The role of hybrid systems in intelligent data management: the case of fuzzy/neural hybrids*

Carlos A. Iglesias, José C. González and Juan R. Velasco
Dep. Ing. Sistemas Telemáticos, E.T.S.I. Telecomunicación,
Universidad Politécnica de Madrid
E–28040 Madrid, Spain
{cif,jcg,juanra}@gsi.dit.upm.es

April 4, 1996

Abstract

Data Management is *per se* a hybrid task where the employment of heterogeneous and intelligent techniques is indispensable. In this paper, the case of fuzzy-neural hybrid systems is studied in some detail. In particular, a general fuzzy-neural model is proposed, designed, implemented and evaluated for a control application in the steel industry.

1 Introduction

Data management involves a set of activities (collection, processing/decision, presentation, etc.) that are heterogeneous by nature. Even at the processing level, completely different deductive, abductive and inductive approaches can be devised for different purposes. These approaches include traditional mathematical modelling, statistical methods, rule-based expert systems, fuzzy techniques, Bayesian inference networks, neural networks, etc.

All these approaches are obviously complementary for data management. This fact is leading to a growing interest in hybrid systems: those involving the cooperation of any number of heterogeneous approaches in a single, integrated system.

In particular, the cross-fertilisation of ideas, concepts and methods from the fields of fuzzy and connectionist techniques has lead to a great variety of hybrid (in the wide sense) systems. One or another paradigms are used for knowledge representation/acquisition or for inference purposes with very different levels of integration.

*This work is funded in part by the Commission of the European Communities under the ESPRIT Basic Research Project *MIX: Modular Integration of Connectionist and Symbolic Processing in Knowledge Based Systems*, ESPRIT-9119.

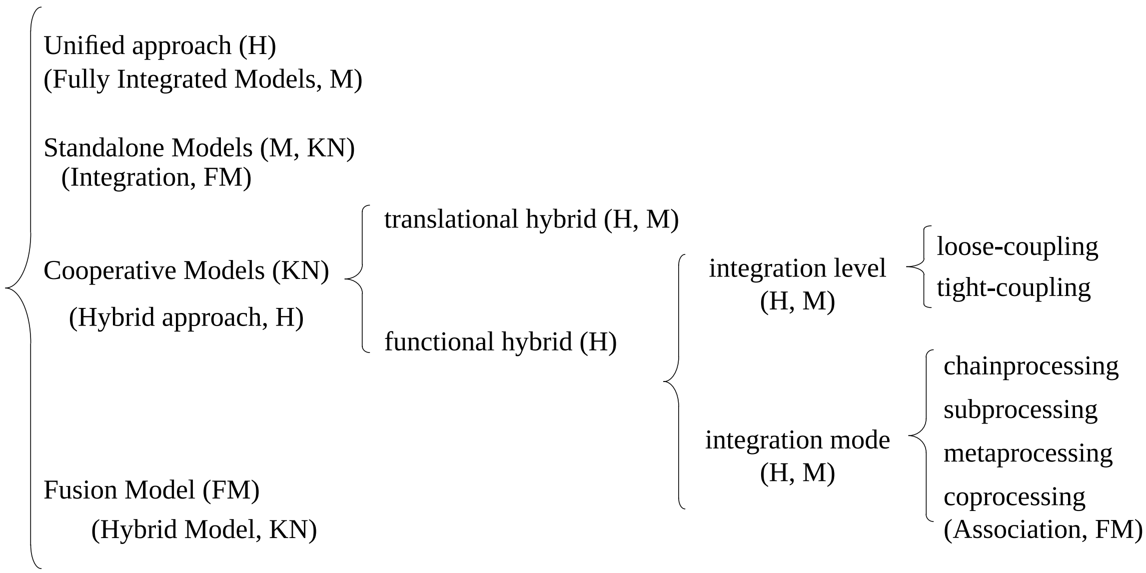

Figure 1: General taxonomy of hybrid systems

The structure of this paper follows this plan: several typologies of fuzzy-neural hybrid systems are initially overviewed and presented in a unified way. Then, a general model for fuzzy-neural hybridization is proposed. A particular instantiation of this model has been implemented as a prototype. This prototype is finally evaluated for a control application in the steel industry.

2 Fuzzy-neural hybrid systems typology

Hilario (H) [4, chapter 7], Medsker (M) [14, chapter 2], Kruse and Nauck (KN) [15] and Funabashi and Maeda (FM) [1] provide different taxonomies for fuzzy-neural hybrid systems. A unified taxonomy is showed in figure 1.

