Using a Fuzzy Classification Query Language for Customer Relationship Management

Andreas Meier Nicolas Werro Martin Albrecht Miltiadis Sarakinos

University of Fribourg
Rue Faucigny 2
1700 Fribourg, Switzerland
andreas.meier@unifr.ch
nicolas.werro@unifr.ch

Swisscom Fixnet AG
Alte Tiefenaustr. 6
3050 Bern, Switzerland
martin.albrecht@swisscom.com
miltiadis.sarakinos@swisscom.com

Abstract

A key challenge for companies is to manage customer relationships as an asset. To create an effective toolkit for the analysis of customer relationships, a combination of relational databases and fuzzy logic is proposed. The fuzzy Classification Query Language allows marketers to improve customer equity, launch loyalty programs, automate mass customization, and refine marketing campaigns.

Keywords: Fuzzy Classification, Relational Database, Query Language, Customer Equity, Customer Relationship Management.

1. Databases & Fuzziness

In practice information systems are often based on very large data collections, mostly stored in relational databases. Due to information overload, it is becoming increasingly difficult to analyze these collections and to generate marketing decisions.

To create a toolkit for the analysis of customer relationships, a combination of relational databases and fuzzy logic is proposed. Fuzzy logic, unlike statistical data mining techniques such as cluster or regression analysis, enables the use of non-numerical values and introduces the notion of linguistic variables. Using linguistic terms and variables will result in a more human-oriented querying process. The toolkit reduces the complexity of customer data and extracts valuable hidden information through a fuzzy classification.

The fuzzy classification is achieved by extending the relational database schema with a context model. A fuzzy Classification Query Language (fCQL) can directly operate on the underlying database so that no migration of the raw data is needed. In addition, fCQL allows marketers to formulate unsharp queries on a linguistic level. To implement this, an fCQL interpreter was developed which transforms fCQL queries into SQL (Structured Query Language) statements for the sharp databases.

Fuzzy classification and fCQL has been applied to the marketing domain of a telecom company. With the fuzzy classification approach, a customer can be treated in different classes at the same time. Based on membership degrees or customer values, the company can devise appropriate marketing campaigns for acquisition, retention, and add-on selling.

A number of fuzzy query languages have been proposed in the literature. A well-known enhancement of the relational domain calculus is the Fuzzy Query Language proposed by Takahashi (1995). It provides a theoretical basis for the development of a human-oriented interface with relational databases. Another query language is the fuzzy SQL published by Bosc and Pivert (2000). This language allows gradual predicates interpreted in the framework of the fuzzy set theory. Finally, there is FQUERY from Kacprzyk and Zadrozny (2000) which extends Microsoft's database system Access in order to answer imprecisely specified questions.

The remainder of the paper is organized as follows: Section 2 introduces a fuzzy classification approach by defining a context model, outlining the fuzzy Classification Query Language and providing an overview of the fCQL toolkit. The application of fCQL to relationship management is explained in section 3; in particular, customer equity, mass customization, and loyalty programs have been derived from fuzzy customer

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, requires a fee and/or special permission from the Endowment

Proceedings of the 31st VLDB Conference,
Trondheim, Norway, 2005

classes. Section 4 summarizes the pros and cons of fuzzy classification. Section 5 suggests future research topics for the database community.

2. Fuzzy Classification

2.1 Context Model

The relational model is extended by a context model in order to obtain a classification space. To every attribute A_j defined by a domain $D(A_j)$ there is added a context $K(A_j)$. A context $K(A_j)$ of an attribute is a partition of $D(A_j)$ into equivalence classes (see Shenoi 1995). In other words, a relational database schema with contexts R(A,K) consists of a set $A=(A_1,\ldots,A_n)$ of attributes with associated contexts $K=(K_1(A_1)),\ldots,K_n(A_n))$.

Throughout this paper, a simple example from relationship management is used. For simplicity, customers will be evaluated by only two attributes, turnover and payment behaviour. In addition, these two qualifying attributes for customer equity will be partitioned into only two equivalence classes. The pertinent attributes and contexts for relationship management are:

- Turnover in dollars per month: The attribute domain is defined by [0..1000] and divided into the equivalence classes [0..499] for low and [500..1000] for high turnover.
- Payment behaviour: The domain {in advance, on time, behind time, too late} with its equivalence classes {in advance, on time} for an attractive payment behaviour and {behind time, too late} for a non attractive one.

