Distributed Embedded Intelligence Room with Multi-agent Cooperative Learning

Kevin I-Kai Wang, Waleed H. Abdulla, and Zoran Salcic

Department of Electrical and Computer Engineering, University of Auckland. Private Bag 92019, Auckland, New Zealand
{kevin.wang, w.abdulla, z.salcic}@auckland.ac.nz

Abstract. In this paper, a novel Multi-agent control system with fuzzy inference learning and its physical testbed are presented. In the Multi-agent system, distributed controlling, monitoring and cooperative learning are achieved through ubiquitous computing paradigm. The physical testbed named Distributed Embedded Intelligence Room (DEIR) is equipped with a fair amount of embedded devices interconnected in three types of physical networks, namely LonWorks network, RS-485 network and IP network. The changes of environment states and user actions are recorded by software agents and are processed by fuzzy inference learning algorithm to form fuzzy rules that capture user behaviour. With these rules, fuzzy logic controllers can perform user preferred control actions. Comparative analysis shows our control system has achieved noticeable improvement in control accuracy compared to the other offline control system.

1 Introduction

Ubiquitous computing is a new paradigm of information processing technology where the computation is carried out by a group of invisible embedded devices rather than by visible computers. By assuming ubiquitous computing in the background, ambient intelligence aims to achieve a system that is aware of the environment context, able to model and adapt to user's behaviour and should respond on user's behalf [1], [2].

In this paper, a novel Multi-agent (MA) control system with cooperative fuzzy inference learning is proposed. In the proposed system, massive amount of embedded devices are connected via different physical networks. A middleware layer above the physical network provides a unique control interface for the high level MA control system to all the physical networks. In the MA control system, software agents are distributed in nature, each monitoring and controlling its own embedded device. The Multi-agent Fuzzy Inference System (MAFIS) is developed to capture and model user's daily activities and to perform sensible control actions on user's behalf. In addition to the software based MA control system, physical testbed named Distributed Embedded Intelligence Room (DEIR) built in the University of Auckland (UoA) is also introduced.

The rest of the paper is organised as follows. Section 2 gives a brief description of our target environment and related researches. Section 3 details our physical testbed, DEIR, including the implementation of physical network infrastructure, middleware

J. Ma et al. (Eds.): UIC 2006, LNCS 4159, pp. 147 – 156, 2006.

and the MA control system. Section 4 explains the process of fuzzy inference learning of MAFIS. Section 5 discusses the comparative analysis carried out between MAFIS and the other offline control system. Section 6 indicates some future works within our project and section 7 concludes the paper.

2 Target Intelligent Environment

2.1 SOHO Environment

The target Intelligent Environment (IE) in our research is the so-called "small office, home office" (SOHO) environment. In such kind of environment, working, living and entertaining functionalities are fully integrated. Typical devices that could be in such environments include TV, Hi-Fi audio system, mobile devices such as cell phone or PDA, air conditioning system, light control system and a variety of sensors and actuators.

In order to model human activities in such environments and to test the developed control system in a realistic way, a physical testbed is essential. In the UoA, a physical testbed called DEIR has been constructed by the Embedded Systems Research Group (ESRG). This physical testbed is equipped with a number of sensors including light intensity, temperature, pressure, smoke, and occupancy sensors to monitor the environment states. It also contains a number of actuators for automating windows, blinds, dimmable lights and house appliances. These devices can be controlled via traditional switch interface, remote control, and PC interface. Users need not to be aware of the existence of vast number of embedded devices. DEIR forms a comprehensive testbed which allows various types of experiments such as human activity analysis, control system verification, and remote monitoring and controlling. More information on physical network, middleware and control system will be given in section 3.

2.2 Related Works

The MIT AI Lab started IE researches around mid 90s [3]. At that time, their research focus was to introduce intelligence via smart sensors and camera networks and can be considered as Human-Computer Interaction (HCI) and sensors network research. In 1999, a MAS called Metaglue which had no built-in intelligence, was developed in that lab to control the IE [4]. However, in the past few years, intelligent knowledge-based resource management [5] and reactive behaviour systems [6] had been developed and integrated into the MAS to introduce intelligence.

The University of Essex is also one of the pioneers in this research field. Their research focuses on online learning of personalised behaviour which is inline with our research [7]. The core learning process of the proposed MAFIS is developed based on their Adaptive Online Fuzzy Inference System (AOFIS) [8].