- *Unified approach:* no symbolic structures are needed and all the symbol processing functionalities are performed by neuronal or connectionist methods.
- *Stand-alone models:* independent systems (symbolic and connectionist) are used and the best one is selected.
- *Cooperative models:* different modules symbolic and connectionist(agents) cooperate to perform a task. There are different criteria to define the level of integration:
  - Hilario: *loose-coupling* if data and knowledge is transferred, and *tight-coupling* if it is shared. There is no need for explicit interaction to communicate a change in the data and knowledge structures.
  - Medsker: *loose-coupling* is performed via files and *tight-coupling* is performed via shared data structures or parameter passing.

Funabashi and Maeda distinguish between two architectures: *Combination* and *Association* depending on the grade of autonomy of the intelligent systems. The former combines intelligent systems to perform a task, the latter requires a distributed architecture where each element works autonomously and cooperatively.

- *Fusion:* there is only one connectionist system with fusion of intelligent techniques.

Existing hybrid systems have been also classified in [4] according to the control flow among their fuzzy and connectionist components. This is the dimension used as *reference* along the hybrid models studied in the MIX project. Along this dimension, we can find chain/sub/meta/co-processing systems.

- Chain-processing: the two models operate in sequence, interaction being limited to a single transfer of control and information from one to the other.
- Sub-processing: one of the two subsystems is embedded and subordinated to the other.
- Meta-processing: one of the two subsystems is the base-level problem solver and the other plays a meta-level role.
- Co-processing: the symbolic and connectionist systems are equal partners in the process involved.

According to the characteristics incorporated from both approaches in a single system, seven categories have been reviewed by Magdalena in [13]:

- Connectionist systems with fuzziness
- Symbolic systems with analytical learning
- Logic connectionist systems
- Connectionist inference systems
- Rule-like connectionist systems
- Connectionist systems with symbolic acquisition of knowledge
- Connectionist systems with symbolic description of knowledge

Most of the work in the field of fuzzy/neural hybridization is application-oriented. In particular, the final goal of the systems described in the literature can be classified as clustering, classification, modelling, control and inference [13].

3 A fuzzy-neural hybrid model

In the framework of the MIX project, a particular model for fuzzy/neural hybrid systems has been studied and implemented. For the selection of this model, the following criteria have been kept in mind:

- On one hand, the model had to able of representing a significant proportion of the work carried out until now in the field. In this sense the model has to serve as a way to clarify the relations among the current approaches, systems or applications.
- On the other hand, the model had to address challenging classes of problems. Following this criterion, conceptually simple problems should be neglected.
- The MIX approach to hybrid systems can be considered a new approach because of the use of intelligent agents for integrating different intelligent technologies, as remarked by Medsker [14, page 238]. This approach corresponds to the Association Model cited by Funabashi and Maeda [1]. This is the main reason because an Association/Cooperative Model is proposed.
- Regarding applications, the model had to be preferably apt for the development of the applications that serve as a testbed for the MIX project.
- Finally, the model had to help in making evident the benefits and relative strengths that it might encompass. In this sense, the model should serve as a way to orient future research and development in the field.

We can imagine at least three groups or reasons for developing a fuzzy-neural hybrid model for any application (starting from a purely symbolic fuzzy implementation or from a pure connectionist system):

- Improving system behaviour. In this case, we try to benefit of the synergy of the involved paradigms to augment system accuracy. The error rate is so reduced via cooperation among different agents.
- Improving system performance. In this case, synergistic paradigms are combined to improve, not the error rate, but the overall performance of the system in terms of time.
- Improving system comprehensibility. Here, the purpose of hybridization is making system knowledge more explicit.

The main application of the fuzzy/neural hybrid model proposed here is the first one: improving system behaviour.

The architecture for the model considered in this document is very general, comprising many different hybrid systems. One particular instance of this model, the fuzzy-neural cooperative learning model, is described and evaluated in the following. The main goal of this model is obtaining an improved behaviour of the system by using the learning capabilities of both, connectionist and fuzzy systems. In particular, we consider a prototype where (figure 2):

