Fuzzy Specification in Real Estate Market Decision Making

Victoria López, Matilde Santos
Faculty of Informatics, Department of Computer Architecture and Automatics
Complutense University, Madrid, 28040 Spain
vlopez@fdi.ucm.es, *msantos@dacya.ucm.es*

Javier Montero
Faculty of Mathematics, Department of Statistics and Operational Research
Complutense University, Madrid, 28040 Spain
monty@mat.ucm.es

Received: 14-10-2009
Accepted: 08-12-2009

Abstract

In this paper we present a software tool designed as a decision aid system for all actors being involved when buying or selling real state, client and realtor, where a main objective for the commercial is to concentrate the client preferences into few alternatives. Since the required previous analysis implies a number of fuzzy concepts, the general procedure here presented considers fuzzy logic to deal with specifications. As a consequence, time devoted to elicitation and requirement analysis is reduced.

Keywords: Fuzzy Logic, Fuzzy Requirements, Decision Making, Estates Valuation, Software Engineering

1. Introduction

The study and analysis of requirements in any kind of application is usually made by using natural language, i.e., in vague terms that are subject to uncertainty and therefore difficult to model. Elicitation and analysis of requirements demand a great amount of time and resources to interpret and model such vagueness. In particular, the process engineers facing vague specifications must generally make a great effort to understand and define a formal expression fitting the customer's preferences. An exhaustive analysis with the client is demanded in order to assure that there is no misinterpretation and to express these specifications in a computational way. But this exhaustive analysis of requirements is time consuming and increases the cost of the design process, delaying as a consequence key activities, such as implementation, validation, tests, etc., sometimes putting at risk the whole project.

The question that arises is: can we assure clarity in the requirements without specifying them in a crisp way? It should be noted that crisp specifications may look clear without fitting reality. Some requirements subject to uncertainty need to be specified as uncertain requirements. Moreover, some requirements must be completely specified before going through the next phase of design, but others can remain ambiguous at this stage. If meetings and negotiation processes with the customers about requirements can be reduced or even avoided, such a saving of time will be of great advantage for both parts. Furthermore, when the design is ready, the client and the analyst can focus on a better orientated task and they do not have to waste time going back to the first steps of the process.

In this paper we follow take into consideration the arguments given in [1,2] showing now an example on how fuzzy logic can help to implement vague preferences within the context of the real estate valuation. Before buying or selling real state, a previous valuation of each possible offer has to be developed by the commercial in order to concentrate the client decision making problem into few alternatives that may fit the client interest. But in this analysis quite a number of fuzzy concepts appear. In this work, we will propose a general procedure for handling vague requirements and uncertainty, considering fuzzy logic to deal with specifications, so we can reduce time in elicitation and requirement analysis.

In our particular application framework, this procedure will benefit not only the customers when they explicitly state their preferences, but also the complete process, as far as it makes easier the design of the required real estate valuation software. Our tool will represent a decision aid system for both parts: client and realtor. A real estate valuation is a critical issue in many countries and fuzzy logic can help to get an agreement between buyers and realtors in an easier, faster and friendly way.

The paper is organized as follows: in section 2, fuzzy requirements are defined and described. Section 3 presents the main parameters to be taken into account in real estate market. Section 4 shows how the fuzzy logic tool XFuzzy can be applied to solve this problem and the decision making procedure (showing the Spanish version of our software, the way it was presented to users). The paper ends with some final comments.

2. Dealing with fuzzy requirements

In this section, the design of a model that deals with uncertain specifications is presented. It is expressed in terms taken from the natural language of experts, both for the definition of the system and for the requirement elicitation. The methodology we propose is based on fuzzy logic. The inputs are colloquial explanations that are modeled by means of fuzzy sets. The fuzzy inference engine is applied so the fuzzy requirements produce clear information about the implementation of the preferences of the customers. This helps them to make a decision[3].

2.1. *Fuzzy requirements*

The requirements, according to the Institute of Electrical and Electronic Engineers (1994), are:

- a condition or capability needed by a user to solve a problem or to obtain an objective,
- a condition or capability that must be fulfilled or that a system must have, or specifications or other kind of documents imposed formally,
- a documented representation of a condition or capability like the previous ones.

Good requirements must be necessary, concise, avoid ambiguities, consistent, complete, attainable, verifiable and clear (see, e.g., Pfleeger[4], Pressman[5], Sommerville[6]). In order to get a formal specification of these requirements and their related functions and variables, some authors have introduced the fuzzy logic approach in formal specification methods[7]. For example, in Matthews *et al.*[8] the authors introduce language Z, a formal specification language widely used in crisp and more recently in fuzzy specification of requirements.

A Fuzzy Requirement will be then any condition or capability that is needed in a system to solve a problem or to reach an objective, which satisfies vague characteristics. These fuzzy requirements are based on fuzzy predicates, which are qualified linguistic terms that describe aspects of the requirements.

In Table 1, some examples of fuzzy requirements (related to the application that is presented in Section 3) and the possible corresponding fuzzy predicates are given. These terms will be represented by fuzzy sets that do not have precise borders and their characteristic function indicates a degree of fulfillment of that attribute[9]. In the first row we can identify two terms that can be assigned to fuzzy predicates. The first one, 'Grade', can simply be scaled between 0 and 1, or can be a predicate like good, acceptable, bad, etc. The second term contains predicates like north, south, northeast, etc. In the second row, the 'Space' can contain predicates like big, very big, small, etc.

Table 1: Fuzzy Requirements Examples

Fuzzy Requirement	Terms with possible Fuzzy Predicates
Natural light in an apartment	Grade (fuzzy) Orientation (linguistic)
Space	Grade (fuzzy) Number of rooms (crisp) Dimension of the rooms (fuzzy)
Garage	Is there? (Boolean) Size (fuzzy)
…	…

Usually, when customers who are thinking of buying an apartment are asked about the space or the luminosity of their future home, they do not know how to express their desires but in vague terms like 'medium room', or 'oriented to the Sun during the morning', for example. There are other requirements that are given by crisp or Boolean values and are easier to be described (they take values such as the number of rooms, garage, etc.)

In order to recognize a requirement that can be classified as fuzzy, we must consider the nature of the information that it handles. If the requirement depends on concise information to be validated, or it is a function that includes crisp values, is not a good candidate. On the other hand, requirements that show in their description some terms that could be assigned to vague characteristics, or features that can be graded, are good candidates to be fuzzy requirements.

2.2. *Fuzzy requirements in software development process*

As pointed out in López *et al.*[10], projects in software engineering in general require a lot of specifications, given in terms of documents, tables, data-flow models, and so on. The importance of defining specifications in a software development is well established. The introduction of fuzzy techniques may help to improve this stage into the software process and it is a must whenever software process deals with some kind of uncertainty[11].

