# Distributed Intelligence for Smart Home Appliances

Cecilio Angulo[1] and Ricardo Téllez[2]

[1] GREC, Knowledge Engineering Research Group,
Universitat Politècnica de Catalunya, 08800 Vilanova i la Geltrú, Spain
r_tellez@ouroboros.org,
[2] ERIC, Engineering & Research in Computational Intelligence,
Universitat Politècnica de Catalunya, 08004 Barcelona, Spain,
cecilio.angulo@upc.es

**Abstract.** A main goal for the researchers designing smart health care technology is to develop strategies allowing both, early detection of and avoid problems that could lead to a decreased independence. Medical analysis, smart sensors, intelligent software agents, distributed control, wireless communication and internet resources are research areas implied in home automation delivering intelligent health care. Ubiquitous computing is the most promising technological approach that can meet this challenge if their non-intrusive and adaptable objectives are reached, meanwhile it is able to maintain privacy. Our work is concerned with the development of a smart home architecture allowing to integrate information from a wide variety of sensors and actuators: information recruited for these elements is processed into microprocessors implementing computational intelligence techniques; cooperative communication between units is implemented through a wireless net into the home; and internet resources allow to link the home with external services. Within this general infrastructure, the proposed architecture expands from the conversion of passive sensors into smart devices (we call them "intelligent hardware units") by adding processing capabilities and using them like plug-and-play mechanisms. Performed experiments on a robotic pet platform show the viability of the present approach and they appoint interesting results to be applied on the entire home.

## 1 Introduction

The domestic environment is predicted by market analysts to be the major growth area over the next decade and a prime site for the development of ubiquitous computing [1]. For example, people aged 65 and older are the fastest growing segment of the US population; by 2030 over 4 million Americans will be over age 85 [2], and 69.4 million over age 65. Furthermore, over 20% of people 85 and over have a limited capacity for independent living [3], with the result that they require continuous monitoring and daily care. The creation of a secure, unobtrusive, adaptable environment for monitoring health and encouraging healthy behavior will be vital to the delivery of health care in the future.

The framework of our work is “smart home technology”, i.e. systems that have sensors and actuators that monitor the occupants, communicate with each other, and intelligently support the occupants in their daily activities [4]. For the continuous monitoring in the home environment unobtrusive and inexpensive sensors must be deployed. Meanwhile these sensors are inherently noisy and unreliable, robustness is usually achieved by the deployment of a large number of sensors. In our opinion, a more flexible and adaptable option is to integrate models and algorithms of computational intelligence processing observed sensor data in order to link them with human behaviors. In this sense, multi-agent systems [4] and architectures based on smart devices [5] have recently been explored as monitoring health care systems. Although some good results are provided from these approaches and interesting lessons have been learnt about reliability and scalability of the architecture, most desirable features as adaptability and learning from the user have not been derived.

Some efforts have been dedicated to the multi-agent, internet link and wireless communication research areas in home automation. Separately, individual specific smart sensors with or without communication capacities have been proposed and designed. However, dedicated distributed architectures on smart sensor nodes [6] is yet a research area to be developed. Our approach is to consider all these devices from an automation perspective [7], and integrate them on a multi-agent system environment to obtain the valuable features of this former technology. Hence, sensor and actuator devices are the key to obtain adaptability, interfacing between the real world of the user and the machine world of the software agents. Physical agents or “intelligent hardware units” are created by embedding flexible computing techniques into the sensors and the actuators, and communication abilities to share information.

In the home of the future, groups of devices should have enough collective awareness to function autonomously based on sensor data. Collective intelligence technology will be essential to analyze data from these distributed sensors. Research through this article focus on achieving the adaptation of soft-computing algorithms, developed usually as software modules in conventional computers, being implemented into specific hardware to obtain adaptation to the users. The most important aim of this vision is to design a collaborative computing structure that merges the intelligent hardware units with the needed information processing in order to generate a friendly operating scene through appropriate user interfaces. This new control structure, just the opposite of the classic hierarchical implementation, aims to become a decisive stage in the integration of the autonomous actuation of intelligent software agents into hardware elements (“intelligent hardware agents”).