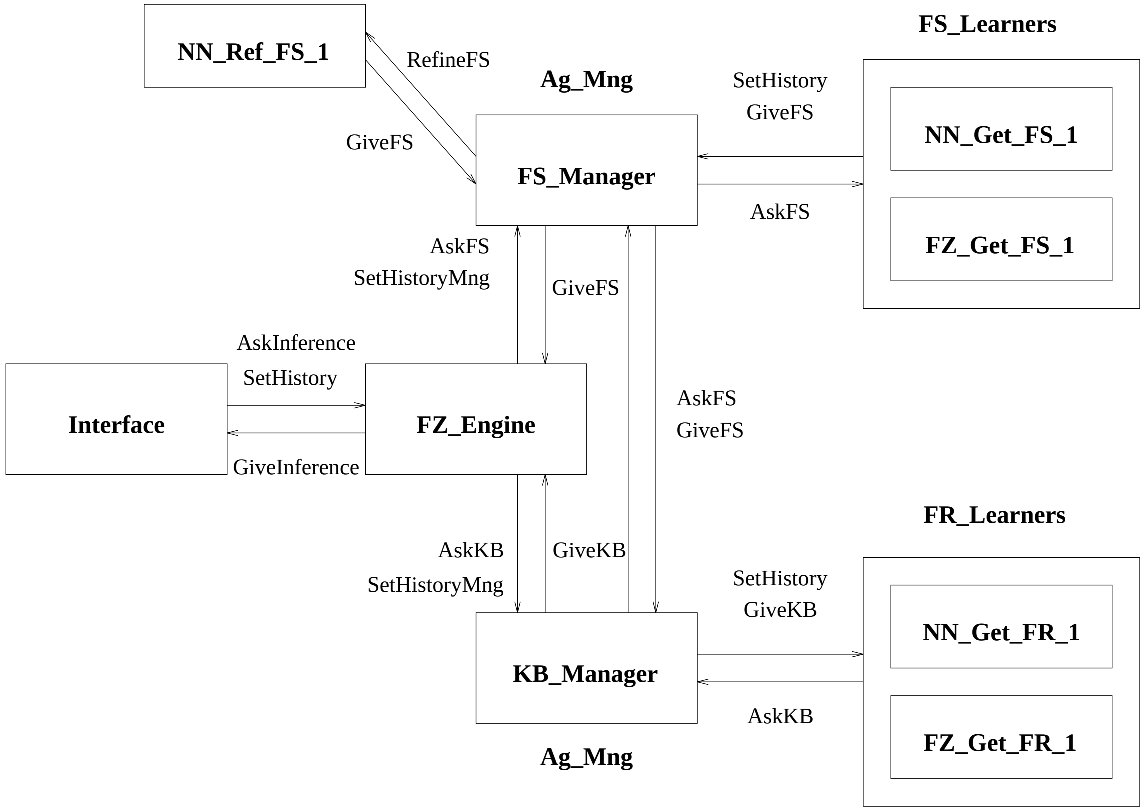

Figure 2: Event flow diagram of the implemented prototype

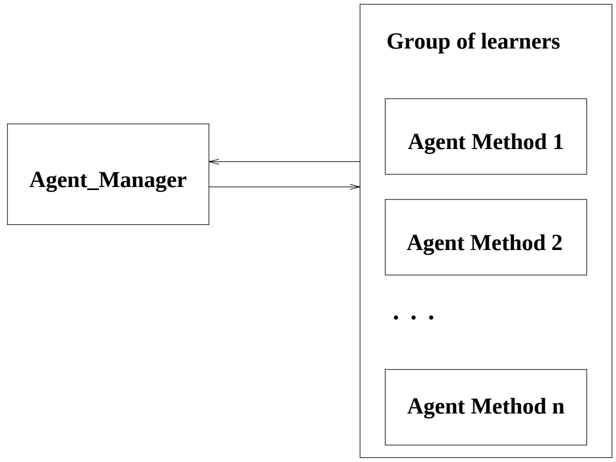

Figure 3: Intelligent standalone model

- the complete problem space is partitioned into a set of regions, each one with its own associated fuzzy knowledge base. The identification of regions is carried out by neural nets and by fuzzy clustering algorithms. Fuzzy rules are acquired inductively (in each region of the problem space) by neural nets and by fuzzy techniques, integrated in a hybrid system. Knowledge extraction could be done, in principle, in on-line or off-line mode. Only the off-line mode has been implemented.

- fuzzy sets are extracted inductively by neural nets and by fuzzy clustering algorithms, both methods being integrated in a hybrid system. Fuzzy sets extraction could be done, in principle, in on-line or off-line mode. Only the off-line mode has been implemented.

The hybrid systems that deal with fuzzy sets and fuzzy knowledge bases are, in their turn, built from an association/cooperative model: the so-called *intelligent stand-alone model*. The agents defined for this fuzzy-neural model will be reused for other multi-hybrid models (FL + CBR + MBR) in the different applications of the project.

