b,F,A,H)$ is another datacube $C'=(D',l'_b,F',A',\Omega)$ where $D'=\{d_1,\ldots,d'_i,\ldots,d_n\}$ where $d_i'=(l_i', \leq_{di}, l_b, l_T)$ having $l'=\{l_j/l_b \leq_{di} l_j\}$ and

$$d'_i.l'_j=\begin{cases}\{v/v\in l_j \wedge \beta(v)\} & if \quad l'_j=l_r\\ \{v/v\in d_i.l_j \wedge \exists x\in l_r \beta(x)\wedge\eta_{rj}(x,v)>0\} & if \quad l'_j\leq_d l_r\\ \{v/v\in d_i.l_j \wedge \exists x\in l_r \beta(x)\wedge\eta_{jr}(v,x)>0\} & if \quad l_r\leq_d l'_j\end{cases},$$

$A'(a_1,\ldots,a_i,\ldots,a_n)=(h,\alpha\otimes\mu_\beta)/a_1\in d'_1.l'_b \wedge \ldots a_n\in d'_n.l'_b \wedge A(a_1,\ldots,a_n)=(h,\alpha)$ where $\mu_\beta=\bigoplus_{c\in d'_i.l'_r}\eta_{rb}(c,a_i)$, and F' is the range of A'.

Definition 16. The result of applying *slice* on dimension d_i using the aggregation operator G in a datacube $C=(D,l_b,F,A,H)$ is another datacube $C'=(D',l_b',F',A',\Omega)$ where $D'=(d_1,\ldots,d_{i-1},d_{i+1},\ldots,d_n)$, $l_b'=(l_{ib},\ldots,l_{i-1b},l_{i+1b},\ldots,l_{nb})$, $A'(a_1,\ldots,a_{i-1},a_{i+1},\ldots,a_n)=G(\{(h,\alpha)/\exists x A(a_1,\ldots,a_{i-1},x,a_{i+1},\ldots,a_n)=(h,\alpha)\})$, and F' is the range of A'.

Definition 17. The result of applying *pivot* on dimensions d_i and d_j in a datacube $C=(D,l_b,F,A,H)$ is another datacube $C'=(D',l_b',F,A',\Omega)$ where $D'=(d_1,\ldots,d_{i-1}$

$,d_j,d_{i+1},\ldots,d_{j-1},d_i,d_{j+1},\ldots,d_n)$, $l_b'=(l_{1b},\ldots,l_{i-1b},l_{jb},l_{i+1b},\ldots,l_{j-1b},l_{ib},l_{j+1b},\ldots,l_{nb})$, and $A'(a_1,\ldots,a_{i-1},a_i,a_{i+1},\ldots,a_{j-1},a_j,a_{j+1},\ldots,a_n) = A(a_1,\ldots,a_{i-1},a_j,a_{i+1},\ldots,a_{j-1},a_i,a_{j+1},\ldots,a_n)$.

Although we now have the operations to work with the structure proposed, this structure can represent objects that are not suitable for the operations defined above. We must therefore say when a datacube is valid to work with it.

Definition 18. A datacube is *valid* if it is *basic* or has been obtained by applying a finite number of operations on a *basic datacube*.

2.4 User View

We have presented a structure that manages imprecision by means of fuzzy logic. We need to use aggregation operators on fuzzy bags in order to apply some of the operations presented. Most of the methods previously documented give a fuzzy set as a result. As this situation can make the result difficult to understand and use in a decision process, we propose a two-layer model: one of the layers is the structure presented in the previous section; and the other is defined on this, and its main objective is to hide the complexity of the model and provide the user with a more understandable result. In order to do so, we propose the use of a fuzzy summary operator that gives a more intuitive result but which keeps as much information as possible. Using this type of operator, we shall define the *user view*.

Definition 19. Given a summary operator M, we define the *user view* of a datacube $C=(D,l_b,F,A,H)$ using M as the structure $C_M=(D,l_b,F_M,A_M)$ where $A_M(a_1,\ldots,a_n)=M(A(a_1,\ldots,a_n))$ and F_M is the range of A_M.

We can define as many user views of a datacube as the number of summary operators used. Therefore, each user can have their own *user view* with the most intuitive view of data according to their preferences by using a datacube. As an example of this type of operator, we can use the one proposed in [2]. This operator proposes the use of the fuzzy number that best fits, in the sense of fuzziness, the fuzzy set or fuzzy bag.

