GUAJE - A JAVA ENVIRONMENT FOR GENERATING UNDERSTANDABLE AND ACCURATE MODELS

José M. Alonso Luis Magdalena

European Centre for Soft Computing (ECSC)
33600 Mieres, Asturias, Spain
{jose.alonso,luis.magdalena}@softcomputing.es

Abstract

The term Soft Computing is usually used to refer to a family of several preexisting techniques (Fuzzy Logic, Neuro-computing, Probabilistic Reasoning, Evolutionary Computation, etc.) able to work in a cooperative way, taking profit from the main advantages of each individual technique, in order to solve lots of complex real-world problems for which other classical techniques are not quite well suited. In the specialized literature there are many Soft Computing tools, most of them freely available as open source software. This work gives an overview on existing tools for system modeling. Moreover, it introduces a new environment for building interpretable and accurate systems by means of combining several preexisting tools.

Keywords: Soft Computing, fuzzy modeling, open source software, interpretability-accuracy trade-off.

1 INTRODUCTION

Soft Computing (SC) is usually defined by its essential properties, as a family of techniques, as a complement of hard computing, and/or as a tool for coping with imprecision and uncertainty [10].

One of the main issues regarding SC techniques is their cooperative nature. Each individual technique, even each individual algorithm, has its own advantages and drawbacks. Therefore, designing hybrid systems made up of different techniques working together let us achieving more powerful systems, overcoming the problems which turn up when dealing with the component techniques alone. That is why hybrid systems like for instance neuro fuzzy systems (NFS) [15] and genetic fuzzy systems (GFS) [5] are becoming more and more popular.

Since this paper deals with modeling understandable and accurate systems, it is mainly focused on those SC techniques which give priority to interpretability what involves two main aspects. First, the system description readability, i.e., the system description has to be transparent enough to present the system as a whole, describing its global behavior and trend. Second, the system understandability, i.e., its explanation ability for considering all possible individual situations and explaining specific behaviors for specific events.

The semantic expressivity of Fuzzy Logic (FL) [18] is well-known to be close to expert natural language yielding powerful tools for linguistic concept modeling. The use of linguistic variables [19] and linguistic rules [11] favors the interpretability of fuzzy models, at least from the readability or structural transparency point of view. As a result, Fuzzy Modeling (FM) [8], i.e., system modeling with fuzzy rule-based systems (FRBS), represents a fruitful research line. Unfortunately, using FL is not enough for building interpretable models. The whole modeling process must be carried out carefully, paying special attention to interpretability from the beginning to the end and imposing several constraints [13]. In addition, when dealing with complex real-world problems fuzzy models can be upgraded with learning capabilities of other SC techniques. Nevertheless, hybrid systems (mainly NFS and GFS) should also be designed keeping in mind the interpretability requirement.

Interpretability must be the central point on system modeling. In fact, some of the most hot and modern research topics like Precisiated Natural Language (PNL), Computing With Words (CWW), and/or Human Centric Computing (HCC) strongly rely on the characteristic interpretability of fuzzy models. However, it is important to remark that generating inter-

pretable systems is not a straightforward task. Nowadays, most fuzzy models are obtained giving priority to accuracy, disregarding their interpretability, what may be a great error because it yields almost black-box models. Notice that, looking for a good interpretability-accuracy trade-off is one of the most complex tasks on system modeling. It demands the aid of powerful software tools.

The rest of the paper is structured as follows. The next section makes a global review on available software that implements SC techniques for system modeling. Then, section 3 presents a new modeling framework based on the combination of several available tools. Finally, section 4 draws some conclusions.

2 AVAILABLE SOFTWARE

Most software for system modeling is available on the Web in the form of libraries and/or small tools which often come from academics and small research groups. Some of the most famous packages and tools are the following. In the field of evolutionary computation, JCLEC[1] (Java Class Library for Evolutionary Computation) and JMetal[2] (Metaheuristic Algorithms in Java) provide two nice frameworks for both evolutionary and multi-objective optimization. JavaNNS[3] (Java version of Stuttgart Neural Network Simulator) is probably the best free suite for neural networks. Regarding fuzzy modeling, Xfuzzy[4] (a development environment for fuzzy-inference-based systems), FisPro[5] (Fuzzy Inference System Professional) and KBCT[6] (Knowledge Base Configuration Tool) represent three useful tools. Finally, regarding neuro-fuzzy algorithms we can point out, among others, to NEFCLASS[7] (Neuro-Fuzzy Classification).

