

New features of the fuzzy logic development environment Xfuzzy

Angel Barriga, Santiago Sánchez-Solano, Iluminada Baturone, *Diego López, **Francisco Moreno-Velo, Piedad Brox, Federico Montesino, Nashaat M. Hussein Hassan, María Brox, Andrés Gersnoviez

Instituto de Microelectrónica de Sevilla,
Avda. Reina Mercedes s/n, Edif. CICA, 41012-Sevilla, Spain
barriga@imse.cnm.es

* RedIRIS, the Spanish National Research and Education Network
Edif. Bronce, Pza. Manuel Gomez Moreno s/n. Planta 2. E-28020 Madrid, Spain

** Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática
Escuela Politécnica Superior. Carretera Huelva - La Rábida
21071 Palos de la Frontera (Huelva), Spain

Abstract

The characteristics of the new version of the fuzzy systems development environment Xfuzzy is presented. The environment covers the aspects related to the specification, verification, adjustment and implementation of fuzzy systems. It is an open environment (in the sense that the user can define many functional and structural aspects) and a free distribution tool that allows proving new formalisms and helps the definition and implementation of complex systems.

1 Introduction

The use of CAD tools provides many advantages for designing complex fuzzy systems such as the reduction of the cost and the time-to-market of a potential product and also easing the exploration of new theoretical aspects. This is why several tools have been tailored to the fuzzy paradigm in the last few years. The limitation is that many of them are dedicated to specific realizations and/or have constraints on the set of fuzzy operations they support, the complexity of the systems they can design, and their capability for automatic tuning, simulation or synthesis.

The fuzzy system development environment Xfuzzy integrates a set of tools that ease the user to cover the several stages involved in the design process of fuzzy logic-based inference

systems, from their initial description to their final implementation.

The new features of Xfuzzy3.1 (<http://www.imse.cnm.es/Xfuzzy/>) using a new release of the specification language called XFL3 which extends its predecessor advantages, and new CAD tools at the verification stage, tuning stage and synthesis stage.

Xfuzzy has been entirely programmed in Java. Hence, it can be executed on any platform where JRE (Java Runtime Environment) is installed. Xfuzzy executes under GPL license based on freeware GNU.

2 XFL3 Specification Language

The core of the Xfuzzy environment is the fuzzy system specification language called XFL3. It allows designing hierarchical rule bases, which can interchange fuzzy or non-fuzzy values and can employ the same or different fuzzy operators. Moreover, the user can define new fuzzy connectives, linguistic hedges, membership functions and defuzzification methods.

In order to describe complex fuzzy systems the specification can be improved by the use of complex antecedent parts in the rules, that is, by connecting the several antecedents using any kind of conjunctive and/or disjunctive connectives, by relating input variables with fuzzy sets using any kind of linguistic hedges, and by even applying linguistic hedges to some

connected antecedents. As a result, two related advantages are obtained: the expressiveness and linguistic interpretability of the resulting fuzzy system increases, and the number of rules and fuzzy sets required to model a linguistic knowledge is reduced. For instance, the following rule shows this fact:

“If y is small and, either x or z are not close to their target values, and z is greater than -90° and smaller than 90° , then drive forward”

This rule is expressed in XFL3 as:

“if(y == Small & (x != Zero | z != Zero) & z > LB & z < RB) -> v = Forward”

The language has been improved with the concept of family of membership functions that facilitates the fuzzy system specification. The definition of membership functions families (they are related to the linguistic labels of a variable) uses a list of parameters to define the whole set of functions.

The definition of membership functions as families introduces several advantages. Firstly, the number of parameters is smaller than if free membership functions would be used. This eases the tuning of the fuzzy system and permits the use of some automatic learning algorithms (such as simulated annealing or genetic algorithms), which are not appropriate with a large number of parameters. Secondly, the use of a common list of parameters facilitates the assignment of a meaning for each membership function. In particular, the membership functions cannot evolve into the construction of a state with highly overlapped or disordered functions in an automatic modification process. Finally, a family of membership functions can be defined with a fixed overlapping degree. This is very useful to simplify the inference mechanism (by using grid partitions), which is essential to perform the hardware synthesis of the fuzzy systems. The main disadvantage of this kind of definition is the imposition of hard constraints in the functions, which could be an obstacle to achieve a suitable system optimization (especially in the description of the output variables).

Moreover, this version of XFL3 introduces a novelty related to the definition of the system specification by means of structural instances, which combine two different types of components: fuzzy rule bases and non-fuzzy blocks.

3 Design methodology

The development environment for fuzzy systems, Xfuzzy, does not impose any design methodology but it allows adapting the needs of the designer for a particular purpose. The tools are integrated into the environment do not impose any restriction to the design flow since they are independent tools to perform different tasks for the development of fuzzy systems.