The intelligent stand-alone model (figure 3) is a compound of an agent *manager* and a set of learning agents which are included into a public group. The *manager* selects the best system or knowledge base that has been learnt. It can also select the different regions on which the learning agents should work. Depending on the results obtained by the different learning agents, the *manager* can learn which agents work better under certain circumstances. So, this *manager* performs meta-processing for determining the regions (macro-states) on which the learning agents will work. The fuzzy-neural cooperative model consists of two instances of this basic intelligent stand-alone model, and a protocol is defined for communication between the *managers* of the different aspects to be optimized.

Table 1 shows a comparison among the different Fuzzy-Neural hybrid models in the literature in relation with the implementation of the tasks carried out by a generic fuzzy system. This generic fuzzy system has been analized using the CommonKADS methodology in [8] (figure 4). The last line in the table corresponds to the model implemented

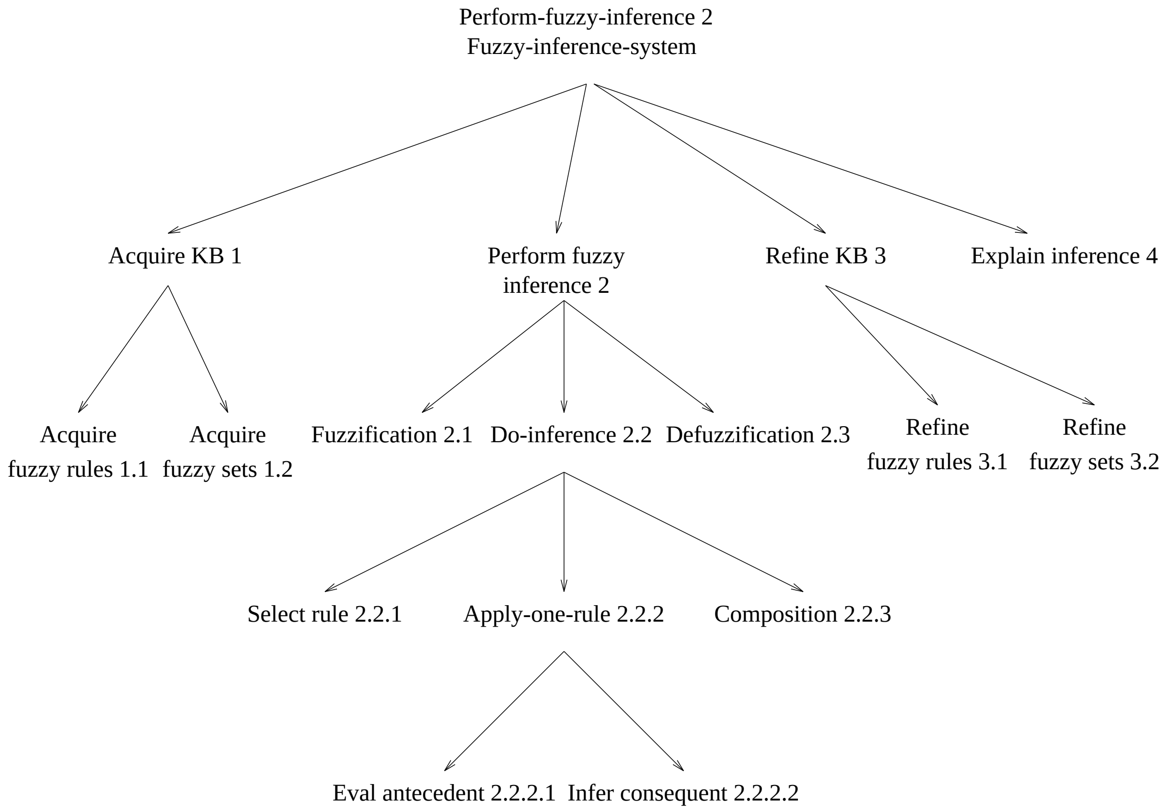

Figure 4: Complete task decomposition

and evaluated.

4 Development of the fuzzy-neural model

4.1 A general architecture for the development of hybrid systems

A framework suitable for developing intelligent hybrid systems should at least feature the following capabilities [3]:

- Modularity: Hybrid systems/models should be developed from basic heterogeneous building blocks.
- Encapsulation: These components should be encapsulated in order to offer homogeneous interfaces.
- Cooperation: Mechanisms suitable for intelligent cooperation should be supported to allow complex interactions among the components of the framework.
- Distribution: Components should be capable of working in distant environments.