The cycle life of a fuzzy requirement is the same of a standard specification. The difference is that the design engineer must identify the fuzzy predicates that define the terms of the requirement, and must be able to express the possible values for these predicates in fuzzy terms. It is at this stage of the software development process when the main benefits of the fuzzy methodology are obtained. A software engineer does not have to waste so much time with the customers. He or she just needs to write down the variables that represent the preferences of the client, and ask the user about a possible value for each specification. For example, the client may like a very luminous house. That is, first the process engineer has to identify the fuzzy requirements and then the fuzzy predicates related to that attribute given by the client. Usually, the customer does not express all the range of possible fuzzy specifications but only the positive ones. Sometimes it may be useful to ask the client for the requirements that he would not like at all for some of the characteristics (for example, a dark apartment or a cold one).

A fuzzy requirement is then described using less time than a crisp one, so later stages of the software development process can be reached faster, as soon as the engineer is not forced to establish crisp limits when implementing fuzzy requirements. Since a fuzzy requirement has more flexibility, there will be more possibilities to fulfill the specification of the client. In this way, software engineers only need to define a small family of predicates to describe the possible values of a linguistic variable. A fuzzy requirement will then allow us to focus specifically on the business, and not on numbers that may make difficult the task of requirements specification[12].

Once the fuzzy requirement has been elucidated, the software engineer must develop an analysis to verify its fuzziness and its correctness. If it is necessary, a requirement will be clarified following the traditional life cycle of the crisp requirements. Generally, the tasks of requirements elicitation and analysis are independent and they are even carried out by different people. But many projects integrate these stages so that verification can be suppressed. Once the requirement has been completely analyzed, validated, renegotiated with the client in order to get the final approval, and documented, we can then decide about the way these fuzzy requirements can be implemented (see figure 1).

If some of the functional specifications have been clearly identified as fuzzy predicates, the possibility of using a fuzzy engine or a business rules management system can be evaluated. At this point, it is necessary to make the last verification about the fuzziness of the information we are dealing with, and it may also be necessary to clarify some requirements. Finally, at the implementation phase and when the design of the requirements suggests a solution based on fuzzy logic, these requirements will be implemented using a Fuzzy Inference Engine. Fuzzy Inference Engines are mathematical tools that are oriented to implement fuzzy rules. They are well known in fields such as electronic and control, but less known in commercial areas. They can be implemented by standard tools (Matlab©) or by free software packages like XFuzzy[13].

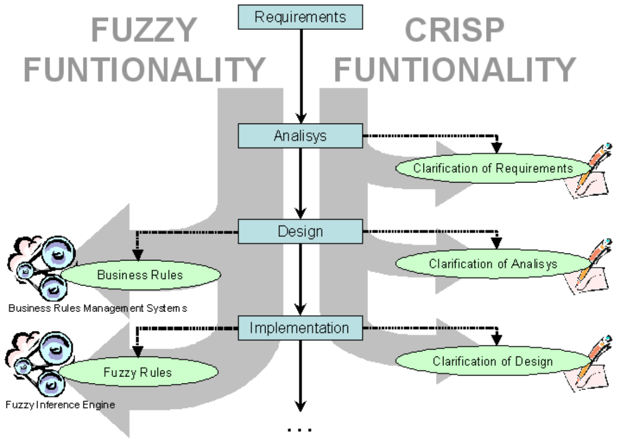


Fig. 1: Methods for fuzzy requirements implementation

On the other hand, a Business Rules Management System is a powerful, robust and expensive software, where the functionalities are modeled by rules in a natural language, using terms of business but limited to specific amounts, which can be modified at any time, even in the production stages. They are known as complementary tools for Business Intelligence, Business Processes Management Systems and Integrators[6].

The selection of one of these inference engines depends on the architecture of the application and on the possibility of handling the fuzzy predicates. If the fuzzy inference engine is going to be used, it will be necessary to limit the predicates at the design stage in order to guarantee the coherence of terms and modifiers. It is necessary to identify and set the limit values of the predicates, that is, the universe of discourse of every fuzzy set. On the contrary, if the Business Rules Management System is chosen, the identification of these values can wait until the production stage (with these tools the limits can be included at any time of the process).

2.3. *Benefits of Using Fuzzy Requirements*

According to Noppen *et al.*[14], one of the main problems in software design is that, in order to model the requirements, it is necessary to have a set of clear requirements, without ambiguities. But quite often, real requirements contain conflicts, are vague in their nature and change with time. Many projects fail because the set of requirements presents this kind of problems and conflicts but they have not been accordingly handled.

The use of fuzzy logic to represent the specifications has some advantages that can be summarized as:

- Time saving in the elicitation of requirements, meaning less time in the negotiation and a faster implementation of the system, and allowing therefore an earlier validation,
- The clarification is postponed until later stages, only when it is strictly necessary,
- From a business point of view, the information is clearer as it is closer to the software engineering language, in such a way that the analysts do not need to consult the customers so often.
- From a client point of view, the customers do not have to make such an effort in order to define accurately the characteristics they are looking for. Each client can express them in vague and ambiguous terms

In the next section, we shall use a Fuzzy Inference Engine to address the estimation of fuzzy preferences of customers in a real estate valuation.

3. Valuation and decision making in the real estate market

Real estate valuations have been a critical issue in many countries. Investors, families and realtors have made a lot of formal and informal valuations. The final price of an estate depends on different factors and sometimes the same product is sold with a very different price. If the markets are running well and banks loan money without problems, investors feel confident about real estate market and buy without paying too much attention to valuations. But the present crisis of real estate market is a fact difficult to analyze. As it happens with other products, sales are influenced by many factors of difficult control by governments. The factors that move the global economy into crisis are sometimes more related to investors than to family behavior. But lately, many people in several countries have invested their money in real estate. Then, if prices are high, mortgages get more expensive and banks get into liquidity problems. As a consequence, families see restricted the amount for their loans. It may be even the case that professional investors lost interest in real estate as an investment product.

Anyway, buyers and renters make their decisions according to a set of preferences, based upon an analysis that in some way pursues to determine which real estate better fits those preferences. First of all, a general expert (named expert 1) elaborates a real estate form for describing every relevant characteristics of each real estate (size, rooms, views, etc.) and a basic classification of all potential clients (buyers and renters). Such a real estate form covers the most frequent criteria and characteristics relevant for client's decisions. In our case, data are stored into a data base (Microsoft Access or SQL), which will be ready to accept queries from the real estate valuation tool.