The Adaptive Building Intelligence (ABI) project collaborated by several Swiss universities uses MAS approach as well. This project is different to ours in the way that it is aimed at providing intelligent building services rather than intelligent living spaces [9].

There are also many other research efforts such as the Microsoft Smart House [10], IBM BlueSpace [11] and MASSIHN project [12]. However, most of them focus on integrating working behaviour and device automation into the control system. These control systems neither capture or model the human behaviour nor adapt to human needs, and do not reveal the true meaning of ambient intelligence.

3 Distributed Embedded Intelligence Room (DEIR) Architecture

In this section, DEIR architecture will be discussed in three components. The first component consists of a collection of physical device networks. The second component includes the corresponding control software for each device network and the middleware layer. The third component is the MA control system with cooperative fuzzy inference learning. The conceptual system architecture can be depicted in Fig. 1.

3.1 Physical Network Infrastructure

In the current version of DEIR, three types of device networks are implemented, namely the IP network, LonWorks network [13] and RS-485 network. In general, LonWorks network is mainly used for connecting embedded sensors whereas RS-485 network is for controlling automatic devices such as windows, blinds and lights. In a higher level, IP network is responsible for controlling IP-based network devices such as IP-cam.

In LonWorks network, an iLon 100 router is used as a hardware interface between the control software and embedded sensors. RS-485 network uses a combination of smart switches and a hardware gateway server to connect a group of automatic devices with the control software. Each smart switch has 2 microcontrollers, Motorola HC11 and PIC, and a infra-red receiver integrated in it which allows up to 3 devices to be controlled via traditional switch interface or infra-red remote control interface. All the switches are connected with the gateway server which in turn communicates with the control software. By using the control software, PC control interface is also possible.

Similar to the other two device networks, multiple IP-based network devices can be connected to the system using one or more network routers or switches depending on the system topology. However, as shown Fig. 1, the IP network is at a higher level in the system architecture. Therefore, IP network is expected to provide other functionalities. By adding another layer of middleware, different low level device networks can be mapped onto the IP network, and hence the high level control system can treat the whole physical network in IP network as one entity. Further, IP network also has the flexibility of integrating other advanced wireless technologies such as Wi-Fi, Zigbee and Bluetooth.