In order to get wider visibility and cooperation with other researchers all tools enumerated above are freely downloadable as open source software, at least for research and education purposes. Thanks to the huge amount of available software it is really easy creating new small prototypes for lots of applications without the effort of starting from scratch. However, such prototypes not always work properly when dealing with real-world complex problems like large scale business applications. The main problem of such developments is their maintenance cost. Keeping a flexible and well-documented source code is a mandatory requirement in order to promote the cooperation of several researchers in a common development. In addition, the coordination and control of subversions is a really difficult task when several researchers, sometimes located at different parts of the world, are only working on the software development during their own free time.

In consequence, the vast majority of freely available software is developed by small isolated groups and it becomes out of date resulting obsolete after a few months or years. A solution to this problem would be creating an international project to promote the cooperation of several research groups with the aim of creating a new open source software GNU[8] Fuzzy that could be taken as a reference implementation [14]. It should be developed following open standards in order to make easier the cooperation of researchers all along the world. In addition, it should let adding easily new functionalities and modules keeping a common interface over a base core. Of course, a much more conservative alternative consists in using commercial tools like the Matlab[9] toolboxes which include the well-known Fuzzy Toolbox and ANFIS (Adaptive Neuro-Fuzzy System) tool. Nevertheless, we should not reject the use of open source software because it would mean loosing the richness of quickly incorporating new developments made by the active research community which is always working in emerging fields.

We are still far from a standard GNU Fuzzy Toolbox, but there are some interesting and successful attempts for going beyond the small and specialized tools, all of them with both research and educational purposes. For instance, FrIDA [4] is free and open source software in the form of a java-based graphical user interface (GUI) that joins several individual tools for data analysis and visualization. In this case all small programs were developed by the same researchers over the years. KEEL (Knowledge Extraction based on Evolutionary Learning) [6] is another more ambitious software tool created as part of a research project with several goals. To start with it includes a huge repository made up of hundreds of evolutionary learning algorithms developed by several authors (belonging to different research groups) as part of their own research works. Furthermore, new algorithms can be easily added. In addition, KEEL offers a user-friendly java GUI for designing experiments where different algorithms can be fairly compared with exactly the same data under a complete statistical analysis. Lastly, another quite famous tool putting together several machine-learning algorithms under the same interface is Weka[10] (Data Mining Software in Java) [17]. It is also developed fol-

[1] http://jclec.sourceforge.net/
[2] http://jmetal.sourceforge.net/
[3] http://www.ra.cs.uni-tuebingen.de/SNNS/
[4] https://forja.rediris.es/projects/xfuzzy/
[5] http://www.inra.fr/internet/Departements/MIA/M/fispro/
[6] http://www.mat.upm.es/projects/advocate/kbct.htm
[7] http://fuzzy.cs.uni-magdeburg.de/nefclass/
[8] http://www.gnu.org/copyleft/gpl.html
[9] http://www.mathworks.com/
[10] http://www.cs.waikato.ac.nz/ml/weka/

lowing the open source philosophy and it counts with a lot of related projects and contributors. It applies the Linux model of releases. It focuses on automatic extraction of knowledge from data but, unfortunately, it does not take care of the interpretability of the generated models and it does not include any algorithms for fuzzy modeling.

3 GUAJE ENVIRONMENT

The main novelty of the GUAJE approach is that it is the first one combining several software tools (not only libraries) with the aim of building interpretable fuzzy models. Notice that, interpretability is the main requirement and it is taken into consideration along the whole modeling process. Of course, accuracy is not forgotten because all kind of system must achieve at least a minimum accuracy, being completely useless otherwise. What we want to highlight is the fact that our proposal is especially designed for humanistic systems (defined by Zadeh as those systems whose behavior is strongly influenced by human judgment, perception or emotions [19]) used in real-world applications (for instance decision-support systems in fields like education, robotics, medicine, etc.) where there is a huge human-system interaction and, as a result, the interpretability of the model is strongly appreciated. Moreover, some loss of accuracy may be tolerated in exchange for a more interpretable model.