Considering this premise a possible design flow is depicted in Figure 1. The specification of the system can be obtained from the knowledge expressed by an expert or it can be provided by a set of numerical data. In the first case, there are tools that facilitate the description of the fuzzy system. In the second case, there are identification tools that allow to extract the knowledge base from numerical data.

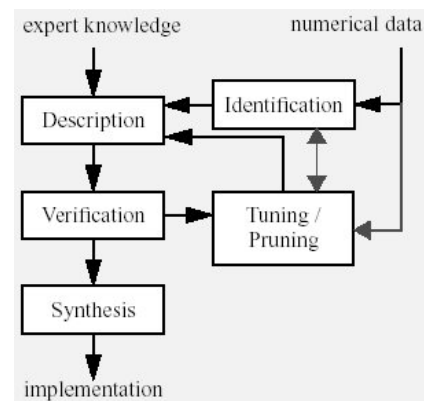


Figure 1: A design methodology

As soon as the knowledge base is defined, it is necessary to verify the behavior of the system. Simulation and representation tools are required to carry out this task.

Another required task consists of fitting the parameters of the system and simplifying the knowledge base, in order to refine the specification, by means of tuning/pruning tools.

The verification and adjustment activities are in the feedback flow of the design task. When the

specification of the fuzzy system fits with the requirements and the specifications, the last stage of the design process is the final implementation. To achieve it, Xfuzzy integrates different types of software and hardware implementation tools.

4 New CAD tools

The new version of Xfuzzy environment includes new tools that extend the capacities of the specification, monitoring, adjustment and synthesis of fuzzy systems. Figure 2 shows the block diagram of the tools included in the environment. These tools are grouped in four clusters associated with the different design stages of a fuzzy system: the description stage, the verification stage, the adjustment stage and the synthesis stage.

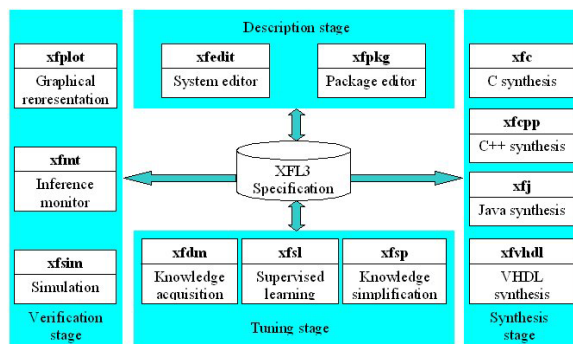


Figure 2: Xfuzzy tools

The description tools facilitate the graphical specification of the fuzzy system. The system is described by means of a specification using the XFL3 language. There are some graphical tools like *xfedit* and *xfpkg*, which help such specification. *Xfedit* tool eases the edition of operator sets and hierarchical systems. Figure 3 illustrates some *xfedit* windows. *Xfpkg* allows to edit function packages that contain the descriptions of the fuzzy operators (binary functions like min, max, etc.), linguistic hedges (unary functions like strongly, more or less, etc.), membership functions (triangles, trapezoids, etc.), and defuzzification methods (center of area, first maximum, etc.). The user can freely define his own functions with this tool.

Once a fuzzy system has been described the verification tools allow testing its functionality.

In this way, *xfplot* lets represent graphically systems of 2 and 3 dimensions as it is shown in Figure 4. The *xfmt* tool allows to modify the input variables values and to visualize the effect on the system (rule base, membership functions and output variables). Finally, *xfsim* is a simulation tool that allows inserting the fuzzy inference engine into a wider system containing other elements described in Java. It allows realizing behavior simulations in order to validate the specification of the fuzzy system.

