INSTITUTO DE ENGENHARIA NUCLEAR

RT-IEN- 07/2002

Fuzzy Unified Modeling Language Approach

por

Antonio Cesar Ferreira Guimarães
Leonardo Falcao Koblitz

Agosto/2002

INSTITUTO DE ENGENHARIA NUCLEAR

Título: **Fuzzy Unified Modeling Language Approach.**

Autor(es): Antonio C. F. Guimarães e Leonardo F. Koblitz.	e-mail: tony@ien.gov.br e lfalcao@ien.gov.br

Identificação: RT-IEN-07/2002	Nº de páginas: 18	Tipo de Divulgação: Irrestrita (x) Restrita ( )	Divulgar para: IEN	Localização: DIRE/SETER e DIAT/SEINF

Publicação externa associada (congresso/periódico): Fuzzy Sets and Systems (p/aprovação)

Palavras chave: Fuzzy, Object Oriented, Data Model, UML.

Resumo:

Este relatório contém a descrição de uma proposta preliminar de modelagem de dados, utilizando uma solução híbrida reunindo a lógica difusa e a linguagem de modelagem unificada. Em recentes estudos de modelagem de dados orientados à objetos, modelos de objetos difusos (*fuzzy*) baseado em grafos e usando outras representações, eram normalmente utilizados para tratamento de dados imprecisos em bases de dados. Nesta época, não existia ainda uma modelagem de dados consagrada e reconhecida internacionalmente. Hoje, a UML, *Unified Modeling Language*, é a linguagem de modelagem padrão utilizada para representação gráfica. Unindo este tipo de representação e a teoria de lógica difusa foi definido e proposto neste trabalho uma modelagem, *fuzzyUML*, que represente graficamente o problema analisado. Para efeito de ilustração, um caso exemplo foi apresentado.

Abstract:

The last approach involving the increasing complexity of real applications in the field of multimedia information systems was proposed in "A Fuzzy Object Oriented Data Model", where the authors make the contribution starting from an existing graph-based object model, a Fuzzy Object Oriented Data (FOOD) model for the management of imprecise data. In that time there wasn't the UML – Unified Modeling Language, forces joined between Grady Booch, Jim Rumbaugh, and Ivar Jacbson, to form a single modeling language. In this paper will be proposed a new approach using a hybrid methodology with Fuzzy and UML, named FuzzyUML. With this hybrid methodology, making use of state of art for model language, we can finally define a fuzzy extensions of the Object Oriented Data Models (OODMs) in order to deal with imprecise and uncertain data. An application of this method to problem of the dinner in a restaurant to determine the tip is presented.

Emissão		Nome	Rubrica	Data
Data: 28/08/2002	Elaboração:	Antonio C. F. Guimarães e Leonardo Falcao Koblitz		28/08/2002
Divisão: DIRE e DIAT	Revisão:	Orlando J. A. Gonçalves Fº		28/08/2002
Serviço: SETER e SEINF	Aprovação :	Orlando J. A. Gonçalves Fº		28/08/2002

Instituto de Engenharia Nuclear:
Via 5 s/n, Cidade Universitária, Ilha do Fundão, CEP 21945-970, CP 68.550, Rio de Janeiro – RJ - Brasil .
Tel.: 00 55 21 2560-4113 Internet: www.ien.gov.br

1. INTRODUCTION

The definitive guides to UML-Unified Modeling Language and the Objectory Software Development Process can be find in many titles today (Object Technology Series - Addison-Wesley). In the other hand, fuzzy extensions of the Object Oriented Data Models (OODMs) have been proposed, in which imprecision and uncertainty are managed both at the level of object attributes and relations. Fuzzy Object Oriented Data Model (FOOD) [1] is the methodology for management of crisp and fuzzy data, with consideration of fuzzy data, means both imprecise data and uncertain information. Other important work [5] presents a modeling approach wich couples fuzzy object-oriented database modeling with fuzzy logic. The modeling approach introduced there handless fuzziness at attribute, object/class and class/superclass levels in adittion to fuzziness in class/class relation and various associations among classes. Logical rules are defined to crisp/fuzzy relationships and associations which cannot be presented esasily with object-oriented modeling features alone in the class hierarchies.

In this point is very import make the difference between modeling language and process. Most methods consist, at least in principle, of both a modeling language and a process. The **modeling language** is the (mainly graphical) notation that methods use to express designs. The **process** is their advice on what steps to take in doing a design. The UML discussed here is the successor to object-oriented analysis and design (OOA&D) methods.

After this explanation, the new approach extend the FOOD [1] proposed before, because a graphical notation is necessary for discuss any design with someone. Our work is pointed out to graphical notation not a process phase to developing a Smalltalk or C++ system.

2. FUZZY UML APPROACH

There is two different kind of approach to applying here: (i) when you use the UML graphical notation to define the conceptual scheme to represent a *direct labeled graph*, like in [1][4] but with state of art, for the notation graphic, and (ii) identify some imprecise data and uncertain information during definition of the *associations* and *subtypes*, i.e., two principal kinds of static relationships of *a class diagram.*

In this paper we recommend a new approach using the second consideration, and then a new unified modeling language is necessary, i.e., a *fuzzy*UML.