To derive fuzzy classes from sharp contexts, the qualifying attributes are considered as linguistic variables, and verbal terms are assigned to each equivalence class. With linguistic variables (see Fig. 1), the equivalence classes can be described more intuitively. In addition, every term of a linguistic variable represents a fuzzy set. Membership functions μ (see Zimmermann 1992) are defined for the domains of the equivalence classes (see also Fig. 2).

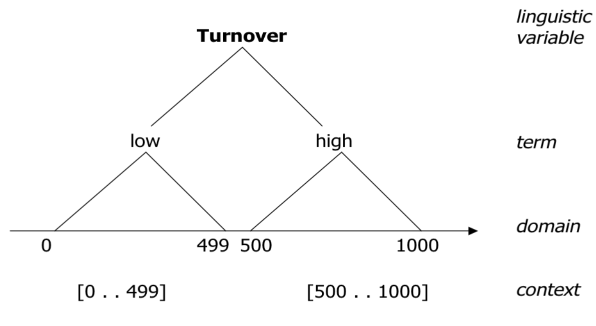

Fig. 1: Concept of linguistic variable

As turnover is a numeric (sharp) attribute, its membership functions $\mu_{\text{high turnover}}$ and $\mu_{\text{low turnover}}$ are continuous functions defined on the whole domain of the attribute. For qualitative attributes like payment behaviour, step functions are used; the membership functions $\mu_{\text{attractive payment behaviour}}$ and $\mu_{\text{non attractive payment behaviour}}$ define a membership grade for every term of the attribute's domain.

The selection of the two attributes turnover and payment behaviour and the corresponding equivalence classes determine a two-dimensional classification space (see Fig. 2). The four resulting classes C1 to C4 could be characterized by marketing strategies such as 'Commit Customer' (C1), 'Improve Loyalty' (C2), 'Augment Turnover' (C3), and 'Don't Invest' (C4).

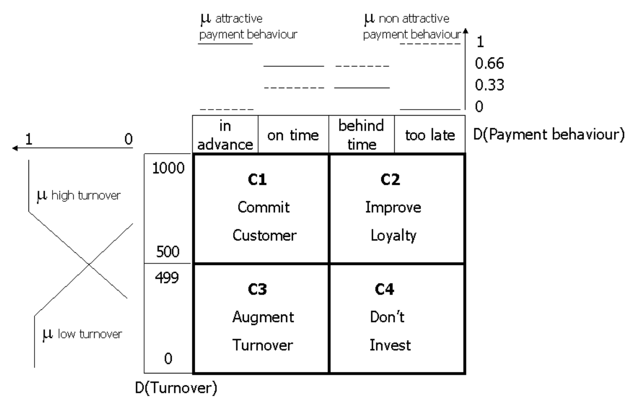

Fig. 2: Fuzzy classification space defined by turnover and payment behaviour

With this context model and the usage of linguistic variables and membership functions, the classification space becomes fuzzy. A fuzzy classification of customers has many advantages compared with the common sharp classification approaches (see section 4 for a summary). Most importantly, with fuzzy classification a customer can belong to more than one class at the same time. This leads to differentiated marketing concepts and helps to improve customer equity.

The selection of qualifying attributes, the introduction of equivalence classes and the choice of appropriate membership functions are important design issues. Database architects and marketing specialists have to work together in order to make the right decisions.