The document is organized as follows. In the next Section the proposed distributed architecture is explained by designing a multi-agent system through the cooperation of intelligent hardware units with neuroprocessing capabilities and learning by co-evolution. Performance of the novel architecture is illustrated in Section 3 by three different experiments, and finally some concluding remarks are presented.

## 2 The Architecture

Size and complexity of computing and communication interfaces in every-day electronic applications is speedily growing due to the accessibility to this technology for a increasing number of users. There currently exists a certain tendency to an excessive centralization on a ‘master’ unit when software is designed, so the functionality of the elements is not exploited in an optimal form. Meanwhile, a number of hardware devices, sensors or actuators, are not as smart as it could be possible, because the information recruited will overflow the processing capabilities of the central control element.

Smart home requires a more robust set of features that standard home automation nowadays provide, including advanced processing algorithms, distributed intelligence, and stronger communication between sensors and actuators. Multi-agent systems or distributed control, depending the research community, is a solution to reduce the complexity into the network of the devices and to obtain collective intelligence. When each unit is built on its own agent and all them are actuating in collaboration, the coordinator unit reduce their tasks, even it would become not necessary no more, or it could be strictly dedicated to failure detection and robustness purposes.

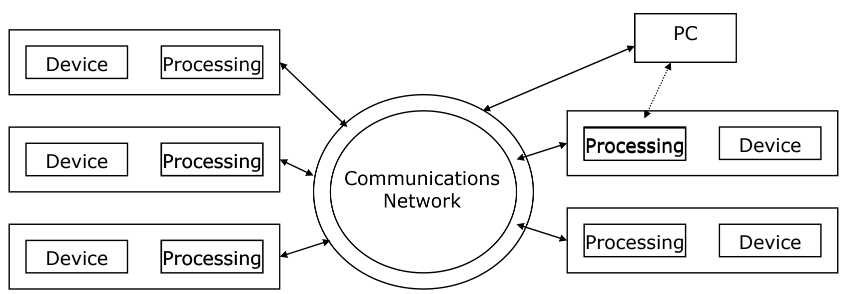


**Fig. 1.** Collaborative control architecture. Each intelligent hardware unit (IHU) is composed of an hardware element, processing resources and communication capabilities.

### 2.1 Multi-Agent System

A multi-agent system (MAS) is usually obtained when a complex system is divided into several agents specialized on a concrete task (planning, mobility, coordinator, ...) and they cooperate to solve a major task [8]. In MAS, all the participating agents and their coordination are usually codified a priori by the designer in order to accomplish the task to be completed. Our interest is to divide the general goal task in a different form.

According to [9], a complex system can be decomposed into a lot of little agents accomplishing a task with reduced complexity, but collaborating in such a form that the final system is smart. In this line, our vision is to design a multi-agent system based on a network of intelligent hardware units (IHU). An IHU is physically composed by a hardware device, or a reduced number of them sensing (actuating on) the 'external world', and an associated micro-controller (like a 8-pins PIC) generating decisions from available data and sharing information with other units (Fig.1), connecting it with the 'internal world'. The micro-controller is the part of an IHU processing information and communicating. An external computer will manages the general goal task to be realized and it will inserts reliability over all the system.

This is the more direct form to deploy agents into the environment in their simplest form. Information flowing through the IHU could be processed by a rule-based expert system when the final effect of its decision be critical for the user safety. Soft-computing techniques will be used when the device must to adapt to or learn from the ambient, or repetitiveness must be avoided to eliminate negative feedback from the user. In order to warranty adaptability, controllers should be not, or not completely, pre-programmed, as a main difference with standard MAS. For instance, if artificial neural networks (ANN) are used, their weights and bias should be initialized with random values. Decisions taken for the embedded agents and relevant data sensed from the world are available all around the network through a communication network. Hence, perception is considered in a modular form, i.e. multiple specialists are dedicated to extract relevant information for each active behavior.