2.1 – FUZZY USE CASES

Why Fuzzy Use Cases? [see Fuzzy Use Case Diagram (Figure 1)] The actors represent what interacts with the system. They represent everything that needs to exchange information with the system. As read in [Jacobson et all, 1992], we differentiate between actors and users [3]. The users is the actual person who uses the system, whereas an actor represents a certain role that a user can play. We regard an actor as a class and users as instances of this class. These instances exist only when the user does something to the system. The same person can thus appear as instances of several different actors, but now with different degree of membership of that class. Looking at the Figure 1 and according with [Fowler & Scott, 1997], we can see that there will probably be many trades in the given organization, but as far as the system is concerned, they all play the same role. A user may also play more than one role. For instance, one senior trader may play the Trading Manager role and also be a regular trader; a Trader may also be a Salesperson. Everything between actors and users, can play a Fuzzy reference relation and Fuzzy instance of relation [1].

2.2 - FUZZY CLASS DIAGRAMS: THE ESSENTIALS AND ADVANCED CONCEPTS

- **Essentials**

If we think about we can have *FuzzyAssociation* and *FuzzyGeneralization*. When the system use some part of the class, the situation is typically *FuzzyAssociation*, i.e., no crisp value must be adopted but a membership value for each instance of the class, and in the real life situation when we generalize, many *types* it can be presented in the superior class describing the *FuzzyGeneralization* and a set of membership must be defined for generalization. This idea is the same proposed em [1], with difference in respect to terminology.

Now we can define a new cardinality and display the Figure 2 to show these comments.

- *Fuzzy Cardinality Notations*

A -------^{1}**B** membership of the A is always associated with one membership of the B

A -------***B** membership of the A is always associated with one or more membership of the B.

- *Classification* refers to the relationship between an object and its type.

- *Fuzzy Multiple Classification* - In fuzzy single classification, object belongs to a fuzzy single type, which may inherit from fuzzy supertypes. You have to look for new cardinality for understand fuzzy single classification and fuzzy supertypes. In fuzzy multiple classification, object may be described by several fuzzy types with partial membership degree. See the Figure 3, where a typical Fuzzy Multiple Classification is showed.

- *Fuzzy Dynamic classification* allows objects to change type within the sub-typing structure; fuzzy static classification does not. With fuzzy static classification, a separation is made between fuzzy types and fuzzy states; dynamic classification combines these notions. See the Figure 4 to a typical Fuzzy Dynamic Classification.

- *Fuzzy Aggregation and Fuzzy Composition* - with aggregation it is the part-of fuzzy reference relationship [1]. With fuzzy composition, the fuzzy part object may belong to only one fuzzy whole. At the Figure 5 shows example of these constructs. In this example, a polygon contains a fuzzy number (high, fewer, ..) ordered collection of points. Fuzzy Composition is used for the relationship to the Graphics Bundle.

Following this way, we can define many others fuzzy concepts, using the UML notation:

- *Fuzzy Derived Associations and Attributes* - are those that can be calculated from other fuzzy associations and fuzzy attributes, respectively, on a fuzzy class diagram. For example, an fuzzy age attribute (young) of a Person can derived if you know that fuzzy Person's date of birth attribute.

- *Fuzzy Reference Objects and Fuzzy Value Objects* - One of the common things said about fuzzy objects is that they have a membership function. This is true, but it is not quite as simple as that. In practice, you find that identity is important for reference objects, but not so important for value objects. For example, it is normal to have hundreds of objects, or membership's function, that designate the same universe of discourse.

3. FUZZY UML APPLICATION

The case example of a dinner accomplished in a restaurant will be presented, involving customers, waiters, cooks and managers, where intends to attribute a value for the **tip** to be pays, due to the services rendered by the employees more the quality of the foods offered by the restaurant. Firstly, the diagram of **use case** will be presented to understand the problem that will be analyzed. In the Figure 6, in a schematic way will be introduced **the actors** with its **relationships**. The use case represents the functionality of the system or a classifier, as a

subsystem or a class. The diagram of fuzzyclasses of the analyzed problem, is presented in the Figure 7.

Repair in the Figure 7 that we could expand the class food, using the inheritance concept, where we could have food with class derived of the type vegetable classes, meat, and fish. The class vegetable could derive in two or more classes of the vegetable type. The same reasoning can be used for the classes meat and fish. Now, as would be a complete rule using the previous classes:

If A is x and Vegetable is light and food is delicious OR service is excellent **then** tip=generous.

Considering now the classes *service*, *food* and *tip*. The rules for a new situation are:

1. **If** Service is poor or food is rancid **then** tip=cheap.
2. **If** service is good **then** tip=average.
3. **If** service is excellent or food is delicious **then** tip=generous.

As the objective of this work is a proposal preliminary of graphic modeling using fuzzy and UML, the use of a programming language is recommended, of the type C++ or Java, to write a program (main) that uses these three classes defined with the concept of FuzzyUML, considering:

- inputs and fuzzy inputs.
- Apply Fuzzy Operation (OR=Max).
- Apply Implication Method (Min).
- Apply Method of Aggregation (Max).
- Defuzzify the aggregation to output (Centroid).

This example will be presented in another way, using two Figures, to assist the programmer's understanding, with relationship to the necessary steps that should be done, using the proposal described previously through FuzzyUML. The Figure 8 is summary of if-then rules: (i) Fuzzify inputs – the inputs are crisp (non-fuzzy) numbers limited to a especific range, (ii) All rules are

evaluated in parallel using fuzzy reasoning, (iii) the results of the rules are combined and distilled (defuzzified), and (iv) the results is a crisp (non-fuzzy) number. The Figure 9, the real full-size fuzzy inference diagram, show everything is needed to do with (i) fuzzy inputs, (ii) fuzzy operation (OR=max.), (iii) implication method (min), (iv) aggregation method (max), (v) defuzzify (centroid), and output crisp value. Only one rule is represented at the Figure 9.