2.2 Fuzzy Classification Query Language fCQL

The Structured Query Language SQL is the standard for defining and querying relational databases. Adding to the relational database schema a context model with linguistic variables and fuzzy sets, the query language has to be extended. The proposed extension is the fuzzy Classification Query Language fCQL, originally described in Schindler (1998):

```
<ClassificationQuery> =
   classify <Object> from <Relation>
   { with <ClassficationCondition> }

<Object> =
   ColumnDefinition

<Relation> =
   RelationIdentifier |
   ViewIdentifier

<ClassificationCondition> =
   <ClassSelection> |
   <AttributeSelection> |
   ( <AttributeSelection> )
   or ( <AttributeSelection> )
   { or ( <AttributeSelection> ) }

<ClassSelection> =
   <ClassCondition>
   { or <ClassCondition> }

<ClassCondition> =
   class is <Description>

<Description> =
   ColumnDefinition

<AttributeSelection> =
   <AttributeCondition>
   { and <AttributeCondition> }

<AttributeCondition> =
   <Attribute> is <EquivalenceClass> |
   ( <Attribute> is <EquivalenceClass>
     or <EquivalenceClass>
   { or <EquivalenceClass } )

<Attribute> =
   ColumnDefinition

<EquivalenceClass> =
   ColumnDefinition
```

The classification language fCQL is designed in the spirit of SQL. Instead of specifying the attribute list in the select-clause, the name of the object column to be classified is given in the classify-clause. The from-clause specifies the considered relation, just as in SQL. Finally, the where-clause is changed into a with-clause which does not specify a predicate for a selection but a predicate for a classification. An example in customer relationship management could be given as follows:

```
classify    Customer
from        CustomerRelation
with        Turnover is high and
            PaymentBehaviour is attractive
```

This classification query would return the class C1, i.e. the class with the semantic 'Commit Customer'. This class was defined as the aggregation of the terms 'high' and 'positive'. As an aggregation operator the γ-operator (see Zimmermann 1992) is adequate, because it is an operator between the conjunction and disjunction depending on the value of the γ-argument.

In this simple example, specifying linguistic variables in the with-clause is straightforward. In addition, if customers have to be classified on three or more attributes, the capability of fCQL for a multi-dimensional classification space is increased. This can be seen as an extension of the classical slicing and dicing operators on a multidimensional data cube.

2.3 Architecture of fCQL Toolkit

As noted above, the fuzzy classification is achieved by extending the relational database schema. This extension consists of meta-tables added to the system catalogue (see Appendix). These meta-tables contain the definition of the linguistic variables and their associated terms, the definition and the description of the classes and all the meta-information regarding the membership functions.

The architecture of the fCQL toolkit is shown in Fig. 3 which illustrates the interactions between the user and the different fCQL toolkit components. The fCQL toolkit is an additional layer above the relational database system (see Meier et al. 2001). This makes fCQL independent of underlying commercial systems. It also implies that the user can always query the database with standard SQL (see case 1 in Fig. 3).

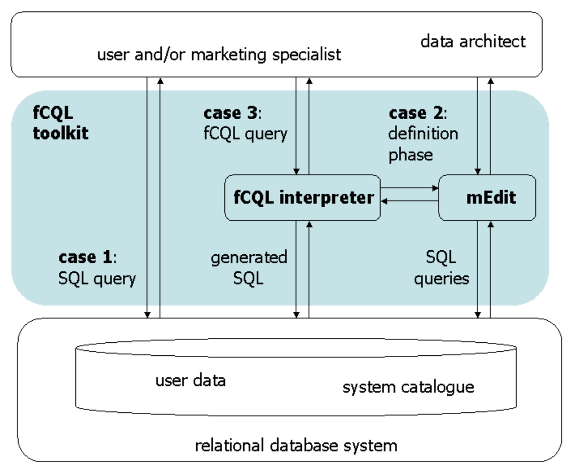

Fig. 3: Overview of fCQL toolkit

The architecture of the fCQL toolkit consists of two main components, the fCQL interpreter and the editor mEdit. The mEdit (see case 2) helps the data architect to select appropriate attributes, and to define equivalence classes, linguistic variables, linguistic terms and membership functions (Dombi 1991). mEdit communicates with the underlying database via classical SQL statements.

The fCQL interpreter allows the user to formulate unsharp queries (case 3). These queries are analyzed and translated into corresponding SQL statements. The resulting relation helps the fCQL interpreter to compute the membership degrees of the classified elements and provides the fuzzy classification for the user.