We can find some software tools dedicated to analyze the preferences of the buyer of a house. Among this type of tools, finders and recommendation systems are the most popular. These tools perform selective searches in their data bases. The results of the search usually show pictures and videos that allow the buyer to throw a look to the product before visiting it or deciding their purchase. The main problem is to carry out a suitable and customized selection for each client. Very often, the supply is so extensive that the client finds difficulties to make a decision, which turns the system inefficient.

On the other hand, the prices are fixed beforehand independently of the interest of the buyer, even overvaluing the product. This paper will also show that the sale price can be adapted to each client “a priori”, according to her or his personal characteristics.

There is usually a great difference between the sale price of a house and its legal valuation. For example in Spain, the *Sociedad de Tasación*[15] makes a valuation based on about 15 properties in the same area that have been valued in the last 6 months. The information about the property to be valuated is very simple: area in the city, type (detached or terraced house), squared meters, age, condition (unimproved, partial improvement, total rehabilitation), quality in comparison with surroundings (luxury, high, medium, and low) and other features (pool, garage, top floor, interior flat, lumber room). Therefore the resulting value is only for guidance and of course, it has no legal validity although the result is used as mortgage value. For this reason, the mortgage value agrees with the value of appraisal. Nevertheless, experts and realtors affirm that the value of appraisal does not agree with the value of sale of a real estate property. They say people do not mind about the value of appraisal of their estate purchases. They only need this data to get a mortgage. In fact, value of appraisal is usually very different to the sale price.

At this point, some questions arise: how is the value of appraisal calculated? And, why is it not important for the customer? According to the information that an expert in the real estate sector has provided us, the initial price usually pays attention to the proprietor and to the rise. The sale price is different because usually there is a negotiation between the client and the salesman and it is only the reference value at which the process starts.

Starting off from the idea of the appraisal value, we propose to elaborate a new value (as initial price) that takes into account a calculation including fuzzy logic to obtain a better approach to the real value of sale. The idea is simple:

$$NewGlobal = \max_{i=1..5} NewValue_i \tag{1}$$

$$NewValue_i = ValueofAppraisal * (\alpha + CF_i) \tag{2}$$

where CF_{i} is the correction factor calculated by the software tool proposed in López *et al.*[16]. The value of this factor is within the interval [0,1] and represents the preferential value of a specific class about a real estate property. The value $\alpha \in [0,1]$ is fixed by an expert in markets (nowadays in Madrid, Spain, we have considered $\alpha=0$).

In order to clarify both formulas above, let us present an example. Let Value of Appraisal = 300,000 €, $\alpha=0$, and $CF_{\text{i}}=0.95$. First two values are given by the help of expert 1, who knows the estate and the actual market situation well. The value $\alpha=0$ means that the estate market tends to low prices. On the other hand, CF_{i} is the value that our software tool gives as a perception between the profile ‘i’ (specific class of the population) and the estate property. This value is obtained by using the help of an expert 2 in defining and classifying population into 5 clusters (see section 4). $CF_{\text{i}}=0.95$ means the profile ‘i’ have a good value of the property, i.e., it fits very well with the personal preferences of the client. Hence,

$$NewValue_i = 300{,}000\ € * 0.95 = 285{,}000\ €$$

3.1. *A fuzzy decision problem*

Being the main goal of our application the estimation of the valuation of a real estate, we must acknowledge that this problem is highly dependent on the perception that the buyer or the renter has about the real estate under consideration. It is therefore necessary to obtain all data of interest, determine the preferences of the customer, and develop a fuzzy system for evaluating the estate regarding to these preferences.

First of all, it is necessary to classify the customers. Classification of potential clients implies a cluster matching with a description of the ideal real estate for such a type of client. According to the general expert 2, we should be considering five basic clusters:

φ_1 young couple;
φ_2 2-3 children family;
φ_3 single male;
φ_4 single female;
φ_5 investors

Since investment is very sensible to many factors, in this paper we limit the analysis to the other four first clusters. All experts we have contacted agree with this classification and they accept that they would be able to classify most clients into one of these groups. When someone does not belong to any of these groups, we refuse the client as a non-typical one.

Descriptions of the clusters are done by means of fuzzy first order logic.

3.2. *Fuzzy preferences*

On the other hand, characteristics (fuzzy and crisp) are going to be classified into two different groups according to the profile of the client: primary and secondary features. Primary characteristics are defined as a proposition in natural language and they are associated to each client profile, for example:

φ_1: 2 bedrooms, downtown, garage and middle price
φ_2: at least 3 bedrooms, garage, garden and pool, good light, in the suburbs area.
φ_3: small apartment (1 bedroom), garage, downtown, middle price
φ_4: 2 bedrooms, good light, elevator, garage if possible, below middle price.

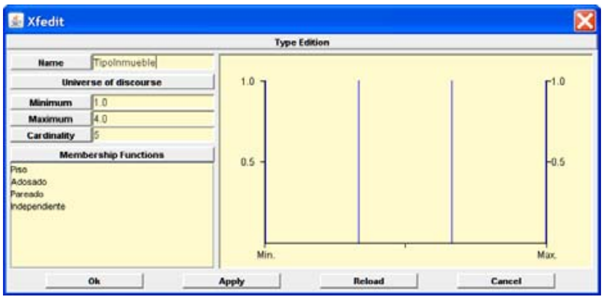
As any other decision making problem, not all characteristics or properties about a real estate are relevant in a first approach. In this paper, we consider a set of 20 basic characteristics that we first divide into two, depending on their crisp or fuzzy nature. The subset of crisp characteristics contains numerical description like the number of rooms, pool, garage, and so on. Fuzzy characteristics are qualitative and not easily associated to an integer number (nice views, good light, comfort, and so on).

Our general expert defines not only both subsets of characteristics but also the form in which all this data will be taken. For example, about lift, it is enough to assign 'yes' or 1 if it exists or 'no' or 0 otherwise. However, a fuzzy characteristic like ‘luminosity’ must be valuated by an expert (someone who has a good knowledge about the flat or apartment).

The value assigned to a fuzzy characteristic has to be chosen between 0 (the worst) and 10 (the best). Later on a fuzzy set will also be used to define this type of characteristics. Of course these data can be given by a unique expert, but in this particular work, data were given by two different experts due to their different goals: Expert 1 helps to formalize characteristics and estimated values in relation to the estate property within the market. Expert 2 helps to classify the population according to their preferences and necessities.

Once the potential buyers have been classified (by expert 2), the next step is then to describe the data according to the corresponding types. As it is said before we have considered two types of data: singletons (crisp data, including Boolean) and fuzzy data.

For example, Figure 2 shows the crisp variable, ‘type of house’. The four possible crisp values for this variable are: ‘flat’, ‘detached’, ‘semidetached’ and ‘terrace’. Fuzzy data are represented by means of fuzzy sets as linguistic variables. They correspond to the Mamdani singleton centered in 1, 2, 3 and 4 respectively, where 1 corresponds to ‘min’ in figure 2 and 4 to ‘max’.