4. FUTURE WORKS

Many others concepts for future developments and study are necessary to define. Some of these are described following:

- *Fuzzy Collections for Multi-Valued Roles* - is one whose multiplicity's upper bound is greater than 1 (for instance, *). The usual convention is that multi-valued roles are thought of as fuzzy sets.
- *Fuzzy Immutability* - is a constraint that is usually applied to an fuzzy attribute, a role, or a class.
- *Fuzzy Qualified Associations* - The fuzzy qualifier says that in connection with an Order class, there may be one Order line for each instance of Product.
- *Fuzzy Association Class* - allow you to add fuzzy attributes, operations, and other features to associations.
- *Fuzzy Parameterized Class* - this concept is most obviously useful for working with collections in a strongly typed language. This way, you can define behavior for sets in general by defining a fuzzy template class set.

Some sugestions are presented, but many others concepts are necessary to define.

5. CONCLUSION

The approach proposed in this paper is begining for complete definition of Fuzzy Unified Modeling Language for graphic notation. We can in the future, define a complete Fuzzy Unified Modeling and the Fuzzy Objectory Software Development Process. The author believe that after this work the FuzzyUML can be used like a superset of UML in fuzzy problem, where imprecision and uncertainty of the data need of the specific notation.

REFERENCES

[1] G. Bordogna, D. Lucarella, *A Fuzzy Oriented Data Model*, 1994 IEEE. (http://www.itim.mi.cnr.it/Staff/Bordogna/pubblicazioni.html)

[2] M. Fowler, K. Scott, UML Distilled - Applying the Standard Object Modeling Language, Series Editors, ADDISON-WESLEY, 1997.

[3] I. Jacobson, Object-Oriented Software Engineering - A Use Case Driven Approach, ADDISON-WESLEY, 1992.

[4] Assis, A, "INTELLIGENT SYSTEM FOR FAILURE DETECTION WITH NON-MONOTONIC REASONING", Master thesis – BSc, Federal University of Rio de Janeiro – COPPE / NUCLEAR, RJ – BRASIL, may 1998.

[5] Yazici, A., Koyouncu, M., "Fuzzy object-oriented database modeling coupled with fuzzy logic", Fuzzy Sets and Systems 89 (1997) 1 – 26.