3. Fuzzy Customer Classes

3.1 Customer Equity

Managing customers as an asset requires measuring them and treating them according to their true value (see Blattberg et al. 2001). With sharp classes, i.e. traditional customer segments, this is not possible. In Fig. 4 for instance, customers Brown and Ford have similar turnover as well as similar willingness to pay. However, Brown and Ford are treated in different classes: Brown belongs to the winner class C1 (Commit Customer) and Ford to the loser class C4 (Don't Invest). In addition, a traditional customer segment strategy treats the top rating customer Smith the same way as Brown, who is close to the loser Ford.

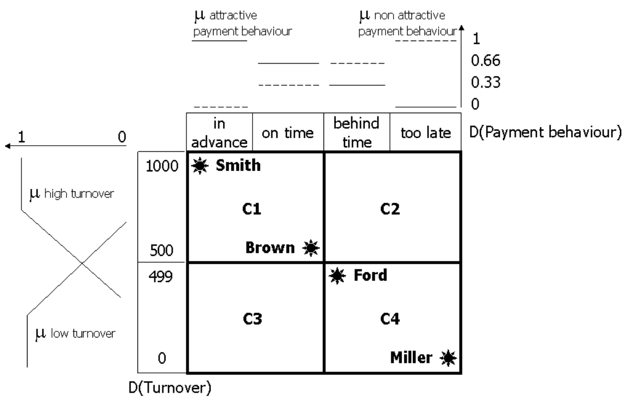

Fig. 4: Customer equity example based on turnover and payment attitude

The main difference between a traditional classification and a fuzzy one is that in the fuzzy classification a customer can belong to more than one class. Belonging to a fuzzy class implies a degree of membership. The notion of membership functions brings the disappearance of sharp borders between customer segments. Fuzzy customer classes reflect reality better and allow companies to treat customers according to their real value.

3.2 Mass Customization

Another advantage of fuzzy classification in relationship management is its potential for mass customization. The membership degree of customers can determine the privileges they get, for example a personalized discount according to the customer value. Discount rates can be associated with each fuzzy class: for instance C1 gets a discount rate of 10% (Commit Customer), C2 one of 5% (Improve Loyalty), C3 3% (Augment Turnover), and C4 0% (Don't Invest). The individual discount of a customer could be calculated by the aggregation of the discount of the classes he belongs to in proportion to his membership degrees.

The top rating customer Smith belongs 100% to class C1 because he has the highest possible turnover as well as the best paying behaviour; the membership of Smith in class C1 would be written as Smith (C1:1.0). Customer Brown belongs to all four classes and would be rated as (C1:0.31, C2:0.20, C3:0.30, C4:0.19). With fuzzy classification, the customers of Fig. 4 get the following discounts:

- Smith (C1:1.0): 1.0 * 10% = 10%
- Brown (C1:0.31, C2:0.20, C3:0.30, C4:0.19):
  0.31*10%+0.20*5%+0.30*3%+0.19*0% =5 %
- Ford (C1:0.19, C2:0.30, C3:0.20, C4:0.31):
  0.19*10%+0.30*5%+0.20*3%+0.31*0% = 4%
- Miller (C4:1.0): 1.0 * 0% = 0%

Using fuzzy classification for mass customization leads to a transparent and fair judgment: Smith gets the maximum discount and a better discount than Brown who belongs to the same customer class C1. Brown and Ford get nearly the same discount rate. They have comparable customer values although they belong to opposite classes. Miller, who sits in the same class as Ford, does not benefit from a discount.

3.3 Customer Loyalty

Many loyalty concepts have been proposed in the marketing literature. Harrison (2000), for instance, proposes two important dimensions, attachment and behaviour of customers. For simplicity again, only two attributes (attachment, repurchases) and four classes will be considered: Class L1 (True Loyalty) with high attachment and numerous repurchases, class L2 (Latent Loyalty) with high attachment but few repurchases, class L3 (Pseudo Loyalty) with low attachment but many repurchases, and finally, L4 (No Loyalty) with low attachment and few repurchases.