Hybrid model	FZ/NN Integration	T1.1	T1.2	T2.1	T2.2	T2.3	T3.1	T3.2	T4
Unified		C	C	C	C	C	C	C	-/S
Stand-alone		C&S	C&S	C&S	C&S	C&S	C/-	C/-	-/S
Combination	Connectionist ES	S	S/C	C	C	C	C	C	-/S
(translational)	CS symb. acq.	S	S	C	C	C	C	C	-/S
	CS symb. desc.	C	C	C	C	C	C	C	S
	Connect. infer sys.	S	S	C	C	C	C	C	-/S
Combination	online FS tuning	S	C	S	S	S	-	C	-/S
(functional)	offline FS tuning	S	C	S	S	S	-	-	-/S
	online rules extr.	C	S	S	S	S	C	-	-/S
	offline rules extr.	C	S	S	S	S	-	-	-/S
Fusion	Logic CS	S/C	S/C	C	C	C	C	C	-/S
Association	Int. stand-alone	S&C	S&C	S&C	S&C	S&C	S&C	S&C	S&C

Table 1: Relationship taxonomies/task analysis

- Ease of use: The overload imposed on researchers to integrate particular pre-existing components in this framework and to organize the interaction of these components should be reduced to a minimum. In any case, such overload should be fully justified in terms of the inherent benefits of this approach.

- Openness: Facilities have to be foreseen for the inter-operation of these components with other distributed frameworks currently under development by companies and research institutions.

These considerations lead the authors to propose a multi-agent architecture as an adequate technology for building a common platform for the MIX project (*MIX: Modular Integration of Connectionist and Symbolic Processing in Knowledge-Based Systems*, European Information Technology Programme, project ESPRIT-9119). An overview of the project objectives and methodology can be found in [5], and a complete description of the MIX architecture in [6]. The use of the architecture for symbolic-connectionist integration is illustrated in [3].

Multi-agent architectures belong to the broader field of Distributed Artificial Intelligence. To summarize, we call agents autonomous entities capable of carrying out specific tasks by themselves or through cooperation with other agents. Multi-agent systems offer a decentralized model of control, use the mechanisms of message-passing for communication purposes and are usually implemented from an object-oriented perspective.

4.2 Implementation of the prototype

The neural model used in the implementation (a Kohonen's self-organizing map [12]) is described in the common pre-document of deliverables D2, D3 and D4. The Kohonen map generates as many codebook vectors as detected regions. For every codebook vector a fuzzy rule is extracted.

The fuzzy engine used has been FuzzyCLIPS 6.02A [11] with standard max-min algorithm for fuzzy inference and COG algorithm for defuzzification.

For rule extraction C4.5 algorithm has been used [18]. Although C4.5 is classification learning algorithm, it can be easily adapted to be a prediction algorithm by classifying in appropriate classes. Input data should be pre-processed to assign the accurate fuzzy label to every variable.

Fuzzy ISODATA algorithm has been selected to determine appropriate fuzzy sets and space regions [2]. This iterative algorithm classifies data according to their distance to a characteristic point of the cluster called centroid. The algorithm is based on the minimization of the cluster volume and maximization of its density.

FuzzyCLIPS [1] [FuzzyCLIPS] is an extended version of the CLIPS rule-based shell [2]. In addition to the CLIPS functionality, FuzzyCLIPS can deal with exact, fuzzy (inexact) and combined reasoning allowing fuzzy and normal terms to be freely mixed in the rules and facts of an expert system. The system uses two basic inexact concepts fuzziness and uncertainty.

In the current implementation we have used max-min as compositional rule of inference and COG (Centre of gravity algorithm) for defuzzification. For fuzzification, the crisp value is fuzzified as a triangular shape centred on the crisp value with zero possibility at value+delta and value-delta, with delta 0.001.

Trapezoidal fuzzy sets have been defined by means of a singleton representation. The definition of the fuzzy sets has been performed using fuzzy clustering and the extraction of fuzzy rules has been performed using C4.5 with pre-processed inputs and outputs as described in [7].

The Stuttgart Neural Nets Simulator (SNNS v3.0, v4.0 and v4.1) [21] was used for the implementation of pure connectionist systems.

5 Prototype evaluation

The MIX platform is currently being used to test different hybrid models for three different applications. The first one is the optimization of a motor/gear-box combination for a turbo-charged engine. The second one consists of the control of a roll-mill in a steel making company [9, 17]. The last application pertains to the medical domain: a monitoring system for an intensive care unit.

In the following, the developed cooperative fuzzy/neural prototype is evaluated on the roll-mill application. In this industrial process, the output of the intelligent system is the suggested rolling force to optimize the steel band production. Twelve parameters are used as the input of the force predictor: three discrete and nine continuous.