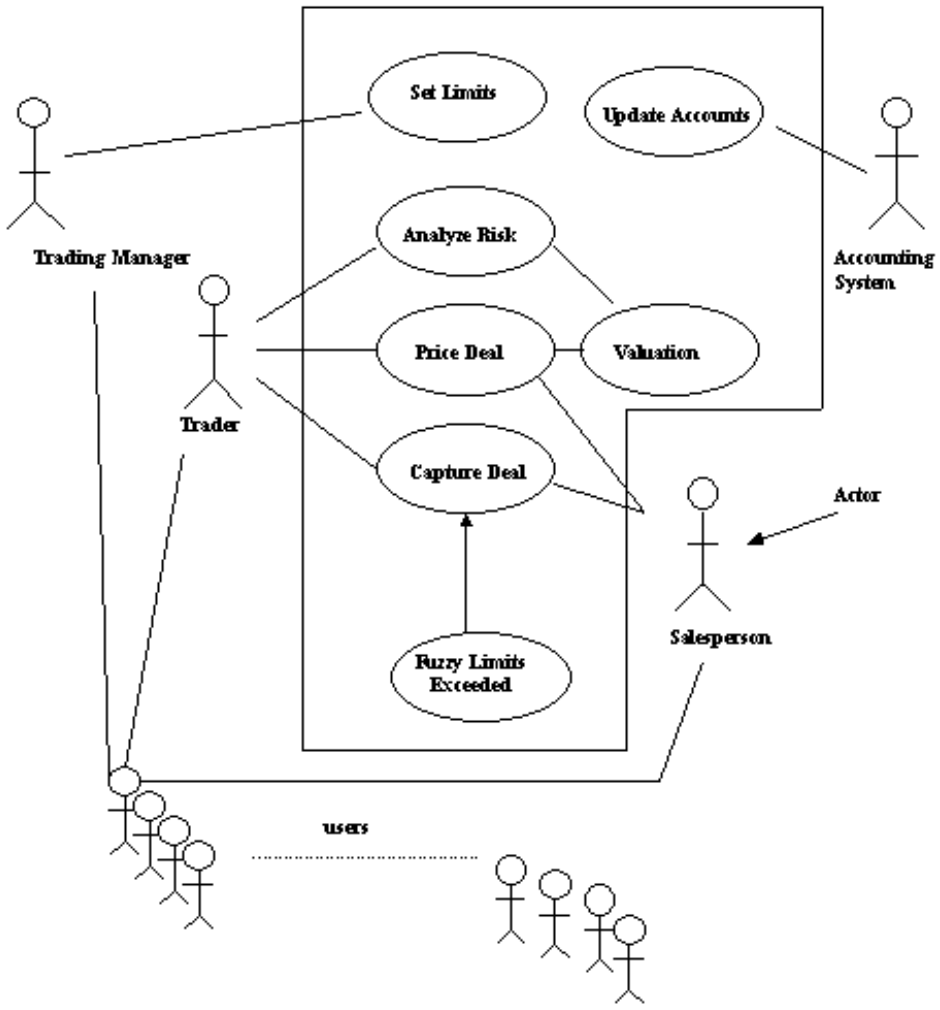

Figure 1 – Fuzzy Use Case Diagram

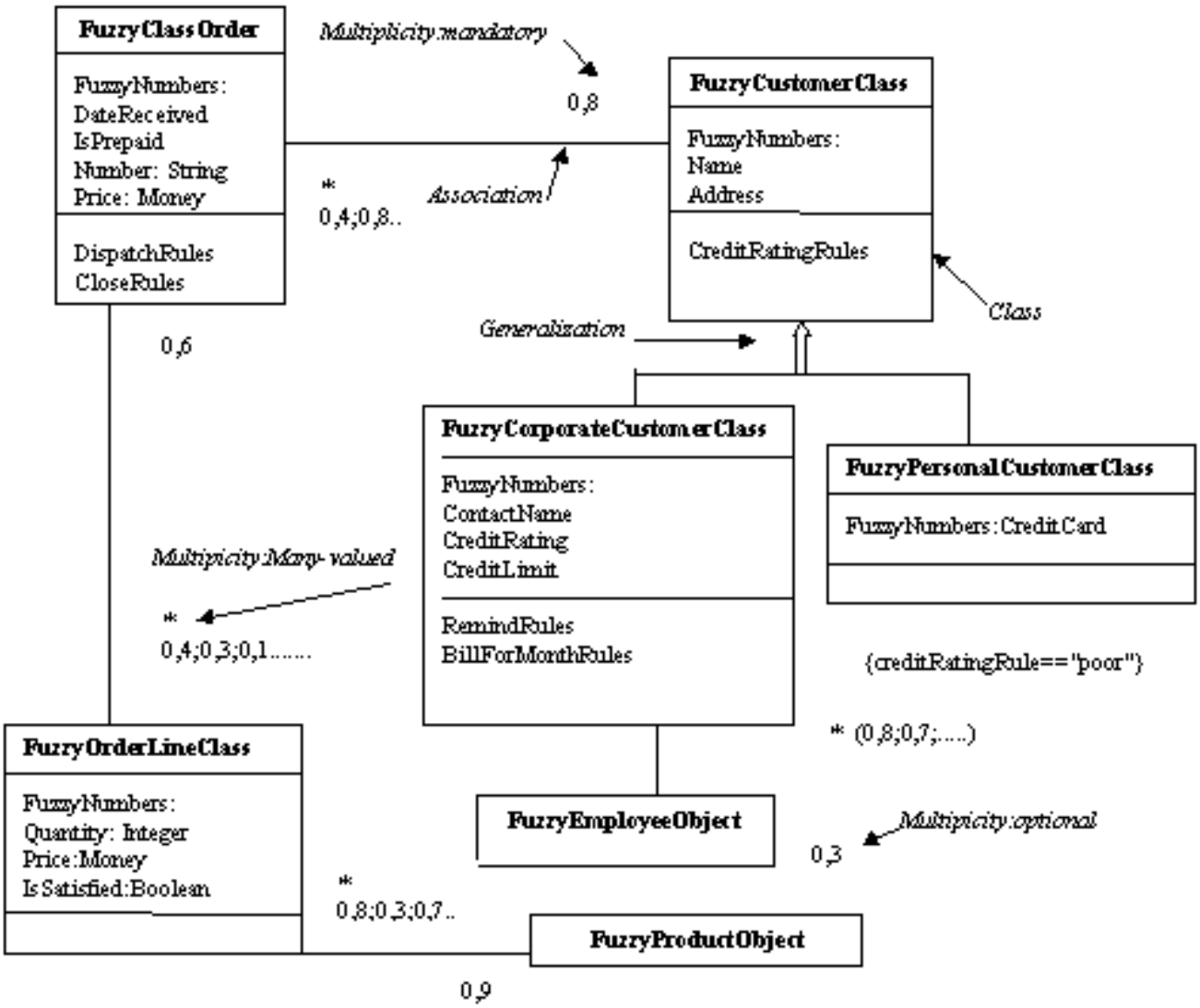

Figure 2 – Fuzzy Class Diagram