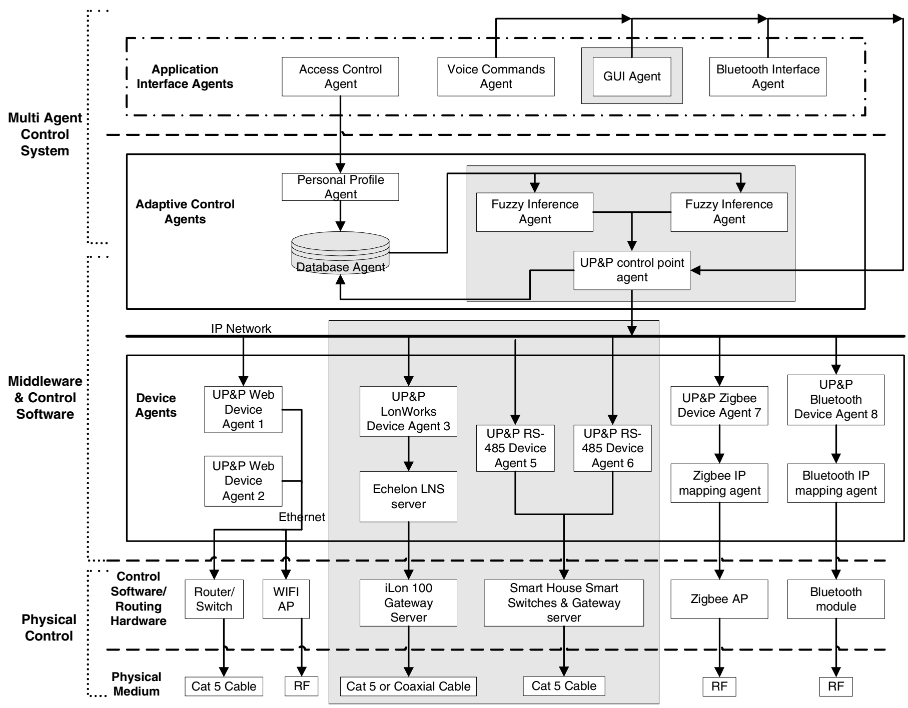

Fig. 1. DEIR control system architecture

3.2 Middleware and Device Control Software

In DEIR, Universal Plug and Play (UPnP) is used as the middleware. In general, UPnP is an IP-based device discovery protocol which enables automatic device discovery, configuration and management. The implementation of UPnP protocol can be considered in two components: the control point and software devices. UPnP control point is the server component which keeps all the information of registered software devices and acts as a software interface between high level MA control system and software devices. UPnP software device is the client component which links the control point with the corresponding device control software. There is a one to one relationship between UPnP software devices and physical devices. The UPnP protocol is incorporated in DIER using CyberGarage UPnP Java API.

LonWorks Network Operating System (LNS) is the control software which controls all the embedded sensors via the iLon 100 router. The corresponding UPnP software devices are implemented using the Java API provided by the LNS developer's kit. On the other hand, the UPnP software devices for RS-485 network devices are implemented in a slightly different way. As RS-485 control software is simply a software interface which converts the high level control commands into RS-485 network commands, this conversion can be integrated directly into the UPnP software devices and the control software is neglected in the actual implementation. The difference in LonWorks and RS-485 networks can be seen in Fig. 1.

3.3 Multi-Agent Control System

According to the functional requirements of an IE defined by Information Society Technologies Advisory Group (ISTAG), a true ambient intelligence should be able to model the environment and human behaviour, to interact with the user, and to provide secure and quality services [1], [2]. While meeting these functional requirements, one must not forget the real ambient intelligence is achieved through ubiquitous computing paradigm. This means, the environment consists of a large number of embedded devices which are smaller in size and have limited computational power. Therefore, the control system must be a distributed system and MAS offers a good solution to this application.

There had been a number of reliable, widely used and tested agent platforms in the past few years. Therefore, rather than developing one of our own MAS for DEIR, an existing MAS, JADE (Java Agent DEvelopment Framework), is used [14]. Agents of JADE are developed using the provided Java API. This allows UPnP components to be incorporated in JADE platform as low level device agents (refer to Fig. 1), which makes the system architecture much more coherent. Different to the device discovery ability that comes with the UPnP protocol, JADE has its own device discovery routine which enables direct communication between any pair of agents in the runtime environment. This extends the capability of UPnP which only allows communications between software devices and control point. With the ability of direct communication, agents can exchange information and achieve cooperative learning and controlling.

4 Multi-Agent Fuzzy Inference System

Multi-Agent Fuzzy Inference System (MAFIS) is implemented in DEIR. The learning technique used in MAFIS is adopted from the Adaptive Online Fuzzy Inference System (AOFIS) developed in the University of Essex [8]. AOFIS uses an unsupervised, data-driven, one pass fuzzy inference learning. The fuzzy rule bases are learned based on captured user's activities and corresponding environmental states. Despite the learning process of MAFIS and AOFIS is the same, the life-long online adaptation ability is yet implemented in MAFIS. The following subsections provide more detailed descriptions for each step of the learning process in sequence.

4.1 Double Clustering Algorithm

Double clustering algorithm [15] is a combination of Fuzzy C-Mean (FCM) algorithm and agglomerative hierarchical clustering algorithm. It takes recorded numerical data (i.e. the environment states and user actions) and converts the data into fuzzy granules which can be represented by fuzzy sets. FCM algorithm [16] is a multi-dimensional clustering process which generates a predefined number of P clusters according to the geographical closeness between the data instances through an iterative process. With r input/output features, each of the P cluster centre is a r-dimensional vector $\bar{c}_i = (c_{i1}, c_{i2}, \cdots c_{ir})$, consists of the one dimensional centres of each feature.

Different to FCM algorithm, agglomerative hierarchical clustering [17] is a one dimensional clustering process. Based on the results of FCM, there are P one dimensional centres, usually called prototypes, for each input/output feature. The prototypes of each feature are grouped into a sorted sequence from minimum to maximum. The number of prototypes in each sequence should equal to the number of fuzzy sets to be derived in the next step. In order to reduce the number of prototypes, two consecutive prototypes which are closest in value are replaced by their mean value until the required number of prototypes is reached.