A data file with 11054 vectors (each vector with twelve inputs and the best output for them) has been used to train the system. 8866 vectors (around 80% data) were the train

[1] Copyright 1994 National Research Council Canada

[2] CLIPS was developed by the Artificial Intelligence Section, Lyndon B. Johnson Space Center, NASA and is available from COSMIC, The University of Georgia, 382 Broad Street, Athens, GA 30602, USA.

set, and the rest (2188), the test set. The value to optimize is the relative error. The experiments performed are shown in the following sections.

5.1 Pure Connectionist System with Unclassified Data

A simple neural net was trained with the complete training set. For this purpose, a fully connected feed-forward Multi-Layer Perceptron with four layers (12*12*6*1) was used. Learning was achieved by using the back propagation algorithm (1000 epochs and a learning rate of 0.2). The average error resulting from this experiment was 12.1%.

5.2 Pure Connectionist System with Clusters

Three of the 12 input values, N_PASS, LUBRICATION and ROLL_FINISH, are discrete, the first two having 3, and the last one having 21 different values. So, it was considered probable that these values had a significant meaning for the process. To find out about this learning and test data sets were divided into six classes with different combinations of the two first discrete variables (there was no sample with N_PASS=0.5). Then, a net with the same architecture was trained for each of these sets and the nets were tested with the test sets of the corresponding class. The average error obtained was 12.3% (slightly higher).

By examining the data graphically, it was seen that two of the input variables, WIDTH and THICKNESS, showed a higher correlation with the output variable than the rest. So, these two were selected as the basis for clustering. By using fuzzy ISODATA algorithm, the bi-dimensional problem space was divided into 7 classes. The fuzzy sets describing these 7 classes were translated into crisp-linear regions.

A net with the same architecture as before was trained for each of these classes and tested with the test data of the class. Results are shown in table 2 (column 4). The table shows that only the results for class 6 were better than with the initial network (column 3), but the average for these classes was worse.

5.3 Fuzzy sets extraction with fuzzy clustering

The ISODATA fuzzy clustering algorithm has been selected to define fuzzy sets for every continuous variable of the system. This iterative algorithm classifies data according to their distance to a characteristic point of the cluster called centroid. Centroids has been selected as the most representative peaks of the data histogram. The algorithm is based on the minimization of the sets volume and maximization of density [20, 2]. For instance, some of the fuzzy sets obtained by the algorithm are shown in figure 5.

5.4 Rule extraction with connectionist methods

Several attempts have been carried out with connectionist systems for rule extraction. Firstly, a public domain software was evaluated: *NEFCLASS* [16]. Although the tool worked properly with some standard data sets (e.g., the well known iris problem), it was unable of dealing with the roll-mill problem (due to implementation limitations).

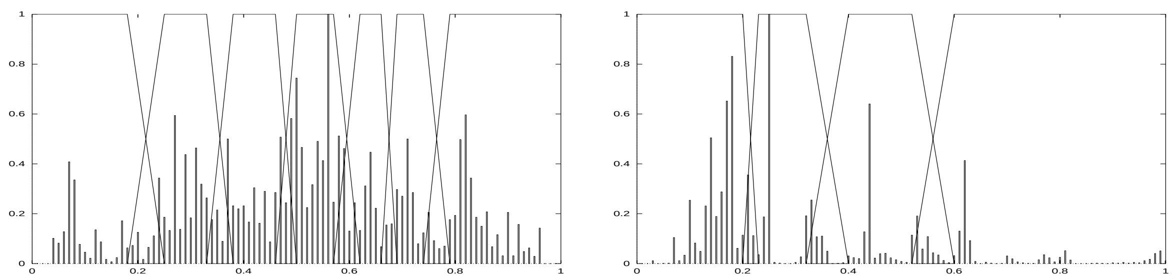

Figure 5: Var 2: WIDTH. Width of the band
Seven fuzzy sets (VeryLow, Low, NormalLow, Normal, NormalHigh, High and VeryHigh)
Var 3: THICKNESS. Thickness of the band
Four fuzzy sets (VeryLow, Low, High and VeryHigh)

Other experiments were carried out by Klaus Eder at Kratzer Automatisierung. These experiments include again the iris data set, the roll-mill application and the gear-box application. The network used in these cases for rule extraction was a Kohonen self-organizing map. However, the results obtained have been poor until now.