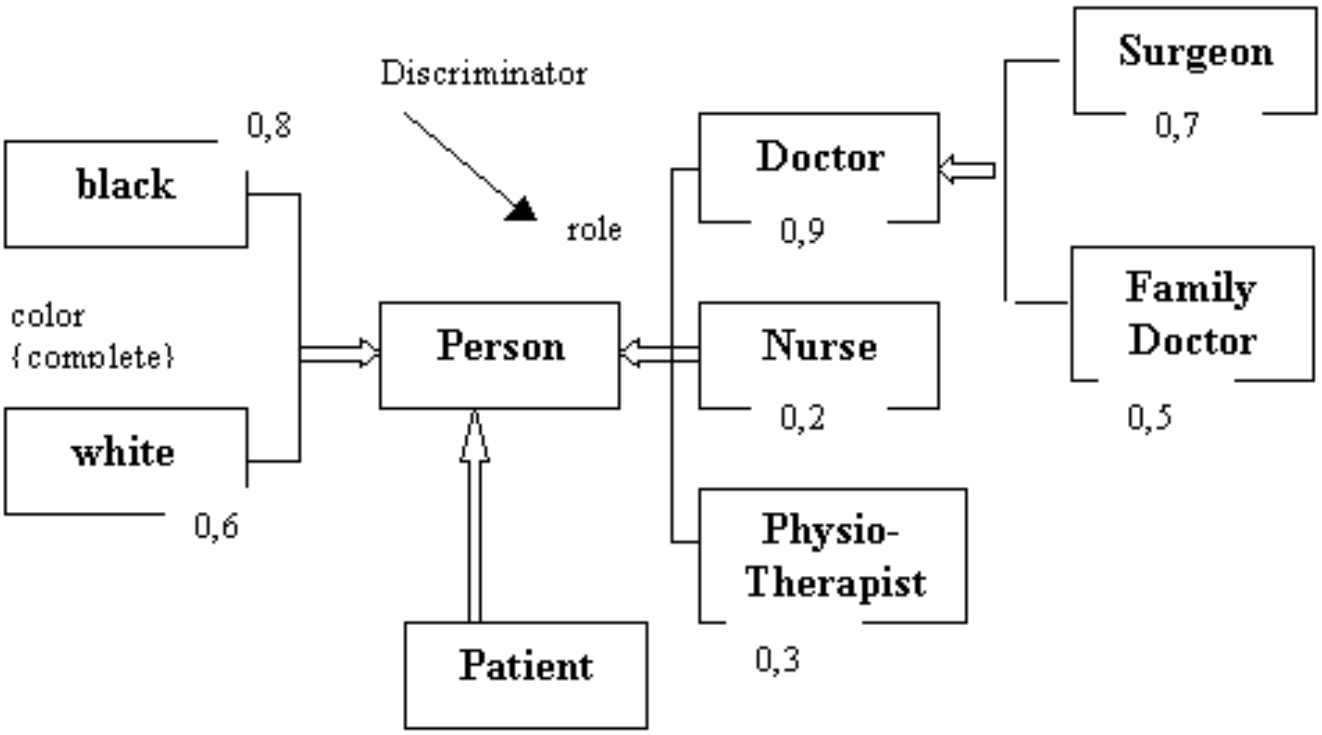

Figure 3 – Fuzzy Multiple Classification

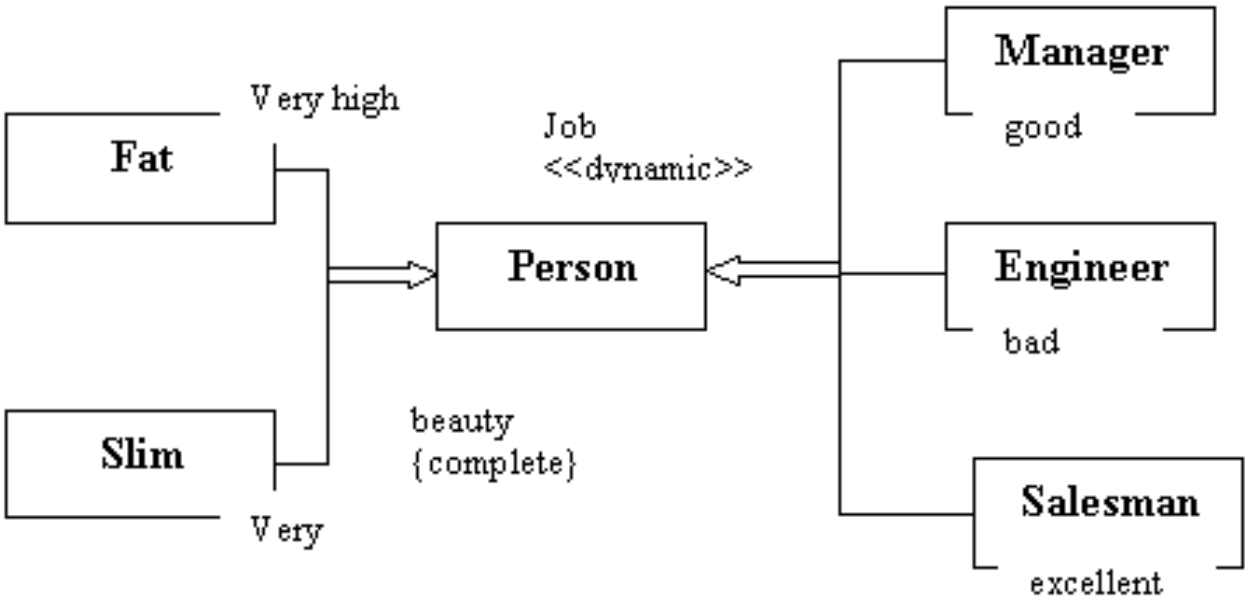

Figure 4 – Fuzzy Dinamic Classification