3 Example

Once we have defined the fuzzy structure and the operations on it, we shall present an example of a simple multidimensional schema in order to show the application of operations on it. This example will be modelled using the classical multidimensional or crisp model to show the differences between both approaches. We will use the schema in Figure 4.

In the fuzzy case, the dimension *Customer* is the fuzzy hierarchy on ages which we have used previously. The remaining elements in both the fuzzy and the crisp case are shown in Figure 5, with the exception of the partial order relations which are clear in the schema. Here we see the first differences between both approaches when we model the levels *group* and *holiday*. In the crisp case, these concepts are modelled using intervals on the ages and dates, respectively. In our approach, we use linguistic labels. The facts used in the example and their relation with the values in the dimension are shown in Table 1. If the user wants to know *"the average amount of sales at Christmas for the different age groups and the quality of the provider"*, the sequence of operations to apply is:

1. dice on the dimension *time*, in the level *holiday* with the condition *β(x)="x is Christmas"*.
2. roll-up in the dimension *time* and level *holiday*, dimension *product* and level *quality* and dimension *customer* and level *group*, using the aggregation operator *average* on the *amount*.

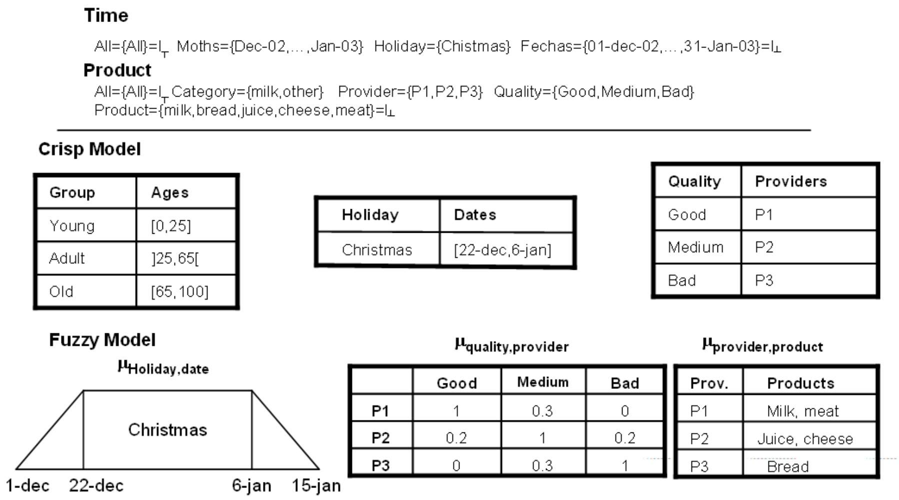

Fig. 5. Dimension structures for the multidimensional schema

In order to apply the roll-up operation, we need the average aggregation operator. Although we can use the classical operator in the crisp case, in the fuzzy model we need an operator that works with fuzzy bags. In the example, we have used the operators proposed by Rundensteiner ([14]) for a fuzzy relational model. The adaptation of these operators to our approach is simple: if R is an aggregation operator defined by Rundensteiner, the operator G_R for our approach is defined as $G_R(h)=(R(h),1)$.

We need another operator to show the results in the fuzzy case. We have used the linguistic summary ([2]) as the summary operator. The results in both approaches are shown in the Tables 2-4. When analyzing the results, we need to bear in mind the differences between both approaches. Therefore, when the user gets the result in the crisp case, for example for the group *young*, the results correspond to the query *"the average amount of sales in the interval [22-dic,6-jan] by the customer with ages in the interval [0,25] and the quality of the provider"*. In the fuzzy case, the user gets a result which is closer to his/her concept of Christmas and youth.

If we want to refine the results in order to obtain *"the maximum average amounts sold by age groups"*, we need to apply *slice* on the dimensions *Products* and *Time*, using the *maximum* aggregation operator. The result is shown in Table 5.