5.5 Rule extraction with symbolic methods

For rule extraction C4.5 algorithm has been used. First, historic numeric data file has to be transformed in a pseudo-fuzzy data file, by changing every number in the file by the label of the fuzzy set with bigger membership value. This data file feeds the C4.5 algorithm to create an Induction Decision Tree and a set of rules. C4.5 algorithm, that has been designed to classify data, can be used to create prediction systems (as in this case) if predicted values are discrete. In this case, predicted values are the fuzzy sets of the predicted variable. We have generated two knowledge bases of fuzzy rules. Each Knowledge base has a different confidence factor. This factor sets the pruning level in the decision tree (intermediate step to create rules) and rules generation. The used factors are 25 (default) and 50. Some obtained rules are shown in figure 6.

5.6 Analysis of the results

Table 2 shows the results obtained for the neural nets and fuzzy knowledge bases generated. The last column remarks the best value of the error rate for each cluster, with the global mean error for these cases. The most important consequences of these results are:

- We have obtained better results than with the pure neural approach. The error rate has been reduced to 50%.
- Our premise of getting better results dividing the data space in regions seems to be promising. More extensive work has to be done, however, to determine a better splitting of the problem space into regions by using both symbolic and connectionist methods.

$$
\begin{array}{llll}
IF & Var2 & \in & VeryHigh & AND \\
 & Var3 & \in & VeryLow & AND \\
 & Var5 & = & 0.00 & AND \\
 & Var12 & \in & NormalHigh & \Rightarrow & Var13 \in VeryHigh
\end{array}
$$

$$
\begin{array}{llll}
IF & Var2 & \in & NormalLow & AND \\
 & Var6 & \in & Normal & AND \\
 & Var8 & \in & VeryLow & AND \\
 & Var11 & \in & High & \Rightarrow & Var13 \in Low
\end{array}
$$

Figure 6: Some rules from the knowledge bases

Data	Neural Nets		Fuzzy K-Bs				**Best**
	Spec.	Gen.	S-25%	S-50%	G-25%	G-50%	
C1	14.1	13.8	9.48	10.19	8.60	**8.43**	8.43
C2	14.6	14.1	13.43	5.90	5.82	**5.12**	5.12
C3	12.1	11.1	**5.93**	8.41	8.25	7.03	5.93
C4	14.7	13.2	7.31	12.67	9.79	**6.50**	6.50
C5	11.1	10.8	7.49	5.47	5.86	**5.81**	5.81
C6	**8.7**	9.6	9.93	11.36	13.32	10.35	8.70
C7	11.1	10.0	13.64	13.10	11.48	**7.03**	7.03
Global	12.77	12.16	9.29	8.94	8.38	6.92	**6.49**

Table 2: Analysis of the results

- More extensive work is needed in the neural part for extracting fuzzy sets and rules.
- Fusion models need to be validated and integrated in our model.

6 Conclusions

It is still early to offer definite results of this work, still under development. However, some initial contributions and conclusions deserve mention here. Concerning hybrid systems, the main contributions are:

- A general purpose distributed architecture for the development of heterogeneous systems has been designed, leading to the implementation of a public domain software platform.
- The CommonKADS methodology has been adopted for developing hybrid systems. The methodology has been previously adapted for the development of multi-agent systems.

Regarding fuzzy-neural hybridization, the most salient results obtained until now are:

- An extensive review of the fuzzy-neural literature has been carried out, leading to a unified taxonomy for fuzzy-neural hybrid systems.
- The definition of an architecture for fuzzy-neural hybrid systems (the cooperative learning model). This model includes our intelligent stand-alone hybrid.
- The design and implementation of a hybrid fuzzy-neural model which is a first attempt towards an association model. This model has the additional advantage of being open, and it can be easily combined with other paradigms.
- The design and implementation of a set of reusable fuzzy and neural agents and services.

7 Acknowledgements

The MIX consortium is formed by the following institutions and companies: Institute National de Recherche en Informatique et en Automatique (INRIA–Lorraine/CRIN–CNRS, France), Centre Universitaire d'Informatique (Université de Genève, Switzerland), Institute d'Informatique et de Mathématiques Appliquées de Grenoble (France), Kratzer Automatisierung (Germany), Fakultät für Informatik (Technische Universität München, Germany) and Dep. Ingeniería de Sistemas Telemáticos (Universidad Politécnica de Madrid, Spain).

References

[1] Motohisa Funabashi and Akari Maeda. Fuzzy and neural hybrid expert systems: Synergetic ai. *IEEE Expert*, pages 32–40, August 1995.

[2] I. Gath and A. B. Geva. Unsupervised optimal fuzzy clustering. *Trans. on Pattern Analisys and Machine Intelligence*, PAMI-11(7):773–781, July 1989.