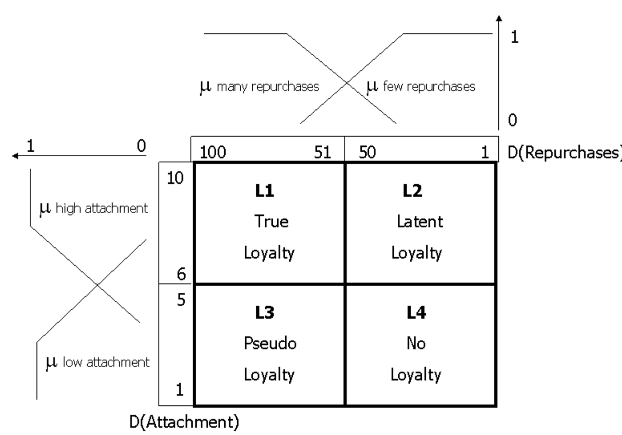

Fig. 5: Fuzzy concept for loyalty

The four fuzzy classes for customer loyalty with appropriate membership functions are illustrated in Fig. 5. This classification scheme can be used to improve the original customer classes of Fig. 2 with the classes C1 to C4. The attribute payment behaviour is replaced by the loyalty concept because willingness to pay is too weak to express the fidelity of customers (see Fig. 6).

3.4 Marketing Campaign

Launching a marketing campaign can be very expensive. It is therefore crucial to select a customer group with potential. Fuzzy classification offers considerable advantages when planning and selecting customer subgroups.

An example is given in Fig. 6. One strategy could be to select the most loyal customers or customers with low turnover. Using membership functions, a subset of customers in class C1 and C3 has been chosen. The application of membership functions allows marketers to evaluate attractive customers in relation to the available campaign budget.

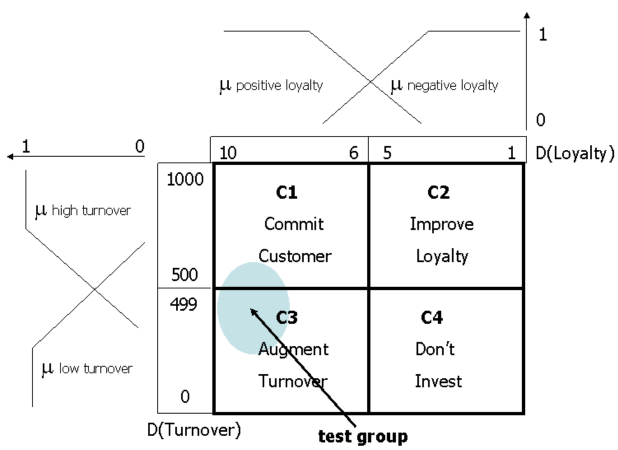

Fig. 6: Selecting customers for a marketing campaign

If the marketing campaign or a testing process has been started, the fuzzy customer classes can be analyzed again. It is important to find out if the invested money is moving the customers in the right direction, i.e., improving their customer value.

4. Fuzzy vs. Sharp Classification

Fuzzy logic aims to capture the imprecision of human perception and to express it with appropriate mathematical tools. With fuzzy classification, marketers are able to use linguistic variables, e.g. loyalty, and linguistic terms, e.g. positive or negative.

There are a number of advantages in using fuzzy classification for relationship management:

- Fuzzy logic, unlike statistical data mining, enables the use of non-numerical attributes. As a result, both qualitative and quantitative attributes can be used for marketing acquisition, retention, and add-on selling.
- With the help of linguistic variables and terms, marketers may describe equivalence classes more intuitively (excellent loyalty, medium loyalty, weak loyalty). The definition of linguistic variables and terms and the naming of fuzzy classes can be derived directly from the terminology of marketing and sales departments.
- Customer databases or data cubes can be queried on a linguistic level. The fuzzy Classification Query Language allows marketers to classify single customers or customer groups by classification predicates such as 'loyalty is positive and turnover is high'.

An important difference between a fuzzy classification and a sharp one is the fact that a customer can belong to more than one fuzzy class. In classical marketing programs, groups or segments of customers are typically constituted by a small number of qualifying attributes. If corresponding data values are similar for two customers, their membership functions are similar too. In the classical case however, they may fall into different classes and be treated differently (customers Brown and Ford in Fig. 4).