The results obtained in each case are different. This occurs because the values involved in each calculation and their importance are different in both approaches. In the crisp case, all the values inside the intervals have the same weight in the aggregation process. In the fuzzy model, on the other hand, the values at the edges of the concepts do not have the same importance as the values in the kernel in the final result. We can also see the role of the user view in the fuzzy model. The multidimensional structure proposed is based on fuzzy logic and the results shown to the user are fuzzy sets which are difficult to understand. The user view helps to interpret the results, showing the information obtained in a more expressive and understandable way to the user (using a fuzzy number and the associated linguistic expression in each case).

Table 1. Data in the datacube example

Fact No.	**product**	**Date**	**Age**	**Price**	**Amount**	**α**	**Fact No.**	**product**	**Date**	**Age**	**Price**	**Amount**	**α**
1	milk	23-dec	19	10	1	1	13	bread	6-jan	17	3	2	1
2	meat	7-jan	40	18	3	1	14	meat	22-dec	65	6	3	1
3	bread	10-jan	45	1	5	1	15	cheese	2-jan	52	10	2	1
4	juice	28-dec	75	2	2	1	16	bread	27-dec	66	5	2	1
5	cheese	3-jan	20	5	1	1	17	cheese	04-jan	70	5	3	1
6	milk	10-jan	20	1	5	1	18	bread	24-dec	60	3	6	1
7	bread	25-dec	22	3	1	1	19	bread	10-jan	65	4	4	1
8	bread	1-jan	55	5	2	1	20	milk	03-jan	64	5	2	1
9	juice	28-dec	23	4	3	1	21	cheese	10-jan	15	5	5	1
10	bread	6-jan	75	6	4	1	22	cheese	28-dec	40	3	5	1
11	milk	23-dec	78	3	3	1	23	bread	02-jan	65	4	5	1
12	meat	29-dec	40	18	2	1	24	milk	26-dec	23	5	5	1

4 Conclusions

In this paper, we have presented a new multidimensional model. The main contribution of this new model is that it is able to operate on data with imprecise facts and hierarchies. Classical models impose a rigid structure that makes it difficult for information from different sources to be merged if there are incompatibilities in the schemata. Our model can handle these problems by means of fuzzy logic which allows our proposal to carry out the integration, relaxing the schemata in order to obtain a new one that covers the others and attempting to preserve as much information as possible. In addition, our model can manage information given by experts which is often imprecise. This data can be used to improve the multidimensional schema so that it may be used by the final user in the decision process. Another advantage is that it can model situations to users more naturally so that they can access the information more intuitively.

Table 2. Result of applying dice on the dimension *Time*, on the level *Holiday* with the condition *β(x)="x is Christmas"* over C. In the fuzzy case, the value shown is the new α of the fact. In the crisp case, X means that this fact satisfies the condition.

Fact	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Fuzzy	1	0.9	0.6	1	1	0.6	1	1	1	1	1	1
Crisp	X	-	-	X	X	-	X	X	X	X	X	X
Fact	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
Fuzzy	1	1	1	1	1	1	0.6	1	0.6	1	1	1
Crisp	X	X	X	X	X	-	X	X	-	X	X	X

Table 3. Result of applying roll-up in the dimension *Time* on the level *Holidays*, dimension *Product* and level *Quality* and dimension *Customer* and level *Group* in the datacube C' in the fuzzy case. *Time* dimension is not shown due to the fact that there is only one value

	Product					
	Good		**Medium**		**Bad**	
Customer	**C''**	**C''_M**	**C''**	**C''_M**	**C''**	**C''_M**
Young	{1/1 0.6/3, 0.4/3.67, 0.2/3.33},1	(1,1,0,1.5) "greater than 1"	{1/1, 0.6/3, 0.3/2.88},1	(1,1,0,1.45) "greater than 1"	{1/2, 0.6/1.5, 0.2/2.4},1	(2,2,0.5,0.39) "around 2"
Adult	{1/2, 0.9/2.5, 0.6/3.4, 0.5/3.33, 0.2/3.3},1	(2,2,0,1.19) "greater than 2"	{1/3.5, 0.6/3.33, 0.3/3.44},1	(3.5,3.5,0.17,0) "a bit less than 3.5"	{1/2, 0.8/4, 0.5/3.8, 0.4/3.33, 0.2/3.3},1	(2,2,0,1.6) "grater than 2"
Old	{1/3, 0.5/2.67, 0.2/2.6},1	(3,3,0.4,0) "a bit less than 3"	{1/2, 0.8/2.5, 0.3/3.22},1	(2,2,0,1.22) "greater than 2"	{1/4, 0.6/3, 0.5/3.75, 0.3/4.2, 0.2/3.71},1	(4,4,0.29, 0.19) "around 4"