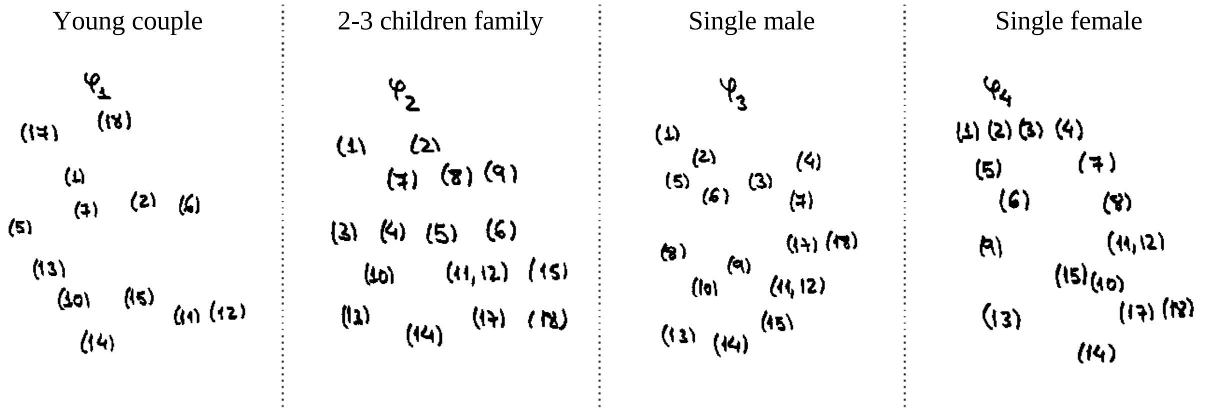
Name: TipoInmueble
Minimum: 1.0
Maximum: 4.0
Cardinality: 5

Piso
Adosado
Pareado
Independiente

Fig. 2: Variable ‘type of house’

Table 2: Set of preferences

		φ1			φ2			φ3			φ4		
	Characteristic	P	S	T	P	S	T	P	S	T	P	S	T
1	Type of house	2	1	2	2	2	2	2	0	2	2	2	2
2	Light	0	1	2	2	2	2	0	2	2	2	2	2
3	House face	0	0	0	0	2	0	0	2	0	0	2	0
4	Representative	0	0	2	2	2	2	0	2	2	2	1	2
5	Doorway	0	1	2	2	2	2	0	1	2	0	2	2
6	Facade	0	1	2	2	2	2	0	1	2	0	2	2
7	Views	0	2	2	1	2	2	0	2	2	2	2	2
8	Number of rooms	0	0	1	2	2	1	0	1	1	2	2	1
9	Number of bathrooms	0	1	2	2	2	2	0	0	2	0	2	2
10	Age	0	0	2	1	2	2	0	2	2	0	2	2
11	Floor	0	1	2	0	2	2	0	2	2	0	2	2
12	Elevator	0	1	2	1	2	2	1	2	2	1	2	2
13	…	0	1	2	1	1	2	0	1	2	0	2	2


Young couple

2-3 children family

Single male

Single female

Fig. 3: Preference diagrams

The profile of a class (population sector) is described as the conjunction of their preferences about data. Also the valuation resulting is defined as a fuzzy data and then it is described by a granulation. The set of preferences is evaluated as specification of requirements.

The set of preferences defined by the customers can be seen in Table 2. These are the primary (P) and secondary (S) characteristics that each class of client (φi) would like to find. A tri-valued logic was applied as they had to choose between 0-2 to give a mark to each feature. The column T means the evaluation of the assessor.

The expert 2 describes the characteristics in another way: by diagrams. Expert 2 knows by experience what a potential client would agree and they will get them into it. Preferences diagrams are used to determine the degree each client assigns to each characteristic. These diagrams are developed by the expert 2 according to the environment and each client profile, reminding Hasse diagrams as a pre-order relation between criteria. Each client profile is characterized by a diagram by means of an easy and intuitive information model close to natural language without losing too much information (Figure 3).

Finally, it must be pointed out that sorting preferences is in general a difficult task, even if the number of items is low, but it may become critical when dealing with fuzzy criteria.

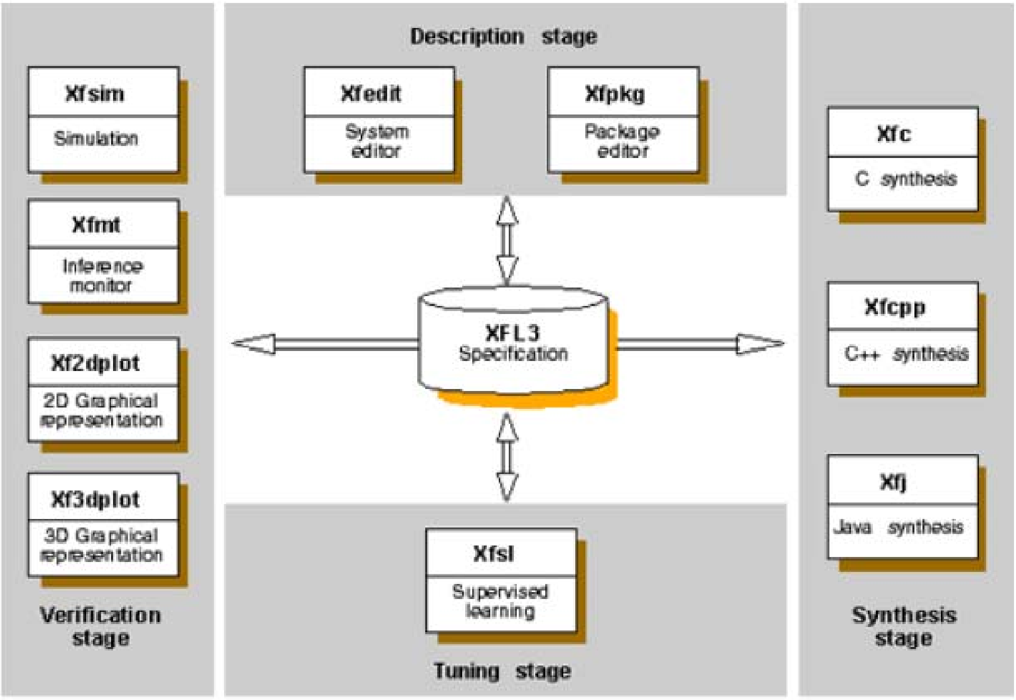


Fig. 4: Overview of XFuzzy

4. Fuzzy implementation of the decision making system

Evaluation of the specifications is done by means of a functional tool for fuzzy first order logic evaluation called XFuzzy[13], created by the Spanish National Centre of Microtechnology. It is a platform for developing inference systems based in fuzzy logic. Xfuzzy has been programmed in Java and has GNU license. Some of its main characteristics are its great flexibility to extend the set of available fuzzy functions and, because of that, the capacity for developing complex systems. XFuzzy allows an automatic code generation in different programming languages (Java, C++, C), to be used in any inference.