4.2 Generation of Membership Function

In order to generate interpretable fuzzy sets, the prototypes of each feature need to be quantified into membership functions. Gaussian membership function is used to quantify the prototypes and to interpret the fuzzy sets of each input and output features. The Gaussian membership function is generated based on the centre and spread, which can be worked out using the prototypes formed in the previous step [8]. Notice that, however, the membership values of the boundary fuzzy sets are fixed to 1 so the membership function can be extended indefinitely beyond their centres.

4.3 Fuzzy Rule Extraction

In order to extract the fuzzy rule base from the previously defined membership functions and captured data instances, the enhanced version of Mendel Wang (MW) method [18], [19] is used. This approach generates fuzzy rules that map multiple input features to multiple output features. Assuming there are N data instances for n input features and k output features and K derived fuzzy sets for each feature. The first step is to run through all the data instances, t, in the dataset and compute the Gaussian membership values $\mu_{A_s^q}\left(x_s^{(t)}\right)$ for each membership function q = 1, 2, ….., K, and for each input feature x_s, s=1, 2, …, n. Then find a particular membership function $q^* \in \{1, \cdots, K\}$ which outputs the maximum membership values among all the other membership functions for each data instance as in Eq. (1):

$$\mu_{A_s^{q^*}}\left(x_s^{(t)}\right) \geq \mu_{A_s^q}\left(x_s^{(t)}\right) . \tag{1}$$

The fuzzy set $A_s^{q^*}$ that outputs maximum membership value is assigned to the particular input x_s . In the end, there should be N rules generated, one for each data instance. However, a lot of rules generated will have the same antecedent parts but possibly different consequent parts. This forms a group of conflict rules. To resolve these conflicts, the weight $w(t)$ of a rule, which measures the degree of membership of the input instance to the fuzzy region covered by the rule, is calculated as:

$$w^{(t)} = \prod_{s=1}^{n} \mu_{A_s^q}(x_s(t)) \ . \tag{2}$$

Assuming there is a group of M conflict rules, which the weight average of all the rules in the conflict group can be calculated using the weight value calculated in Eq. (2), the mathematical expression of the weight average is shown as follows:

$$av = \frac{\sum_{u=1}^{M} y^{(t_u)} w^{(t_u)}}{\sum_{u=1}^{M} w^{(t_u)}} . \tag{3}$$

where u is the index of the conflict rules, M is the number of the conflict rules and t_u is the index of data instance corresponding to the conflict rule u.

Based on the weight average, the consequent parts of the rule can be evaluated to resolve the conflicts and generate a final set of fuzzy rules. Among the possible K output fuzzy sets $B^1, \cdots, B^K$, find a B^* such that,

$$\mu_{B^*}(av) \geq \mu_{B^K}(av), \quad * \in K \ . \tag{4}$$

The process of calculating weight average and finding the output fuzzy set that generates the maximum membership value of the weight average is carried out for each output feature to cope with rules of multiple output features. The final fuzzy rule derived would be in form of Eq. (5).

$$\begin{gathered} \text{If } x_1 \text{ is } A_1^{(l)} \text{ and ..... and } x_n \text{ is } A_n^{(l)} \\ \text{Then } y_1 \text{ is } B_1^{(l)} \text{ and .... and } y_k \text{ is } B_k^{(l)} \ . \end{gathered} \tag{5}$$

where n is the index of input features, k is the index of output features and l is the index of the fuzzy rule.

Once the fuzzy inference agents have generated membership functions and fuzzy rule bases, Fuzzy Logic Controller (FLC) agents are ready to control the environment on user's behalf using the learned fuzzy rules. In MAFIS, singleton fuzzification, max-product composition, product implication and height defuzzification are deployed in the FLCs [8].

5 Results and Discussions

To examine the control accuracy of MAFIS, a comparative analysis with its centralised version, AOFIS, has been performed. Refer to Fig. 2, the Scaled Root Mean Square Error (SRMSE) is used to measure the control accuracy. The traditional RMSE is scaled to take into consideration the different output ranges. For example, dimmable lights can output values range from 0 to 100 according to its intensity whereas non-dimmable lights output 0 to 1 according to switch status. In order to conduct a fair comparison, the same dataset is used to evaluate the performance of MAFIS and AOFIS. The dataset used for the analysis contains 7 input features, namely internal light sensor, external light sensor, internal temperature sensor, external temperature sensor, chair pressure sensor, bed pressure sensor and time; and 10 output features including 4 dimmable lights, blinds, desk light, bed light, heater and two PC applications: MS Word and MS Media Player. This particular dataset contains 408 data instances collected over 3 consecutive days monitoring real user activities. The data instances are split into 272 data instances in the training set and 136 data instances in the testing set. Different to AOFIS which models the relationship between all the inputs and outputs, MAFIS associates relevant inputs and outputs into separate groups and models each group separately. Refer to Fig. 3, 4 device groups are defined. Each device agent contributes its own data to the others in the group to achieve cooperative learning. These four groups are modelled by four independent fuzzy inference agents at the same time and each device group can be modelled with optimised number of fuzzy sets. As shown in Fig. 2, MAFIS achieves 5% reduction in overall control errors comparing to AOFIS.