With fuzzy classification it is possible to treat each customer individually. This allows managers to allocate marketing budgets more precisely. In addition, cost savings can be achieved. For instance, when offering a discount (see section 3.2), discount rates can be chosen according to the individual customer value. Companies can try to retain the more profitable customers by giving them individualized privileges.

Needless to say there are also drawbacks when applying fuzzy classification. The definition of attributes, equivalence classes and membership functions remains a challenging task. In our experience, the design of fuzzy classes requires marketing specialists as well as data architects. Beyond this, a methodology is needed for the entire planning, designing, and testing process.

5. Suggestions for Future Research

Database technology remains one important basis for developing customer relationship programs and optimizing marketing processes (see Meier 2003). As concepts such as customer equity, loyalty, mass customization, and closely tailered marketing campaigns will be in increasing demand in all industrial sectors, the use of fuzzy concepts will need to evolve. The following topics should therefore be studied and researched, by both practitioners and database researchers:

- *Fuzzy data models*: Classical data models cannot handle uncertain and imprecise information very well. Typically, uncertainty and imprecision become a matter of concern in many applications in economics, control theory, customer relationship management, performance measurement, and many other fields. For this reason, the theory of fuzzy relational models should also be considered and studied by the database community.

- *Reduction of information overload*: The onset of information in web-based applications asks for reducing the complexity of data. Defining meta layers and ontologies is a must in knowledge-based information systems. In addition, the meta information should be defined in terms of the user, e.g. by introducing linguistic variables.
- *Visualization of multi-dimensional data*: Defining fuzzy classes implies that one entity can belong to several classes. The visualization of fuzzy classes and subclasses, e.g., the result of a fuzzy querying process, has to be examined. For instance, a browser for multi-dimensional cubes, spheres, ellipsoids or cons has to be developed (see the selection of test groups for a marketing campaign in Fig. 6).
- *Design methodology for fuzzy knowledge discovery in databases*: The selection of qualifying attributes, the definition of equivalence classes and membership functions, and the naming of linguistic variables and terms are all important tasks during the design process of a fuzzy classification. A design methodology for this complex process is needed.
- *Fuzzy data warehouses*: Introducing equivalence classes of attributes produces a multi-dimensional fuzzy data cube. The main operators of a data warehouse, i.e. drill-down, roll-up, slicing, and dicing have to be extended to operate in a fuzzy data warehouse environment. Specific aggregation operators and/or composition/decomposition algorithms should be provided.
- *Fuzzy query and classification languages*: As briefly discussed in section 1, there are a number of proposals for extending query languages. Analysis and comparison of these approaches should be carried out in order to develop a standard extension of the widely used query language SQL.
- *Architectures and technical frameworks*: Storing the data in relational databases and developing a separate layer for fuzziness on top of a database system is a straightforward approach. However, fuzzy data and classes are multi-dimensional and should therefore be stored in multi-dimensional data structures.

These research questions illustrate the need to combine database technology with fuzzy logic. Although some methods and technical concepts have to be extended, a fuzzy classification approach with an appropriate classification query language remains a fruitful toolkit for customer relationship management.

Acknowledgements: Most of this work is based on experience gained from the joint venture between the telecom company Swisscom and the University of Fribourg. The authors would like to thank all their professional colleagues for developing the fCQL toolkit: Christian Mezger, Christian Nancoz, Christian Savary, Günter Schindler, and Yauheni Veryha. In addition, the authors are grateful for the support of Swisscom management, especially René Vaucher.

Literature

Blattberg, R. C., Getz, G., Thomas, J. S.: Customer Equity – Building and Managing Relationships as Valuable Assets. Harvard Business School Press, Boston, 2001.

Bosc, P., Pivert, O.: SQLf Query Functionality on Top of a Regular Relational Database Management System. In: Pons, O., Vila, M. A., Kacprzyk, J.: Knowledge Management in Fuzzy Databases. Physica Publisher, Heidelberg, 2000, pp. 171-190.

Dombi, J.: Membership Functions as an Evaluation. In: Fuzzy Sets and Systems. Volume 35, Elsevier, 1991, pp. 1-21.