The core of GUAJE is the last downloadable version of KBCT (version 3.0) which has been upgraded with new functionalities and implementing the HILK (Highly Interpretable Linguistic Knowledge) fuzzy modeling methodology [2]. GUAJE combines the following six preexisting tools (the first five ones are freely available as open source) making use of the tools and methods that they provide:

- **KBCT**. Open source software for knowledge extraction and representation which combines expert knowledge and induced knowledge (knowledge automatically extracted from data) [1]. The combination of both kind of knowledge is made carefully and it includes consistency analysis, simplification, and optimization tasks.
- **FisPro**. An open source tool for creating fuzzy inference systems (FIS) to be used for reasoning purposes, especially for simulating a physical or biological system. It includes many algorithms (most of them implemented as C programs) for generating fuzzy partitions and rules directly from experimental data. In addition, it offers data and FIS visualization methods in a java-based user-friendly GUI.
- **Xfuzzy**. A free software development environment for generating FIS. It integrates a set of tools that ease the user to cover the several stages involved in the whole designing process, from their initial description to their final implementation, including simulation, edition and program synthesis. It is written in Java and all its tools are based on a common specification language named XFL3.
- **ORE**[11] (Ontology Rule Editor) [12]. A java-based open source platform-independent application for defining, managing and testing inference rules on a model represented by a specific ontology.
- **Weka**. An open source tool providing lots of algorithms for data mining. It includes the implementation of many classical algorithms like for example J48 which corresponds to the well-known C4.5 algorithm.
- **Matlab Fuzzy Toolbox**. It is the most widely used commercial tool for fuzzy systems. Its main advantage is that it takes profit from the fact that it is fully integrated with all functionalities provided by Matlab environment which is commonly used in engineering for both educational and business applications.

The Figure 1 illustrates how all these tools cooperate in the GUAJE environment. It shows the tasks made by each tool.

We consider two main sources of knowledge, experimental data and expert knowledge. An expert is a person who perfectly knows the problem under analysis. The expert is able to describe the system behavior and his/her background (knowledge, experience, preferences, etc.) is essential in order to get a good model. For that reason, as it can be seen in the diagram expert knowledge plays a key role being present all along the modeling process. Although GUAJE can work in a fully automatic way it also lets expert supervision and interaction at each step. Such integration only is possible if both expert and induced knowledge are formalized using the same language. In this case we use FL along with a set of constraints to guarantee the interpretability of the generated model.

Elicitation of expert knowledge is a really complex task and it usually becomes a bottleneck in the whole modeling process [7, 9]. In order to make easier expert knowledge extraction and representation we can use an intermediate level, the domain ontology. There are so many web ontologies that it is really easy to find one