Scaled Root Mean Squared Error (SRMSE)					
	MAFIS				AOFIS
No. of Fuzzy Sets	Group 1 SRMSE	Group 2 SRMSE	Group 3 SRMSE	Group 4 SRMSE	SRMSE
2	0.6726	0.7963	0.6608	0.5415	0.2148
3	0.2102	0.2406	0.3157	0.1390	0.1476
4	0.1798	0.2057	0.3127	0.1094	0.1461
5	0.1389	0.1775	0.2687	0.0819	0.1364
6	0.1204	0.1705	0.2199	0.0853	0.1352
7	0.0979	0.1739	0.2220	0.1047	0.1261
8	0.0931	0.1911	0.1536	0.0735	0.1326
9	0.0893	0.1496	0.1249	0.0665	0.1472
10	0.0974	0.1354	0.1004	0.0652	0.1537
11	0.0835	0.1290	0.0972	0.0697	0.1696
12	0.0865	0.1335	0.1330	0.0735	0.1999
13	0.0716	0.1126	0.1566	0.0912	0.2246
14	0.0731	0.1140	0.1404	0.0866	0.2337
15	0.0742	0.0976	0.0735	0.0735	0.246
16	0.0807	0.1080	0.0891	0.0547	0.2459
17	0.0792	0.1154	0.0976	0.0773	0.2732
18	0.0780	0.1258	0.0857	0.0676	0.2747
19	0.0900	0.1080	0.0819	0.0971	0.2771
20	0.0783	0.1349	0.0949	0.0488	0.2839
Optimised SRMSE for MAFIS			0.0729		

Fig. 2. Comparison of 2 offline control systems

Group 1 (9 features)		Group 2 (9 features)	
Input set	**Output set**	**Input set**	**Output set**
Int. Light Sensor	Dimmable Light 1	Int. Light Sensor	MS Word
Ext. Light Sensor	Dimmable Light 2	Ext. Light Sensor	MS Media
Chair Pressure	Dimmable Light 3	Int. Temp. Sensor	
Bed Pressure	Dimmable Light 4	Ext. Temp. Sensor	
Time		Chair Pressure	
		Bed Pressure	
		Time	
Group 3 (6 features)		**Group 4 (7 features)**	
Input set	**Output set**	**Input set**	**Output set**
Int. Light Sensor	Blind	Int. Light Sensor	Bed Light
Ext. Light Sensor	Heater	Ext. Light Sensor	Desk Light
Int. Temp. Sensor		Chair Pressure	
Ext. Temp. Sensor		Bed Pressure	
		Time	

Fig. 3. Relevant input/output devices grouping

6 Future Works

With the proposed control system architecture, future work can be carried out in both control and object level. In control level, online adaptation characteristic can be integrated into MAFIS to achieve a life long learning control system which provides more satisfactory services by adapting itself according to the changes of user behaviour. Also, different machine learning techniques can be mixed in the MA control system to handle different groups of devices to achieve a better performance.

In terms of the object level, rather than having predefined groups of relevant input and output devices, automatic grouping of relevant devices should be implemented. As the system complexity grows with increasing number of embedded devices, it is not always possible for human to predefine optimised device groups. In addition, UPnP software devices can be integrated into the smart switches to achieve real plug and play of those known devices.

7 Conclusions

In this paper, a novel Multi-agent Fuzzy Inference System (MAFIS) and its physical testbed, DEIR, are presented. DEIR consists of a fair amount of embedded devices. Embedded devices are interconnected and have limited computational powers which make DEIR a true ubiquitous computing testbed. MAFIS takes the advantage of ubiquitous computing to achieve cooperative learning and ubiquitous intelligence. Multiple groups of relevant input and output devices are monitored by separate fuzzy inference agents in parallel to model the user activities and to perform control actions on user's behalf. MAFIS and DEIR are well integrated to meet the requirements of ambient intelligence such as the ability to understand environmental context, to model human behaviour and to make sensible controls on human's behalf. The comparative analysis shows that our system has achieved a notable improvement in control accuracy compare to the centralised control system, AOFIS. In our future work, life-long cooperative learning with mixed machine learning techniques and automatic device configuration abilities are targeted.