Harrison, T.: Financial Services Marketing. Pearson Education, Essex, 2000.

Kacprzyk, J., Zadrozny, S.: Data Mining via Fuzzy Querying over the Internet. In: Pons, O., Vila, M. A., Kacprzyk, J.: Knowledge Management in Fuzzy Databases. Physica Publisher, Heidelberg, 2000, pp. 211-233.

Meier, A.: Customer Relationship Management – Opportunities & Risks when Applying IT. Proc. of the 2nd International Conference on Management Education for the 21st Century, Hanoi, Vietnam, September 10-12, 2003, pp. 22-29.

Meier, A., Savary, Ch., Schindler, G., Veryha, Y.: Database Schema with Fuzzy Classification and Classification Query Language. Proc. of the International Congress on Computational Intelligence – Methods and Applications, University of Wales, Bangor, U.K., June 19-22, 2001.

Schindler, G.: Fuzzy Data Analysis Through Context-Based Database Queries (in German). PhD Thesis No. D82, Technical University of Aachen, Germany, 1998.

Shenoi, S.: Fuzzy Sets, Information Clouding and Database Security. In: Bosc, P., Kacprzyk, J.: Fuzziness in Database Management Systems. Physica Publisher, Heidelberg, 1995, pp. 207-228.

Takahashi, Y.: A Fuzzy Query Language for Relational Databases. In: Bosc, P., Kacprzyk, J.: Fuzziness in Database Management Systems. Physica Publisher, Heidelberg, 1995, pp. 365-384.

Zimmermann, H.-J.: Fuzzy Set Theory – and Its Applications. Kluwer Academic Publishers, London, 1992.

Appendix: Schema of the meta-tables

The meta-information necessary for a fuzzy classification consists of eight meta-tables added to the system catalogue. The relationships between the meta-tables are shown in Fig. 7.

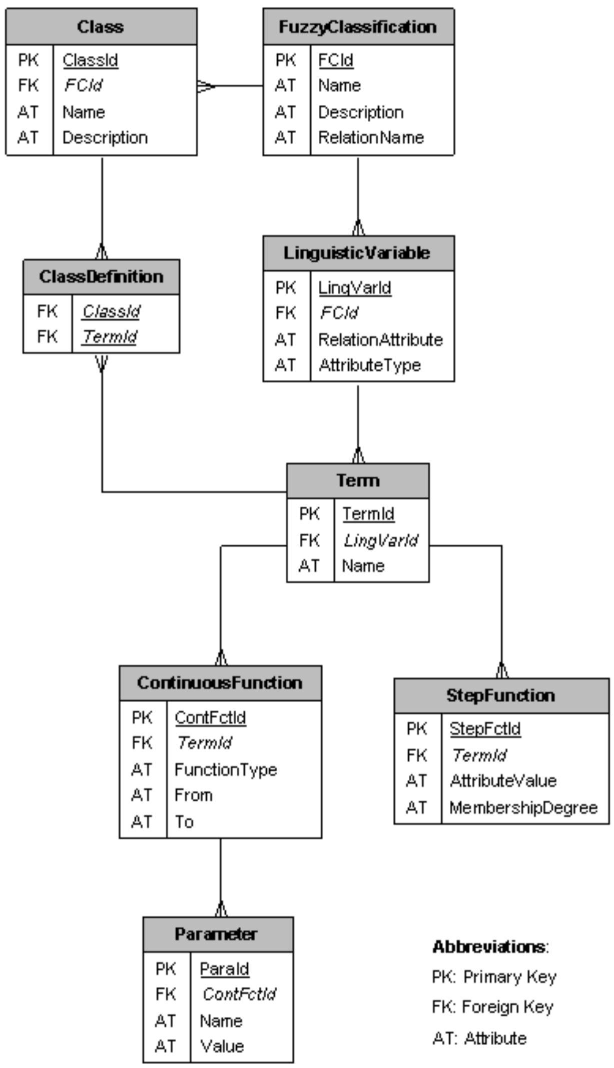

Fig. 7: Schema of the meta-tables

For the customer equity example discussed in Section 3, it is assumed that a customer relation (resp. view) contains the elements to be classified as well as the qualifying attributes (see Table 1).