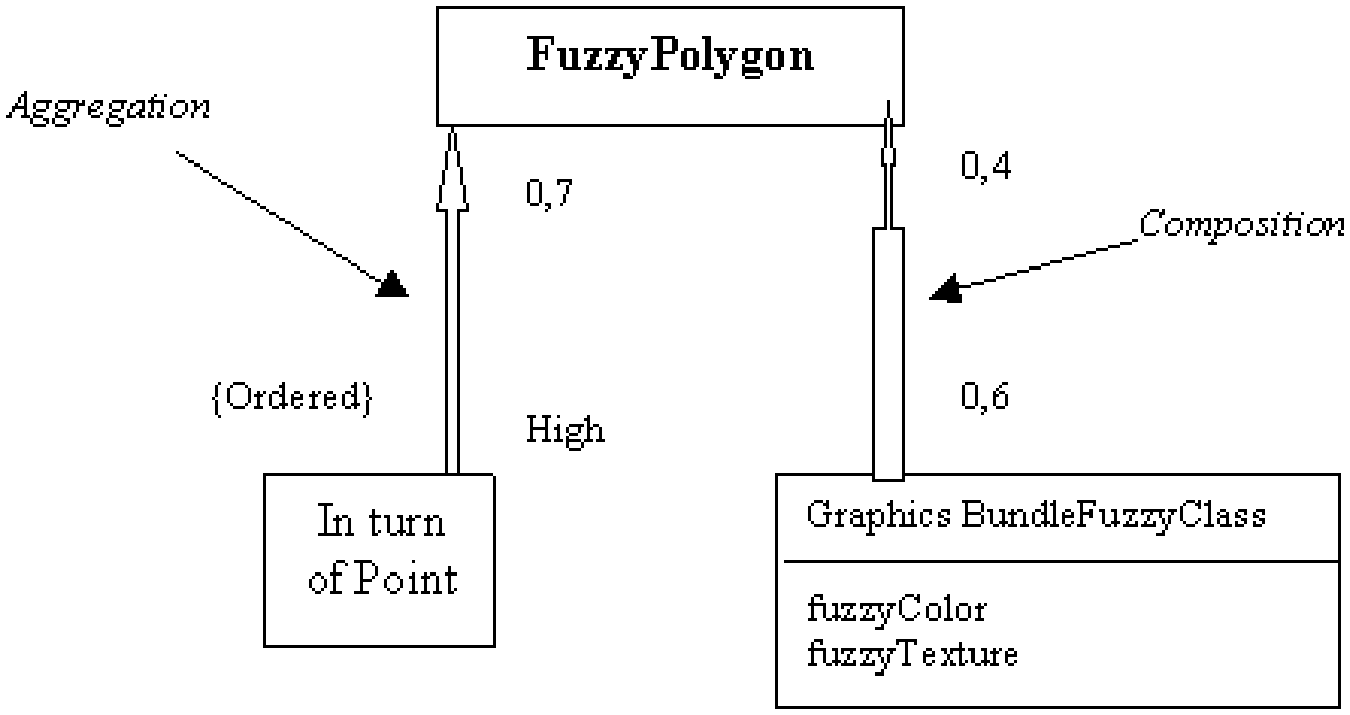

Figure 5 – Fuzzy Aggregation and Composition

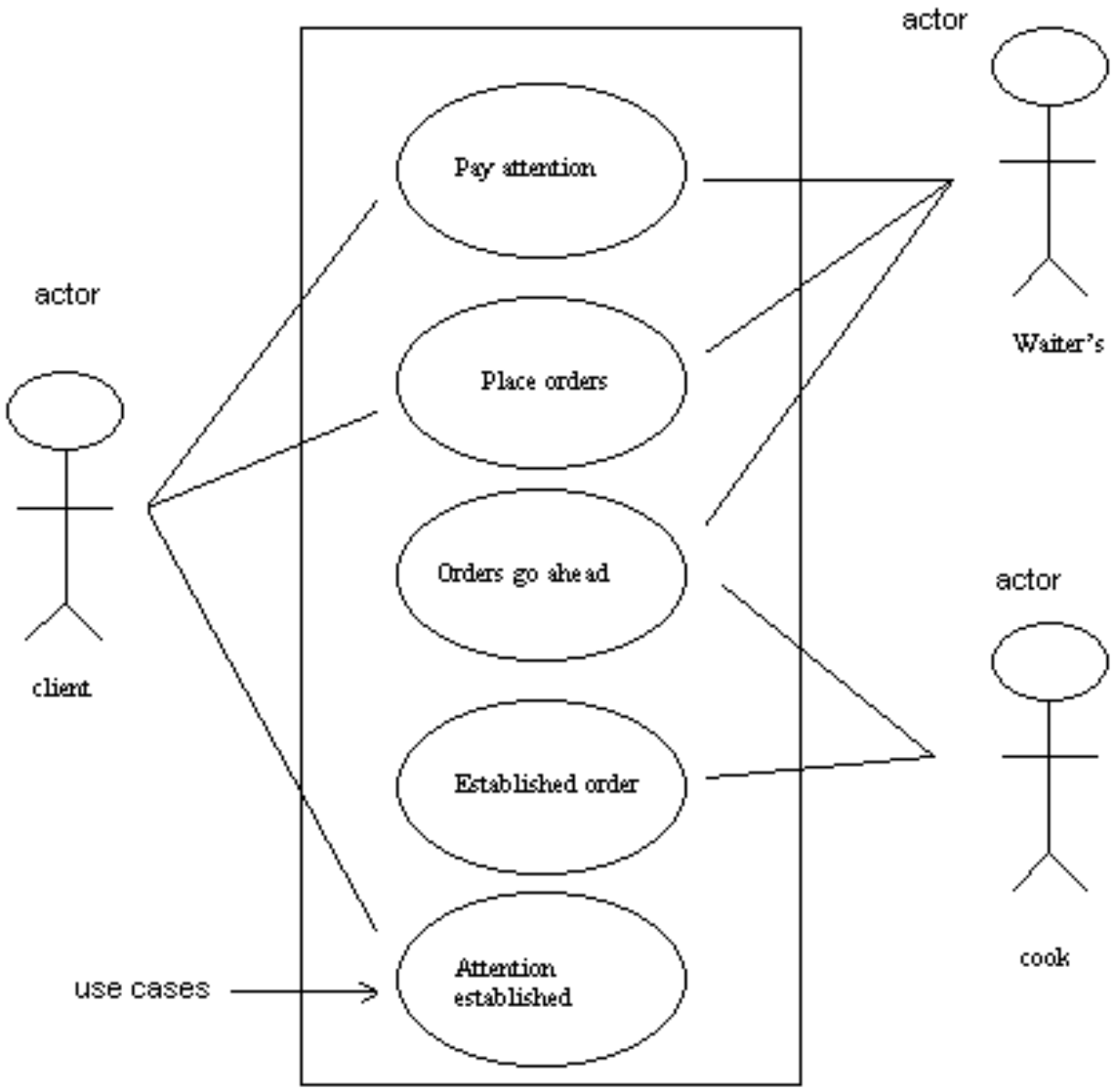