Acknowledgement. This research is supported by UARC research grant 3604552 and top achiever doctoral scholarship. The authors would like to thank Faiyaz Doctor and Prof. Victor Callaghan for their kind contribution of providing the dataset for comparative analysis and various helps regarding to the use of their AOFIS learning technique.

References

1. Ducatel, K., Bogdanowicz, M., Scapolo, F., Burgelman, J.-C.: Scenarios for Ambient Intelligence in 2010. Information Soc. Technol., Advisory Group (ISTAG), Inst. Prospective Technol. Studies (IPTS), Seville (2001)
2. Riva, G., Loreti, P., Lunghi, M., Davide, F.: Presence 2010: The Emergence of Ambient Intelligence. Amsterdam, The Netherlands:Ios Press (2003)
3. Brooks, R. A.: The Intelligent Room Project. In: Second International Conference on Cognitive Technology, (1997) 271-278
4. Philips, B. A.: Metaglue: A Programming Language for Multi-Agent Systems. M.Eng. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (1999)
5. Gajos, K.: A Knowledge-Based Resource Management System For The Intelligent Room. M.Eng. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2000)
6. Kulkarni, A. A.: A Reactive Behavioral System for the Intelligent Room. M.Eng. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2002)
7. Hagras, H., Colley, M., Callaghan, V., Clarke, G., Duman, H., Holmes, A.: A Fuzzy Incremental Synchronous Learning Technique for Embedded-Agents Learning and Control in Intelligent Inhabited Environments. In: Proc. of the 2002 IEEE Int. Conf. on Fuzzy Syst., (2002) 139-144
8. Doctor, F., Hagras, H., Callaghan, V.: A Fuzzy Embedded Agent-Based Approach for Realizing Ambient Intelligence in Intelligent Inhabited Environment. In: IEEE Trans. Sys. Man Cybern., vol. 35, no. 1, (2005) 55 – 65

9. Rutishauser, U., Schaefer, A.: Adaptive Building Intelligence: A multi-Agent approach. Diploma thesis, University of Applied Science Rapperswil, Switzerland and Institute of Neuroinformatics, Swiss Federal Institute of Technology and University of Zurich , Switzerland (2002)
10. Brumitt, B., Cadiz, J. J.: Let There Be Light! Comparing Interfaces for Homes of the Future. Microsoft Research, Redmond, WA 98052, MSR-TR-2000-92 (2000)
11. Yoshihama, S., Chou, P., Wong, D.: Managing Behavior of Intelligent Environments. In: Proc. of the First IEEE Int. Conf. on Pervasive Comp. and Communications, (2003) 330-337
12. Tsai, C. F., Wu, H. C.: MASSIHN: A Multi-Agent Architecture for Intelligent Home Network Service. In: IEEE Trans. on Consumer Electronics, vol. 46, (2002) 505-514
13. Echelon Corporation, “LonWorks Overview, ” February 2006, http://www.echelon.com/solutions/overview/default.htm
14. Wang, K. I., Abdulla, W. H., Salcic, Z.: A Multi-Agent System for Intelligent Environments using JADE. In: IEE Int. Workshop on Intell. Environ., (2005) 86-91.
15. Castellano, G., Fanelli, A. M., Mencar, C.: Generation of interpretable fuzzy granules by a double clustering technique. In: Arch. Contr. Sci., vol. 12, no. 4, (2002) 397-410
16. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum (1981)
17. Jalin, A. K., Dubes, R. C.: Algorithms for Clustering Data, Englewood Cliffs, NJ: Prentice Hall (1998)
18. Wang, L. X.: The MW method completed: A flexible system approach to data minig. In: IEEE Trans. Fuzzy Syst., vol. 11, no. 6, (2003) 678-782
19. Wang, L. X., Mendel, J. M.: Generating fuzzy rule by learning from examples. In: IEEE Trans. Syst. Man Cybern., vol. 22, no. 6, (1992) 1414-1427