Evaluation of the specifications consists of several tools that cover the different stages in the design process of a fuzzy inference system (Figure 4).

- The *description* module includes graphical tools for the definition of the fuzzy system and the functions library (inference base rules and membership functions),
- The *verification* unit is composed by tools for simulation, debugging and graphical visualization of the system behaviour.
- The *tuning* module facilitates the application of learning algorithms in order to improve the system.
- The *synthesis* unit includes tools to generate code in high-level languages (C, C++ and Java), for software and hardware implementations of the designed fuzzy inference system.

The different tools of Xfuzzy can be executed independently. The simulation environment integrates them under a graphical interface that facilitates the design procedure to the users. In the main screen of Xfuzzy, these tools are divided into those modules.

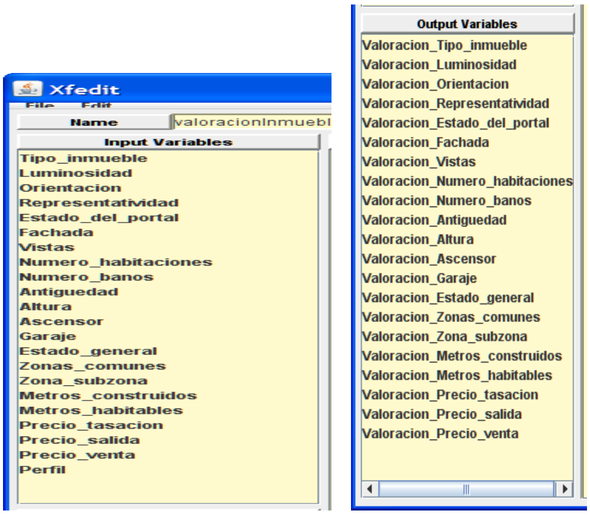
Xfuzzy uses the specific language XFL (XFuzzy Language). It is a flexible and powerful language that allows, not only to express complex relations between fuzzy variables, but also to use hierarchical rule bases and connectives, linguistic modifiers, membership functions and different methods for defuzzyfication. It also serves as a link between all the tools.

In this paper we have chosen a Java implementation in order to link it with a valuation program we developed in Java [10] as a GUI for this application.

Following the above steps, we have generated a fuzzy system that returns the valuation of a real estate according to the client preferences. In our model, the system uses as input data nine primary parameters that come from the union of all primary characteristics in all profiles: type of estate (flat, apartment, villa, etc.), area, number of rooms, good light, lift, garage, housing development, community features and cost (see Table 2). They are defined as linguistic variables in the fuzzy inference system (Figure 5 left).

Then the system generates several output variables according to the valuation of the real estate with respect to each profile and the different specifications that fulfils every variable (Figure 5). Variables of figure 5 will be modelled as fuzzy sets.

Notice that some variable names are given in Spanish language because this software tool is being tested with clients in Spain (we are showing screens the way they are presented to users).



Name: valoracionInmuebl

Input Variables

Tipo_inmueble
Luminosidad
Orientacion
Representatividad
Estado_del_portal
Fachada
Vistas
Numero_habitaciones
Numero_banos
Antiguedad
Altura
Ascensor
Garaje
Estado_general
Zonas_comunes
Zona_subzona
Metros_construidos
Metros_habitables
Precio_tasacion
Precio_salida
Precio_venta
Perfil

Output Variables

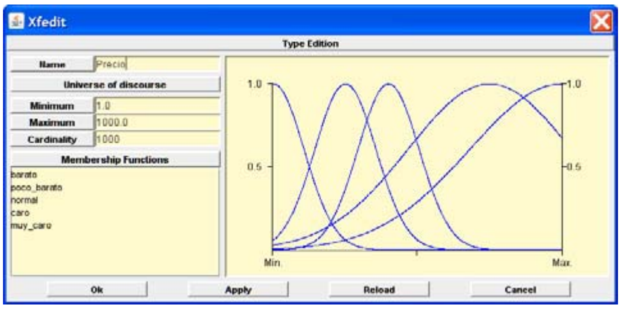
Valoracion_Tipo_inmueble
Valoracion_Luminosidad
Valoracion_Orientacion
Valoracion_Representatividad
Valoracion_Estado_del_portal
Valoracion_Fachada
Valoracion_Vistas
Valoracion_Numero_habitaciones
Valoracion_Numero_banos
Valoracion_Antiguedad
Valoracion_Altura
Valoracion_Ascensor
Valoracion_Garaje
Valoracion_Estado_general
Valoracion_Zonas_comunes
Valoracion_Zona_subzona
Valoracion_Metros_construidos
Valoracion_Metros_habitables
Valoracion_Precio_tasacion
Valoracion_Precio_salida
Valoracion_Precio_venta

Fig. 5: Input (left) and output (right) variables in Xfuzzy

In a first stage, Xfuzzy requires a fuzzyfication of the input variables by defining membership functions. Such a fuzzyfication allows a faster computation in a fuzzy framework.

A membership function is then related to an input variable in order to point out the degree to which this characteristic matches certain property. Next figures show some examples of membership functions.

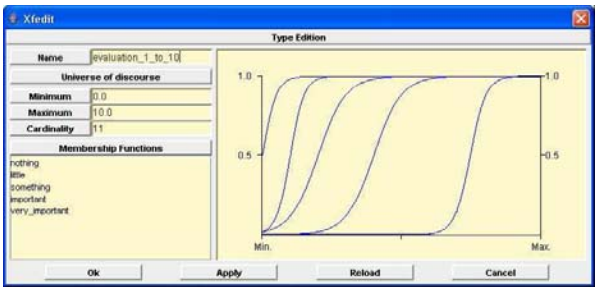
Figure 6 shows the variable ‘initial price’. Five fuzzy sets: 'cheap', ‘less cheap’, ‘medium’, ‘expensive’, ‘very expensive’ have been assigned to this specification. The fuzzy sets are defined by sigmoidal membership functions.