Table 4. Result of applying roll-up in the dimension *Time* on the level *Holiday*, dimension *Product* and level *Quality* and dimension *Customer* and level *Group* in the datacube C' in the crisp case

	Product		
Customer (Age group)	**Good**	**Medium**	**Bad**
Young	3	2	1.5
Adult	2	3.5	4
Old	3	2.5	3.7

Table 5. Result of applying slice on the dimensions *Product* and *Time* in the datacube C''

	Fuzzy		**Crisp**
Customer	**C'''**	**C'''_M**	**Fact**
Young	{1/2, 0.6/1.5, 0.2/2.4, 0.6/3, 0.3/2.88, 0.4/3.67, 0.2/3.33},1	(2,2,0.5,1.3) "around 2"	3
Adult	{1/3.5, 0.8/4, 0.6/3.8, 0.6/3.33, 0.3/3.44, 0.5/3.67},1	(3.5,3.5,0.17,0.5) "around 3.5"	4
Old	{1/4, 0.6/3, 0.5/3.75, 0.3/4.2, 0.2/3.71, 0.3/3.22},1	(4,4,0.99,0.2) "around 4"	3,7

In order to complete the model, we need to study the properties of the operations on the structure. Another line is to develop a graphical means of representing the results of the operations so that the information obtained may be read more intuitively. To finish the decision process, we need to study the integration process so as to obtain a formal way to merge data from different sources, including experts´ knowledge.

References

1. Agrawal, R. Gupta, A., Sarawagi, S.: Modeling Multidimensional Databases. IBM Research Report, IBM Almaden Research Center, September 1995

2. Blanco, I., Sánchez, D., Serrano, J.M., Vila, M.A.: A New Proposal of Aggregation Functions: the Linguistic Summary. Proceedings of IFSA'2003 Istanbul (Turkey) 2003

3. Cabibbo, L., Torlone, R.: A Logical Approach to Multidimensional Databases. Advances in Databases Technology (EDTB'98) No. 1337 in LNCS pp. 183-197 Springer 1998

4. Cabibbo, L., Torlone, R..: Querying Multidimensional Databases. Proceedings of the 6th Int. Workshop on databases programming languages (DBPL6) Estes Pork (U.S.A.) 1997

5. Codd, E.F.: Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. Technical report, E.F. Codd and Associates, 1993

6. Delgado, M., Martín-Bautista, M.J., Sánchez, D., Vila, M.A.: On A Characterization of Fuzzy Bags. Proceedings of IFSA'2003 Istanbul (Turkey) 2003

7. Dyreson, C.: Information Retrieval from an Incomplete Data Cube. Proceedings of the 22nd Int. Conf. on VLDB pp. 532-543. Morgan Kaufman Publishers, 1996

8. Gorry, G.A., Scott Morton, M.S.: A Framework for Management Information Systems. Sloan Management Review 13 (1) (1971) 50-70

9. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Data Mining and Knowledge Discovery 1 (1997) 29-53

10. Kimball, R.: The Data Warehouse Toolkit. Wiley, New York, 1996

11. Laurent, A., Bouchon-Meunier, B., Doucet, A.: Flexible Unary Multidimensional Queries and their Combinations. Proceedings of IPMU 2002, Annecy (France) 2002

12. Li, C., Wang, X.S.: A Data Model for Supporting On-Line Analytical Processing. Proceedings of the 5th Int. Conf. on Information and Knowledge Management (CIKM) 1996

13. Pedersen, T.B., Jensen, C.S., Dyreson, C.E.: A Foundation for Capturing and Querying Complex Multidimensional Data. Information Systems 26 (2001) 383-423

14. Rundensteiner, E.A., Bic, L.: Aggregates in Possibilistic Databases. Proceedings of the 15th Conf. on Very Large Databases (VLDB'98), Amsterdam (Holland), 287-295, 1989

15. Yager, R.R.: Aggregation Operators and Fuzzy Systems Modelling. Fuzzy Sets and Systems 67 (1994) 129-145