Figure 6 – Use case Diagram example

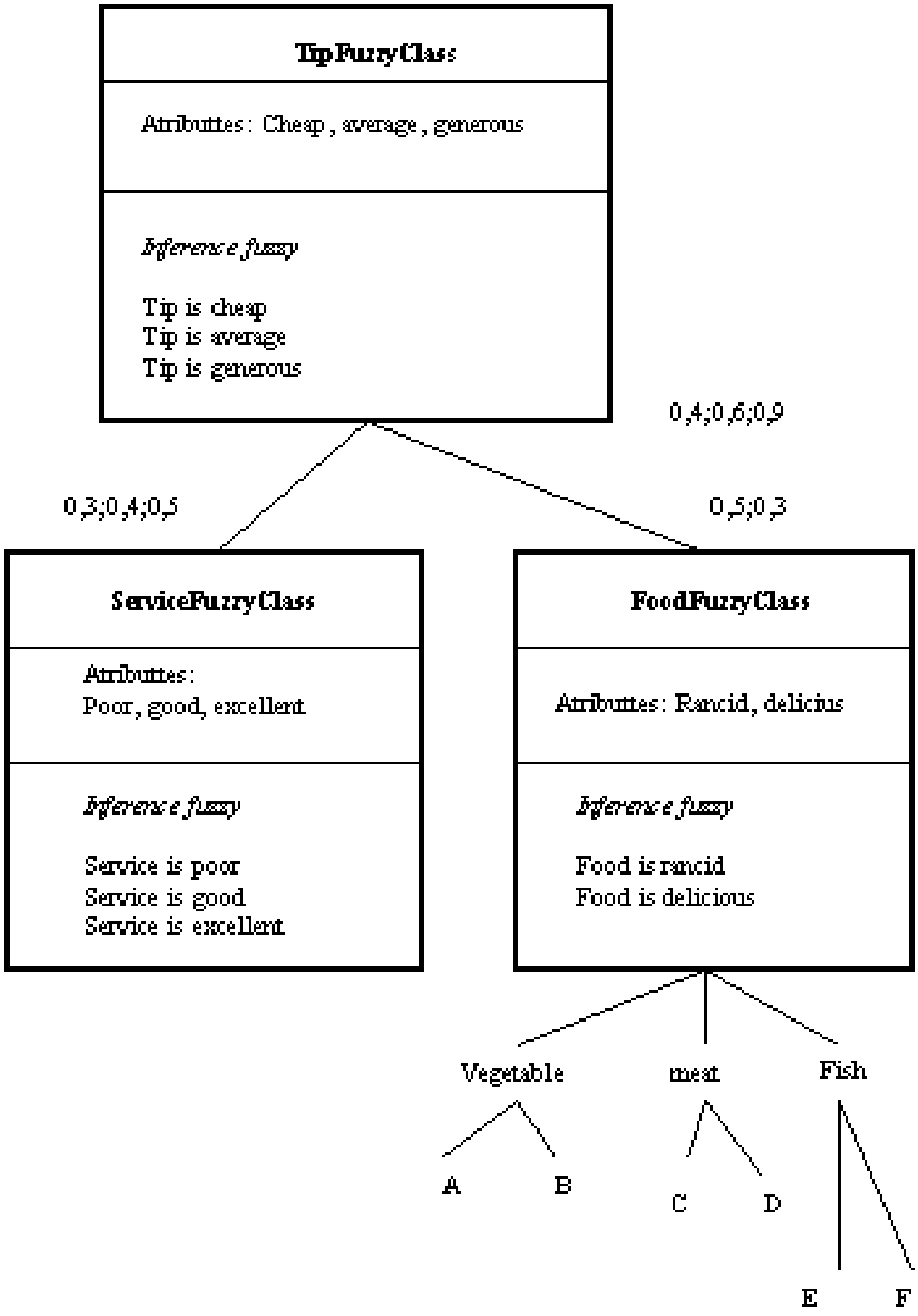

Figure 7 – Diagram of fuzzy class example

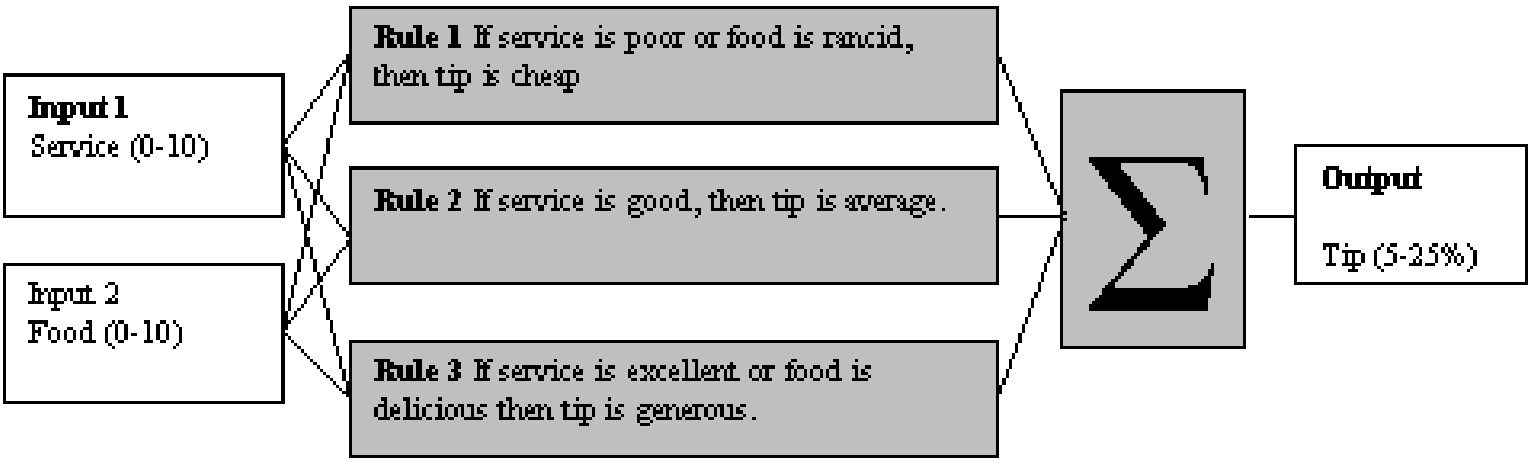