Name: Precio
Universe of discourse
Minimum: 1.0
Maximum: 1000.0
Cardinality: 1000
Membership Functions
barato
poco_barato
normal
caro
muy_caro

Fig. 6: Variable ‘initial_price’ and its corresponding fuzzy sets

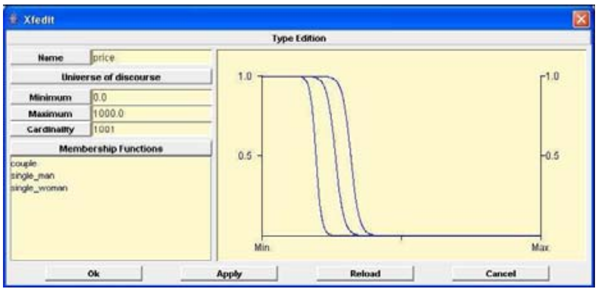
Figure 7 shows the variable “evaluation” taking values between 0 and 10 (for the clients, according to the importance they give to those characteristics). For example, we may choose between labels like 'nothing', 'little important', 'something important', 'important', 'very important', in order to represent the degree of fulfilment for each client of each variable.



Name: evaluation_1_to_10
Universe of discourse
Minimum: 0.0
Maximum: 10.0
Cardinality: 11
Membership Functions
nothing
little
something
important
very_important

Fig. 7: Variable evaluation with values between 0 and 10

Variable ‘Price’ (Figure 8) has only three labels, one per each profile (except the 2-3 children profile, which does not show the characteristic ‘price’ as a primary characteristic). We can see in this figure how some fuzzy preferences are assigned for each class of customer.



Name price

Minimum 0.0

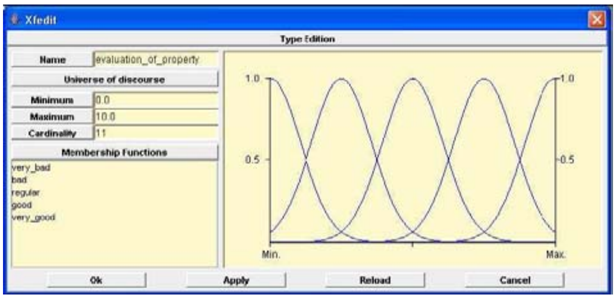
Maximum 1000.0

Cardinality 1001

couple
single_man
single_woman

Fig. 8: Output variable price in Xfuzzy

Figure 9 shows the membership functions for the final output variable ‘valuation_of_property’. This time it is defined by means of five labels: 'very bad', 'bad', 'regular', 'good' and 'very good'.



Name evaluation_of_property

Minimum 0.0

Maximum 10.0

Cardinality 11

very_bad
bad
regular
good
very_good

Fig. 9: Fuzzy sets for the Valuation_of_property

The valuations are given by a value in the range of [0,1] which can be used as a correction factor for the price of appraisal in (2) following two different ways:

(i) As information for the salesman, classifying the estates from largest to smallest degree

(ii) As a new initial value calculated by applying the correction factor to the value of appraisal, as it is shown in the previous section, in (2).

As it is possible to see in this section, there are different ways to show the fuzzy information about either the customers or their preferences. Depending on the variable we are relating to, different conclusions can be obtained.

Hence, the set of fuzzy inference rules will be defined taking into account preferences for each profile in the following sense: let P = {C_1, ..., C_k} be the set of criteria that fits with the profile of the client. If an apartment satisfies all criteria then this apartment must be valuated as 'very good' for this profile. If the apartment does not satisfy any one, then it will be 'very bad' valuated. If it satisfies a few criteria, then it will be valuated as 'bad', and so on. For example, one of the rules is:

If ‘luminosity’ is ‘very bad’ and ‘profile’ is ‘single_male’ then ‘luminosity_valuation’ is ‘bad’ as this feature is not so important for this type of client.

All these rules are edited by the designer in XFuzzy 3.0. The rules that are related to the characteristic ‘luminosity’ for the different profiles are shown in Figure 10. The consequence of the rules is the valuation of this requirement for each customer profile expressed in fuzzy terms.

The behaviour of the fuzzy inference system depends on the mathematical operators that are selected: the connectives of the rules, the operators which implement the fuzzy implication and the defuzzyfication method. Initially the product xfl.prod() has been assigned to the operator "and". The default implication in Xfuzzy is Larsen’s, and the defuzzyfication method is given by the function xfl.FuzzyMean().

The last part consists of the implementation of the application core, the graphic interface, and the connections between the application, the inference engine and the database. The graphic interface is appointed to interact with the user, and it allows the access to the information stored in the database connected to the system.

The system has a database of buildings (MS Access) and every line fulfils the conjunction of requirements for each profile, with an appraisal according to a degree given by a value belonging to the interval [0,1]. All data about real estate properties are stored also in this database and the system uses it to be able to manage the information required from it.

Database normalization technique is used to avoid duplication of information. A Third Normal Form (3NF) of degree of strictness has been applied in the design. Each flat or apartment consists of up to 31 fields: 5 are of text type, 13 are numerical, 7 Boolean and 3 monetary ones.

The fuzzy inference engine is used to produce the results. The application has been developed using Eclipse (platform for software development primarily written in Java) and JDK 1.6. All fuzzy matrix labels are pre-developed in Java with Xfuzzy, latterly adapted and adjusted to the system architecture and specific requirements.

We have also developed a graphic user interface (GUI) for this application that is oriented to interact with the customer. It allows the user to enter the required input data and transmits it to the Valuation of the States module.

When all the calculations have finished, the fuzzy valuation tool gives the results by applying the inference rules with the fuzzy specification of the profiles, therefore obtaining a fuzzy valuation as a result (see Figure 11).

Once the system is defined, it is necessary to export it to a programming language for its simulation. Using the Synthesis tool, the system is exported to Java (BEA System) generating four classes:

- ***FuzzyInferenceEngine.java***: it is the file that describes the inference method applied by the fuzzy system.
  The application provides a graphic interface appointed to interact with the user.
  Once the information stored in the database connected to the system is accessed, a fuzzy inference engine is used to show an approximated calculation of the valuation of the estates for the selected profile.
- ***MembershipFunction.java***: it is the file that contains the description of the interface to describe a fuzzy number.
  It only admits a method called ‘compute’ that calculates the membership degree of each value of the input variables.
- ***FuzzySingleton.java***: this file is a class that implements the MembershipFunction interface, and it represents a crisp value as a fuzzy number (fuzzyfication).
- ***EstateValuation.java***: it is the file that contains the class that describes the complete fuzzy system. This class is a particular implementation of the FuzzyInferenceEngine interface.

Therefore, the public methods that implement the inference in this system are crispInference and FuzzyInference.

It is possible to simulate the system using the exported Java classes. The BEA Weblogic server and its IDE BEA Workshop will be used to implement a pageFlow (Servlet for the BEA technology), that performs the simulation. However, as the fuzzy inference system is exported as a Java class, it can be simulated with any other visualization method: Servlets, Swing, AWT, Midlets, etc.