[11] http://sourceforge.net/projects/ore/

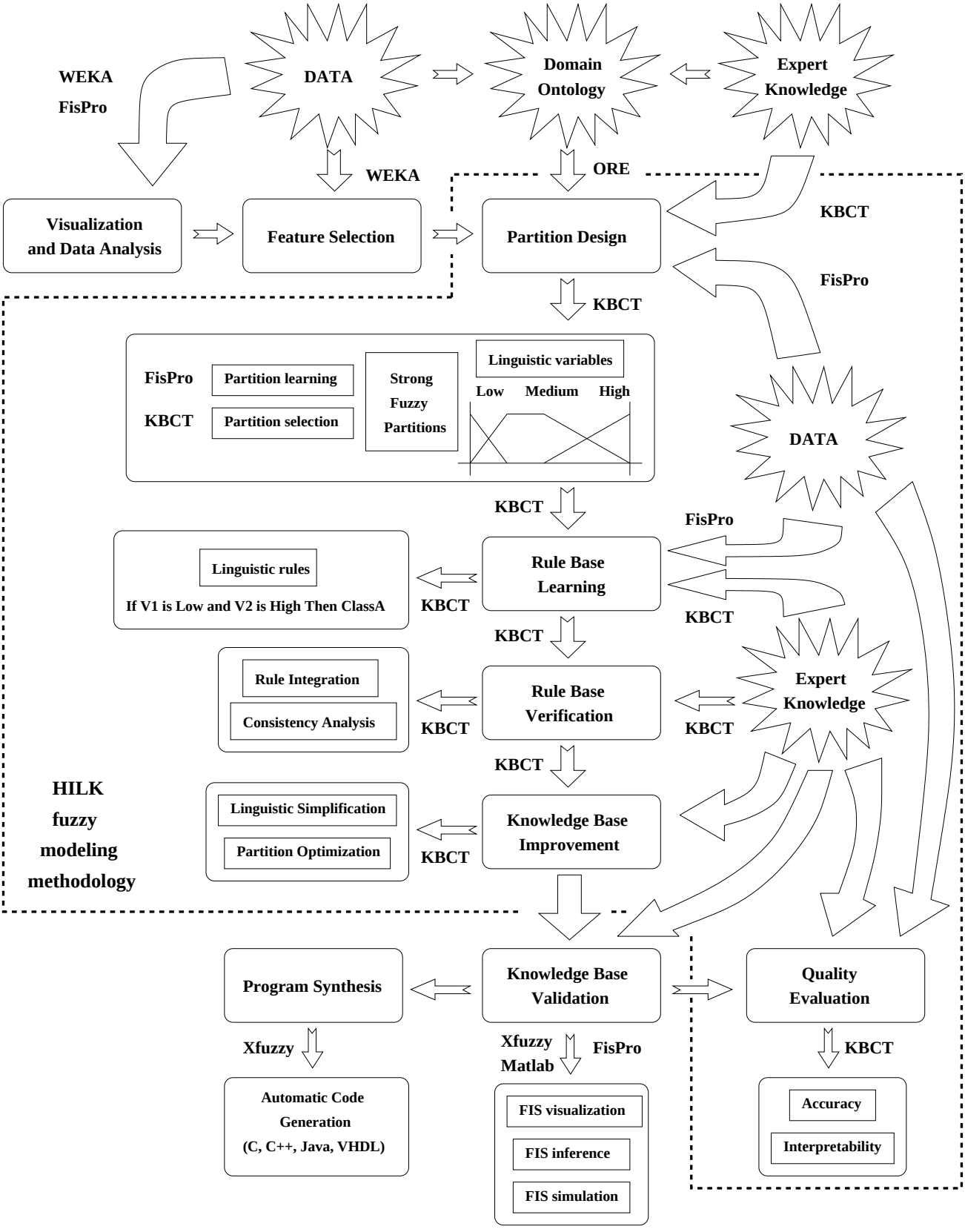

Figure 1: Scheme of the proposed GUAJE environment.

related to the problem under consideration. The hierarchy of concepts included in the selected ontology can be shown graphically to the expert who should identify the main influential input-output variables as well as a small set of basic expert rules [3]. All tasks to be done in relation with ontologies such as web searching, downloading, handling, representation, and so on are made calling to ORE functions directly from the KBCT graphical interface. As result of this preliminary stage, regardless of whether we use ontologies or not, we obtain a first simple expert knowledge base (KB) that has to be complemented and upgraded later with knowledge induced from experimental data.

The handling of data includes both visualization and analysis tasks that are carried out by KBCT making calls to Weka and FisPro algorithms. The first step consists in feature selection which is crucial to keep interpretability when dealing with large data files made up of dozens of input variables. Weka provides many algorithms for feature selection. Then, it is possible to use algorithms provided by FisPro for partition and automatic rule generation from data. Such algorithms are especially designed for generating interpretable partitions and rules. As a result, we can build a whole KB directly from experimental data.

It is important to remark that expert and induced KBs can be built in parallel or sequential steps. However, in order to get a unique KB the integration must be made carefully at both partition and rule levels. Therefore, for the sake of interpretability we recommend first closing the partition design stage (including both expert and data), i.e., defining fuzzy partitions with a global semantics before starting the rule base definition. Although it is not mandatory, we recommend the use of strong fuzzy partitions (SFP) [16] which satisfy most demanded semantic constraints (distinguishabilityy, coverage, normality, convexity, etc.) to design interpretable partitions. Notice that, linguistic comparison for consistency analysis is only feasible when all rules (expert and induced ones) are defined using the same linguistic terms defined by the same fuzzy sets.

Once we have achieved a unique and consistent KB it is time to think about the interpretability-accuracy trade-off. KBCT offers powerful algorithms for linguistic simplification with the aim of increasing even more the KB interpretability while preserving the accuracy. It starts looking for redundant elements (labels, inputs, rules, etc.) that can be removed without altering the system accuracy. Then, it tries to merge elements always used together. Lastly, it forces removing elements apparently needed but not contributing too much to the final accuracy.

After getting a compact KB, accuracy can also be increased by applying optimization techniques for tuning the fuzzy partitions. Notice that, KBCT provides some optimization algorithms which are strongly constrained with the aim of not penalizing too much interpretability. The partition tuning process must keep the matching in between fuzzy sets and linguistic terms which should be fully meaningful according to the problem context and the expert background.

To sum up, the five main stages of the HILK methodology (partition design, rule base learning, rule base verification, knowledge base improvement, and quality evaluation) constitute the core of GUAJE environment. They are surrounded by a dash line in Figure 1 and they are directly implemented in the new enhanced version of KBCT. Although, for the sake of clarity, the five steps are represented sequentially, in practice the tool is absolutely flexible and the whole process is iterative. Everything can be supervised by an expert who can decide going back to the previous steps at whatever moment.