Figure 8 – Summary of If-Then rules

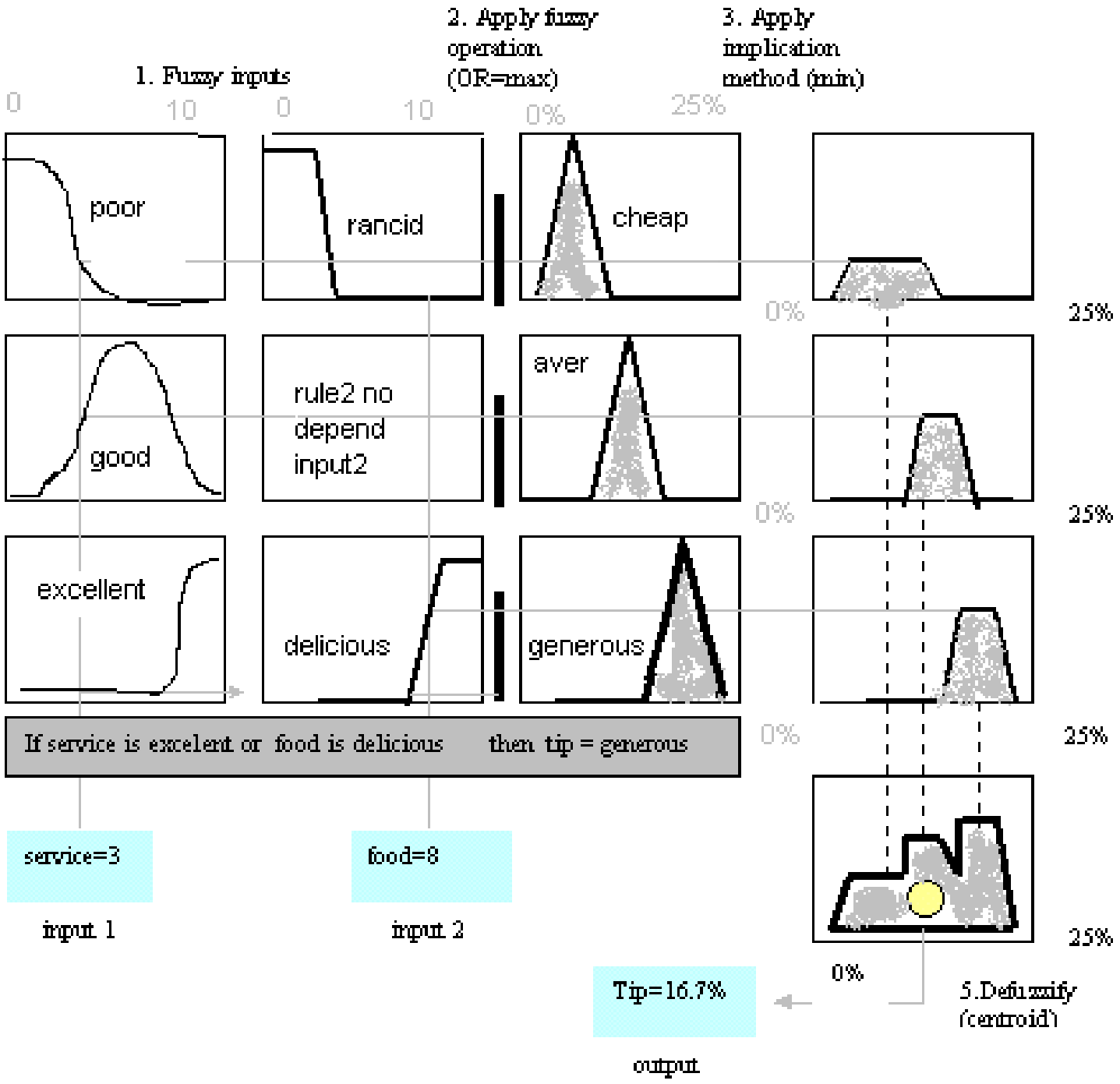

Figure 9 – The Real Full-Size Fuzzy Inference Diagram