When having installed the server BEA Weblogic 8.1, it is necessary to create a domain to lodge the project that is going to be created. Using the default functions of the Configuration Wizard of BEA, a simple domain with all the necessary characteristics can be generated to execute the example.

The project must be created at this instance of the server as a Web Application, where the pageFlow and the visualization page are in a Web Project, and a Control Project has to be added with the classes generated by Xfuzzy. Finally, the Control Project must be compiled and then it runs on the server.

5. Final Comments and Future Work

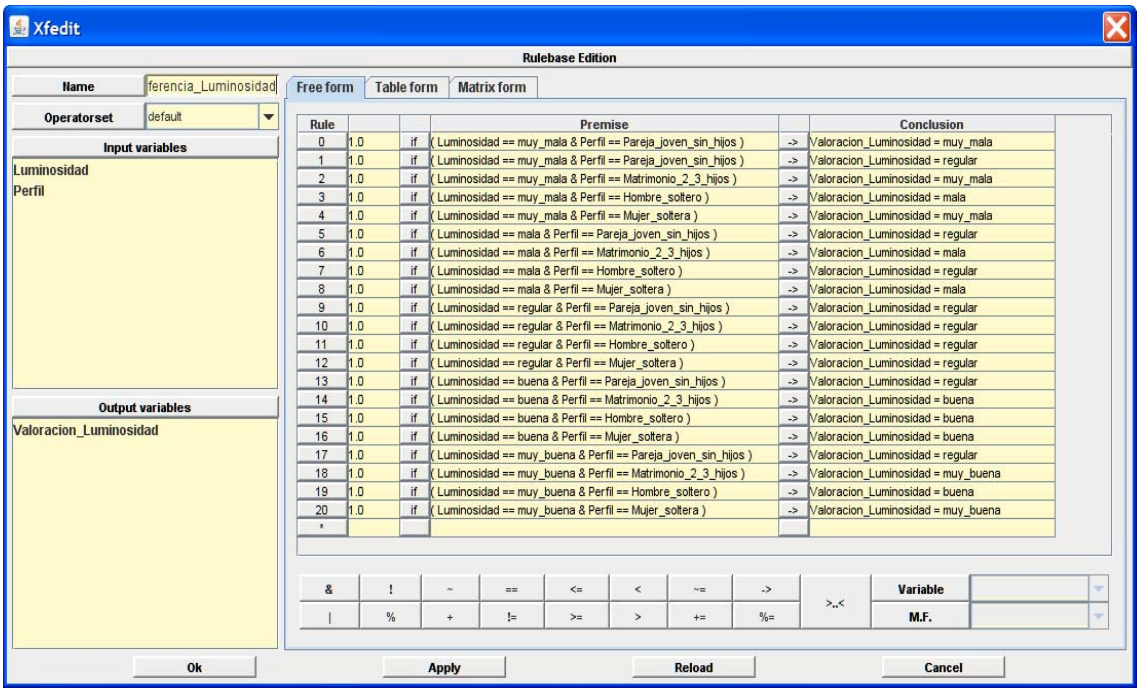
The main original contribution of this work is to offer a first approach to handle requirements with fuzzy information. The use of fuzzy logic helps to shorten time during the different steps in a software engineering process, especially in the elicitation of knowledge phase, reducing in this way some critical phases of the processes that involve the client, mainly requirements statement.

Our proposal has been applied to real state valuation. The development of this applications shows the different phases of implementation of this proposed methodology that allows the clients and the buyers to make a decision about the right price of a house and the right selection according to some fuzzy preferences. This implementation allows a first analysis of practical advantages.

The procedure here proposed benefits not only the customers when they explicitly state their preferences, but also the process engineer, making the software design task easier.

Furthermore, the same technique can be applied to similar situations. For example, in Augusto *et al.*[17] the management of uncertainty in Smart Home by monitoring the human behaviour and sensor signals is presented.

A fuzzy specification of human behaviours and its application to the management of Smart Home and other environments is proposed as future work.