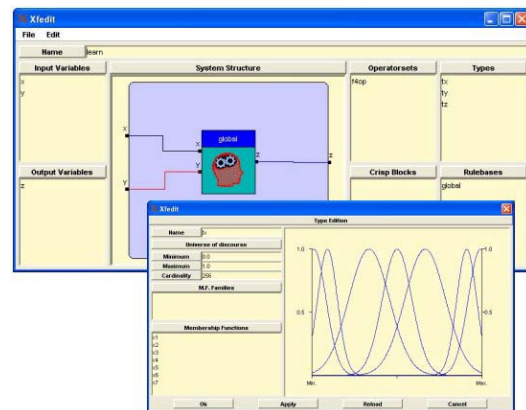


Figure 3: Xfedit windows

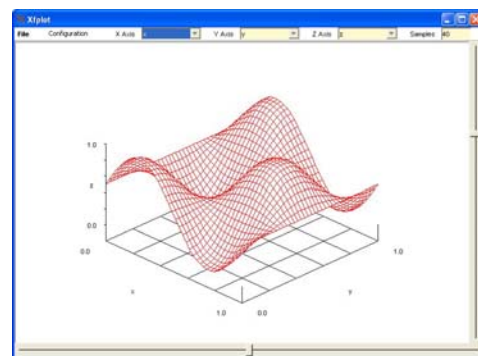


Figure 4: Xfplot tool

xfsl tool performs different supervised learning algorithms. It has been renewed to include new algorithms as well as pre- and post-processing techniques to simplify the obtained rule bases. In this sense, it includes a wide set of supervised learning algorithms and it is able to cope with complex fuzzy systems. In particular, *xfsl* (figure 5) is able to adjust hierarchical fuzzy systems; systems described by the user freely using different membership or connective functions, defuzzification methods, or even linguistic hedges; and fuzzy systems with continuous outputs (such as fuzzy controllers) as well as categorical outputs (such as fuzzy classifiers).

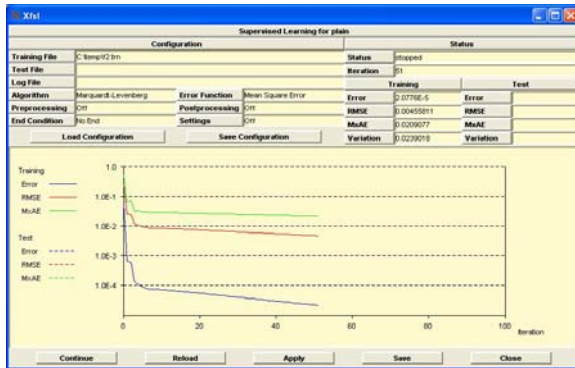


Figure 5: *Xfsl* window

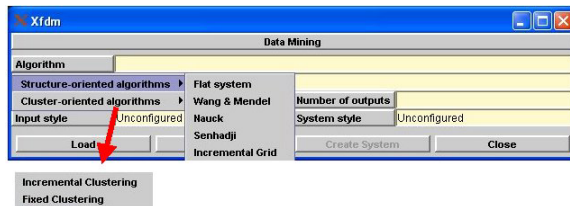


Figure 6: *Xfdm* window

There are also identification tools (*xfdm*) that extract the fuzzy system knowledge from data applying various data mining techniques as well as a knowledge base simplification tool (*xfsp*). Several techniques have been proposed in the literature to extract symbolic knowledge (rules) from data. Among those dedicated to generate fuzzy modules, two groups can be distinguished: one group generates fuzzy systems based on a grid partition of the input universes of discourse, while the other techniques use partitions based on data clustering. The efficiency of a particular grid- or clustering based technique depends very much on the application. This is why we have developed a CAD tool to cover as much as possible the different possibilities. Figure 6 shows the *xfdm* window with the implemented algorithms.

The system implementation can be performed by means of synthesis tools. There are two sets of tools: software and hardware synthesis tools. The software one produces the system implementation as C (*xfc*), C++ (*xfc++*) or Java (*xfj*) functions. The hardware synthesis tool *xfvhdl* produce the fuzzy system description in terms of VHDL hardware description language. This kind of description can be synthesized using a standard hardware synthesis tool. The implementation strategy is based on specific architecture of fuzzy processing element.

Acknowledgements

This work was partially supported by projects TEC2005-04359/MIC from the Spanish Ministry of Education and Science as well as TIC2006-635 from the Andalusian regional Government

References

- [1] I. Baturone, A. Barriga, S. Sánchez-Solano, C. Jiménez, D.R. López, "Microelectronic design of fuzzy logic-based systems", CRC Press., 2000. ISBN: 0-8493-0091-6.
- [2] F.J. Moreno-Velo, S. Sánchez-Solano, A. Barriga, I. Baturone, D.R. López, "XLF3: a New Fuzzy System Specification Language", in Advanced in Scientific Computing, Computational Intelligence and Applications. Edited by N. Mastorakis, V. Mladenov, B.S. Suter, L.J. Wang. WSES, pp. 361-366., 2001.
- [3] F. J. Moreno Velo, I. Baturone, S. Sánchez Solano, A. Barriga, "The Parametric Definition of Membership Functions in XFL3", IEEE International Conference on Fuzzy System (FUZZIEEE'2004), Budapest (Hungary), July 2004.
- [4] F. J. Moreno-Velo I. Baturone F. J. Barrio-Lorite, S. Sánchez-Solano, A. Barriga, "A design methodology for complex fuzzy systems", European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (EUNITE'2003), Oulu (Finland), July 2003.
- [5] F. J. Moreno-Velo, I. Baturone, S. Sánchez-Solano, and A. Barriga, "Rapid design of fuzzy systems with XFUZZY", IEEE International Conference on Fuzzy System (FUZZIEEE'2003), St. Louis (USA), May 2003