[3] José C. González, Juan R. Velasco, and Carlos A. Iglesias. A distributed platform for symbolic–connectionist interoperation. In Ron Sun and Frederic Alexandre, editors, *Proceedings of the IJCAI'95 Workshop on Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, pages 94–99, Montreal, Canada, August 1995. International Joint Conference on Artificial Intelligence. (An extended version of this paper will appear in *Connectionist Symbolic Integration*, Lawrence Erlbaum Associates, 1966.).

[4] Melanie Hilario. An overview of strategies for neurosymbolic integration. In Ron and Alexandre [19], pages 1–6.

[5] Melanie Hilario, Christian Pellegrini, and Frédéric Alexandre. Modular integration of connectionist and symbolic processing in knowledge-based systems. In *Proceedings of*

the International Symposium on Integrating Knowledge and Neural Heuristics, pages 123–132, Pensacola, Florida, May 1994.

[6] Carlos A. Iglesias, José C. González, and Juan R. Velasco. MIX: A general purpose multiagent architecture. In M. Wooldridge, K. Fischer, P. Gmytrasiewicz, N.R. Jennings, J.P. M:uller, and M. Tambe, editors, *INTELLIGENT AGENTS II: Agent Theories, Architectures, and Languages*, volume 1037 of *Lecture Notes in Artificial Intelligence*, pages 251–266. Springer-Verlag, 1996. (An earlier version of this work was published in the *Proceedings of the IJCAI'95 Workshop on Agent Theories, Architectures and Languages*, Montréal, Canada, August, 1995, pages 216–224.).

[7] Carlos A. Iglesias, José C. González, and Juan R. Velasco. Prototype of a fuzzy-neural hybrid model. evaluation report. Doc. Id. MIX/WP2/UPM/3.0, March 1996.

[8] Carlos A. Iglesias, José C. González, Juan R. Velasco, and Klaus Eder. Prototype of a fuzzy-neural hybrid model. implementation report. Technical Report MIX/WP2/UPM/2.0, Dep. Ingeniería de Sistemas Telemáticos, E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid, January 1996.

[9] Report on definition and specification of industrial applications: Steel rollmill application. Technical Report MIX/WP3/INRIA/1.0, INRIA Lorraine, March 1995.

[10] Abraham Kandel and Gideon Langholz, editors. *Hybrid Architectures for Intelligent Systems*. CRC Press, 1992.

[11] Knowledge Systems Laboratory - Institute for Information Technology - National Research Council Canada. *FuzzyCLIPS Version 6.02A - User's Guide*, September 1994. http://ai.iit.nrc.ca/fuzzy/fuzzy.html.

[12] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, third edition, 1989.

[13] Luis Magdalena. A first approach to a taxonomy of fuzzy–neural systems. In Ron Sun and Frederic Alexandre, editors, *Proceedings of the IJCAI'95 Workshop on Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, pages 112–118, Montreal, Canada, August 1995. International Joint Conference on Artificial Intelligence.

[14] Larry R. Medsker. *Hybrid Intelligent Systems*. Kluwer Academic Publishers, 1995.

[15] Detlef Nauck and Rudolf Kruse. Choosing appropiate neuro-fuzzy models. In *European Congress on Fuzzy and Intelligent Technologies*, pages 552–557, Aachen, September 1994.

[16] Detlef Nauck and Rudolf Kruse. NEFCLASS - a neuro fuzzy approach for the classification of data. In *Symposium on Applied Computing during 1995 ACM Computing Week*, Nashville, March 1995.

[17] Nicolas Pican, Frédéric Alexandre, and Patrick Bresson. Artificial neural networks for the presetting of a steel temper mill. *IEEE Expert*, 11(1):22–27, February 1996.

[18] J.R. Quinlan. *C4.5, Programs for Machine Learning.* Morgan Kaufmann, 1993.

[19] Sun Ron and Frederic Alexandre, editors. *Connectionist-Symbolic Integration: From Unified to Hybrid Approaches.* IJCAI 95 Workshop, August 1995.

[20] Juan R. Velasco and F. Enrique Ventero. Some applications of fuzzy clustering to fuzzy control systems. In *Proceedings of the Joint Conference on Information Sciences*, pages 363–366, Pinehurst, NC, USA, November 1994.

[21] Andreas Zell, and others. *SNNS (Stuttgart Neural Network Simulator) - User Manual v. 4.1.* University of Stuttgart, Institute for Parallel and Distributed High Performance Systems (IPVR). Applied Computer Science / Image Understanding, November 1995. http://vasarely.informatik.uni-stuttgart.de/snns/snns.html.