Name: ferencia_Luminosidad

Operatorset: default

Input variables

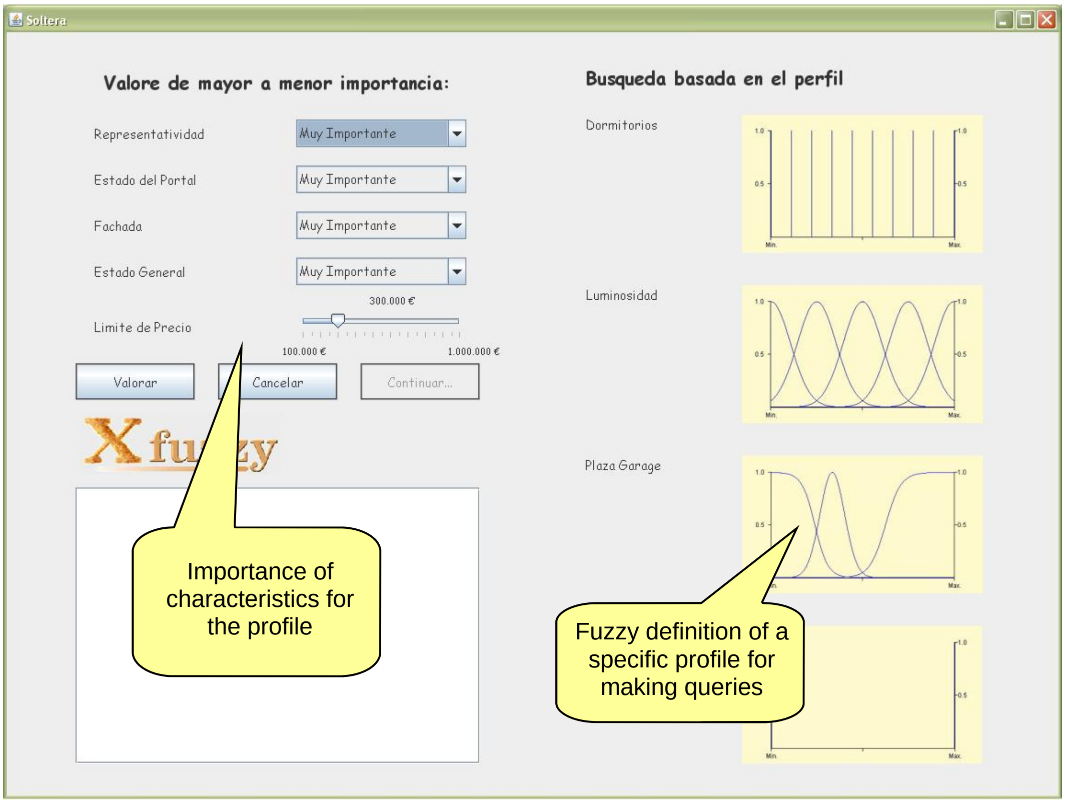
Luminosidad
Perfil

Output variables

Valoracion_Luminosidad

Rule			Premise		Conclusion
0	1.0	if	(Luminosidad == muy_mala & Perfil == Pareja_joven_sin_hijos)	->	Valoracion_Luminosidad = muy_mala
1	1.0	if	(Luminosidad == muy_mala & Perfil == Pareja_joven_sin_hijos)	->	Valoracion_Luminosidad = regular
2	1.0	if	(Luminosidad == muy_mala & Perfil == Matrimonio_2_3_hijos)	->	Valoracion_Luminosidad = muy_mala
3	1.0	if	(Luminosidad == muy_mala & Perfil == Hombre_soltero)	->	Valoracion_Luminosidad = mala
4	1.0	if	(Luminosidad == muy_mala & Perfil == Mujer_soltera)	->	Valoracion_Luminosidad = muy_mala
5	1.0	if	(Luminosidad == mala & Perfil == Pareja_joven_sin_hijos)	->	Valoracion_Luminosidad = regular
6	1.0	if	(Luminosidad == mala & Perfil == Matrimonio_2_3_hijos)	->	Valoracion_Luminosidad = mala
7	1.0	if	(Luminosidad == mala & Perfil == Hombre_soltero)	->	Valoracion_Luminosidad = regular
8	1.0	if	(Luminosidad == mala & Perfil == Mujer_soltera)	->	Valoracion_Luminosidad = mala
9	1.0	if	(Luminosidad == regular & Perfil == Pareja_joven_sin_hijos)	->	Valoracion_Luminosidad = regular
10	1.0	if	(Luminosidad == regular & Perfil == Matrimonio_2_3_hijos)	->	Valoracion_Luminosidad = regular
11	1.0	if	(Luminosidad == regular & Perfil == Hombre_soltero)	->	Valoracion_Luminosidad = regular
12	1.0	if	(Luminosidad == regular & Perfil == Mujer_soltera)	->	Valoracion_Luminosidad = regular
13	1.0	if	(Luminosidad == buena & Perfil == Pareja_joven_sin_hijos)	->	Valoracion_Luminosidad = regular
14	1.0	if	(Luminosidad == buena & Perfil == Matrimonio_2_3_hijos)	->	Valoracion_Luminosidad = buena
15	1.0	if	(Luminosidad == buena & Perfil == Hombre_soltero)	->	Valoracion_Luminosidad = buena
16	1.0	if	(Luminosidad == buena & Perfil == Mujer_soltera)	->	Valoracion_Luminosidad = buena
17	1.0	if	(Luminosidad == muy_buena & Perfil == Pareja_joven_sin_hijos)	->	Valoracion_Luminosidad = regular
18	1.0	if	(Luminosidad == muy_buena & Perfil == Matrimonio_2_3_hijos)	->	Valoracion_Luminosidad = muy_buena
19	1.0	if	(Luminosidad == muy_buena & Perfil == Hombre_soltero)	->	Valoracion_Luminosidad = buena
20	1.0	if	(Luminosidad == muy_buena & Perfil == Mujer_soltera)	->	Valoracion_Luminosidad = muy_buena
*					

Fig. 10: Fuzzy inference rules in Xfuzzy



Valore de mayor a menor importancia:

Representatividad: Muy Importante

Estado del Portal: Muy Importante

Fachada: Muy Importante

Estado General: Muy Importante

Limite de Precio: 300.000 € (100.000 € – 1.000.000 €)

Busqueda basada en el perfil

Importance of characteristics for the profile

Fuzzy definition of a specific profile for making queries

Fig. 11: Fuzzy Application for Estate Valuation

6. References

1. V. López, L. Garmendia, J. Montero and G. Resconi, Specification and computing states in fuzzy algorithms, *Uncertainty, Fuzziness and Knowledge-Based Systems*, **16** (2008) 301-336.
2. V. López and J. Montero, Software engineering specification under fuzziness, *Multiple-Valued Logic and Soft Computing* **15** (2009) 209-228.
3. J. Montero, V. López and D. Gómez, The role of fuzziness in decision making in *Fuzzy Logic: an spectrum of applied and theoretical issues,* (Springer, 2007), pp. 337-349.
4. S.L. Pfleeger, *Software Engineering, theory and application* (Pearson Education, 2002).
5. R.S. Pressman, *Software Engineering, a practical approach*, 5th edn. (Mc Graw Hill, 2002).
6. I. Sommerville, *Software Engineering*, 7th edn. (Pearson Education, 2005).
7. C. Matthews and P.A. Swatman, Fuzzy concepts and formal methods: some illustrative examples in *Proc. 7th Asia-Pacific Software Engineering Conf.* (IEEE Computer Society, Los Alamitos, 2000).
8. C. Matthews P. A. and Swatman, Fuzzy concepts and formal methods: A Fuzzy Logic Toolkit for Z in *ZB*, in *Proc 1st Int. Conf. of B and Z Users on Formal Specification and Development in Z and B* (Springer-Verlag, London, UK, 2000), pp. 491-510.
9. L. A. Zadeh, Fuzzy Sets, *Information and Control*, **8** (1965) 338-353.
10. V. Lopez, J. Montero and A. del Monte, Fuzzy logic in real estate valuation in *Computational Intelligence in Decision and Control*, World Scientific, **1** (2008) pp. 1021-1026.
11. D. Jacobs, Requirements Engineering So Things Don't Get Ugly *CrossTalk, The Journal of Defense Software Engineering.* (2004) 19-25.
12. V. López and J. Montero, Fuzzy Specification in Software Engineering, in *Proc. JCIS-07* (World Scientific, USA, 2007), pp. 1377-1383.
13. XFuzzy 3.0. Centro Nacional de Microelectrónica, Instituto de Microelectrónica de Sevilla, http://www.imse.cnm.es/xfuzzy, 2008.
14. J. Noppen, P. Van den Broek, and M. Askit., Dealing with Fuzzy Information in Software Design Methods in *North American Fuzzy Information Processing Society*. (IEEE 2004) **1**, pp. 22-27.
15. Taxation Society, *http://web.st-tasacion.es/html/index.php*, 2008.
16. V. López and J. Montero, REV: Valuation and price adjustment in a fuzzy logic framework, in *Proc. Int. Conf. on Intelligent System and Knowledge Engineering ISKE2008* (China, 2008), pp. 640-645.
17. J. C. Augusto, J. Liu, P. McCullagh and J.B. Yang, Management of Uncertainty and Spatial-Temporal Aspects for Monitoring and Diagnosis in a Smart House, *Int. J. Computational Intelligence Systems* **1**(4) (2008) 361-378.