Finally, at the last stage, KBs built with KBCT can be exported to the format recognized by FisPro, Matlab, and Xfuzzy. In consequence, designed KBs can be used with the inference engines provided by such tools. In addition, we can take profit of the well-known Matlab Simulink environment in order to include the modeled system as part of a more complex simulated system. Furthermore, the program synthesis made by Xfuzzy is really useful for generating a final standalone module to be embedded in a real application. Note that the inverse translation is not allowed, i.e., KBs modified with FisPro, Matlab, or Xfuzzy can not be imported and opened again by KBCT. This is a restriction to preserve the interpretability of the final model because FisPro, Matlab, or Xfuzzy may violate the interpretability constraints imposed and satisfied by KBCT.

4 CONCLUSIONS

This paper has presented a new system modeling suite mainly focused on designing FRBSs with a good interpretability-accuracy trade-off by means of combining several preexisting tools. This approach lets us saving a lot of time because we reuse many algorithms already freely available on the Web as part of other tools which are distributed as open source software.

New algorithms can be added in the future with the aim of complementing the existing ones or adding new functionalities. For instance, we may incorporate some of the algorithms provided by Xfuzzy for partition tuning as well as for rule induction and simplification. We also would like to explore the possibility of

incorporating some algorithms for data analysis, like for example the fuzzy c-means clustering, included in FrIDA. Another open research line regards on adapting HILK methodology (and its implementation included in KBCT) to multi-objective problems. We are thinking on reusing some of the metaheuristic algorithms provided by JMetal.

GUAJE is freely available as open source software. If you are interested in trying its last release, please contact directly by e-mail with the first author of this contribution.

Notice that, due to space limitation we have not included any picture showing graphical interfaces nor application examples. Please, the interested reader is referred to the cited links and papers for further information.

Acknowledgements

GUAJE can be seen as an enhanced version of KBCT. The first version of KBCT was developed as part of the European research project ADVOCATE II supported by the European Commission (IST-2001-34508). The initial development started from FisPro what explains why both tools are so closely linked.

References

[1] J. M. Alonso, L. Magdalena, and S. Guillaume. KBCT: A knowledge extraction and representation tool for fuzzy logic based systems. In *IEEE International Conference on Fuzzy Systems*, pages 989–994, 2004.

[2] J. M. Alonso, L. Magdalena, and S. Guillaume. HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems*, 23(7):761–794, 2008.

[3] J. M. Alonso, A. Muñoz, J. A. Botía, L. Magdalena, and A. F. Gómez-Skarmeta. Uso de ontologías para facilitar las tareas de extracción y representación de conocimiento en el diseño de sistemas basados en reglas borrosas. In *XIV Spanish ESTYLF conference on fuzzy logic and technologies*, pages 233–240, 2008.

[4] C. Borgelt and G. González-Rodríguez. FrIDA - a free intelligent data analysis toolbox. In *IEEE International Conference on Fuzzy Systems*, pages 1892–1896, 2007.

[5] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, volume 19. Advances in Fuzzy Systems - Applications and Theory, World Scientific Publishing Co. Pte. Ltd., 2001.

[6] J. Alcalá-Fdez et al. KEEL: A software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009.

[7] F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. *Building expert systems.* Addison-Wesley, 1983.

[8] H. Hellendoorn and D. Driankov. *Fuzzy model identification.* Springer-Verlag London, UK, 1997.

[9] A. L. Kidd. *Knowledge elicitation for expert systems: A practical handbook.* Plenum Press, 1987.

[10] L. Magdalena. What is soft computing? revisiting possible answers. In *8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 3–10, 2008.

[11] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic systems. *IEEE Transactions on Computers*, 26(12):1182–1191, 1977.

[12] A. Muñoz, A. Vera, J. A. Botía, and A. F. Gómez-Skarmeta. Defining basic behaviours in ambient intelligence environments by means of rule-based programming with visual tools. In *1st Workshop of Artificial Intelligence Techniques for Ambient Intelligence. ECAI*, 2006.

[13] C. Mencar and A. M. Fanelli. Interpretability constraints for fuzzy information granulation. *Information Sciences*, 178:4585–4618, 2008.

[14] D. Nauck. GNU Fuzzy. In *IEEE International Conference on Fuzzy Systems*, pages 1019–1024, 2007.

[15] D. Nauck, F. Klawonn, and R. Kruse. *Foundations of neuro-fuzzy systems.* J. Willey & Sons, Chichester, UK, 1997.

[16] E. H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22–32, 1969.

[17] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques.* 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[18] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[19] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Parts I, II, and III. Information Sciences*, 8, 8, 9:199–249, 301–357, 43–80, 1975.