Table 1: Relation containing the elements to be classified

Customer				
CustId	Customer	Turnover	PaymentBehaviour	…
1	Smith	990	in advance	…
2	Brown	510	on time	…
3	Ford	490	behind time	…
4	Miller	10	too late	…
…	…	…	…	…

The first meta-table shown in Table 2 defines a fuzzy classification by giving it a name, a description and by specifying which relation holds the elements to be classified. It is possible to define several fuzzy classifications on the same relation as the classification criteria may be different.

Table 2: Meta-table defining a fuzzy classification

FuzzyClassification			
FCId	Name	Description	RelationName
1	CRM example	…	Customer

The second step is to define the linguistic variables (the dimension of the classification space) by assigning to every linguistic variable an attribute of the data relation (see Table 3). The type of the attribute (e.g. numeric or non-numeric) will determine the use of continuous or step membership functions.

Table 3: Meta-table specifying the linguistic variables

LinguisticVariable			
LingVarId	*FCId*	RelationAttribute	AttributeType
1	1	Turnover	numeric
2	1	PaymentBehaviour	non-numeric

For every linguistic variable, several terms can be defined in order to express the semantic of the equivalence classes (see Table 4).

Table 4: Meta-table identifying the linguistic terms

Term		
TermId	*LingVarId*	Name
1	1	high
2	1	low
3	2	attractive
4	2	nonattractive

For the terms referring to a numeric attribute, several (continuous) functions can be composed in order to draw their membership function. In the discussed example, the membership functions were only made of linear functions but the generic S-shaped function proposed by Dombi (1991) can also be used. All the functions are implemented in the application code for efficiency reasons so that the meta-tables only contain the function's type, interval and parameters. For instance Table 5 shows that the membership function of the term 'high turnover' (TermId 1) is made up of three linear functions.

Table 5: Meta-table holding the continuous functions

ContinuousFunction				
ContFctId	*TermId*	FunctionType	From	To
1	1	Linear	0	332
2	1	Linear	333	750
3	1	Linear	751	1000
…	…	…	…	…

As the parameters of a linear function are different from those of an S-shaped function, another meta-table is required. This meta-table, shown in Table 6, contains the parameters of all the functions listed in Table 5. In order to be generic, a parameter is described by a name and a value. This approach also allows the implementation of new types of function for specific needs.

Table 6: Meta-table listing the functions' parameters

Parameter			
ParaId	*ContFctId*	Name	Value
1	1	startvalue	0
2	1	endvalue	0
3	2	startvalue	0
4	2	endvalue	1
5	3	startvalue	1
6	3	endvalue	1
…	…	…	…

For the terms based on a non-numeric attribute, a membership degree has to be defined for every value of the attribute's domain (see Table 7). All the defined values for a given term will draw its step membership function.

Table 7: Meta-table listing the step functions

StepFunction			
StepFctId	*TermId*	AttributeValue	MembershipDegree
1	3	in advance	1
2	3	on time	0.66
3	3	behind time	0.33
4	3	too late	0
…	…	…	…

The definition of the linguistic variables and their associated terms will determine a classification space. Each resulting class has to be named and described with its particular semantic (see Table 8).

Table 8: Meta-table describing the classes

Class			
ClassId	*FCId*	Name	Description
1	1	C1	Commit Customer
2	1	C2	Improve Loyalty
3	1	C3	Augment Turnover
4	1	C4	Don't Invest

In the given example, each class is defined by two terms. The last meta-table associates the classes and the terms together (see Table 9). This association allows the calculation of the membership degree of the classified elements in the classes, based on the partial membership degrees of the associated terms.

Table 9: Meta-table defining the classes

ClassDefinition	
ClassId	*TermId*
1	1
1	3
2	1
2	4
3	2
3	3
4	2
4	4

For simplicity some non fundamental attributes of the meta-tables have been ignored. The presented meta-schema should however show the clear separation between the user's data and the meta-tables as well as the ease of installation of the fCQL toolkit.