Main objective is that a certain proportion of agents learn in a no supervised form what an individual task they must to implement on their device to reach the common goal. Because their simple structure, their task is primarily to adequately translate the received or sent signals by the hardware device connected to them, in such a form that shared translated information allows to reach the goal task demanded for the user. Learning and adaptation derived from the information processing allows the smart home to improve its performance in a number of forms: (a) knowledge is inserted into the system (facts, behavior, rules); (b) concepts are generalized from multiple instances; (c) information is more efficiently re-organized into the system; (d) new concepts are discovered or designed; (e) experience is used.

Appropriated devices selection, wireless communication, user interfaces, and internet link are necessary to design the whole smart home, however it is out of the scope of the present document, and it would be considered that they are available in the home.

### 2.2 Cooperation of IHUs

The intelligent hardware units conforming the whole net must be doted of such a sufficient information and decision coordination that a basic behavior can emerge after a training phase of the processing elements embedded into the micro-controllers. Processing in each EHI is double. First, by using a soft-computing algorithm, to translate and to interpret information shared through the communication network and to determine what to do about. A learning in two phases, off-line and on-line must be developed. Second task is, from sensed data, processing the information and augment it to translate for the rest of agents.

Communication between IHUs becomes a very important working element in this implementation [10]. It allows IHUs to send information to each other in order to correctly cooperate. During the training phase, the information sent between physical agents is used to learn coordination. When training is finished, IHUs have learnt how to treat information about the state of other elements to collaborate with them, and they will use communication to maintain that coordination. Furthermore, communication gives reaction capabilities in front of unexpected situations [11] to the automation home, including plugging and shut down of some integrated devices. Similar architectures have successfully been applied in real time control systems, like for example the control of two building elevators, or warehouse management, where a complex communication system including contracts and requests, was required for a proper collaboration of the different agents [12].

Common goal to be achieved for the system is sent to the cooperative network in the form of instructions of middle-level language. Fuzzy logic, kernel machines and qualitative reasoning are soft-computing techniques dealing with this kind of information. Each device (sensor, actuator, multi-sensor, sensor-actuator, multi-actuator) into an IHU is governed by a micro-controller (Fig.2). Its computational capacity varies according to the process to be implemented, memory resources and execution time. A real time communication network recruits shared information sent by the IHU's, and themselves determine what to do with this data, according to both, the knowledge about the global task and the information relative to the associated device.

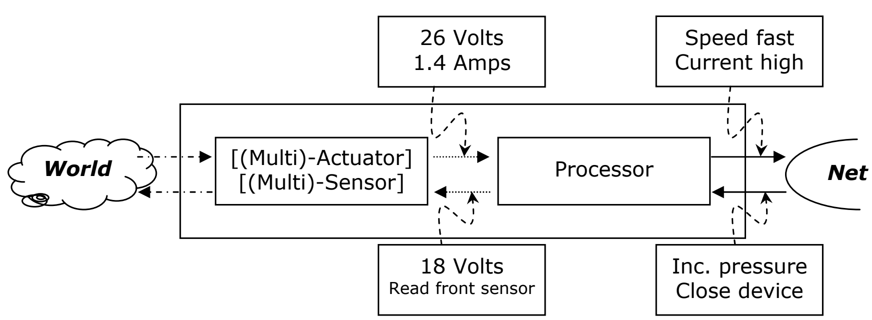


**Fig. 2.** Working methodology for a certain intelligent hardware unit (IHU).

### 2.3 Learning Algorithm: Distributed ESP

Learning the required system controller can be done by several methods. One demonstrated successful is neuro-control, it is, the use of neural networks to learn to implement the control policy. Neuro-control learning can be implemented using supervised training or neuro-evolution. While supervised learning uses a series of examples to train the networks by methods such as backpropagation, neuro-evolution uses genetics algorithms to evolve the required neural nets. The advantage of neuro-evolution over supervised learning is that it doesn't needs training examples to acquire the control knowledge. Neuro-evolution will also allow the robot to learn any required task to complete the job, without been specified before hand.

Due to the necessity of controlling multiple IHUs and to allow cooperation between them, a co-evolution method to obtain a proper control policy will be required. Co-evolution is a neuro-evolution method for the evolution of different nets with different roles in a common task. There are two types of co-evolution methods: competitive co-evolution, where the role of each agent is against the role of the other agents; and cooperative co-evolution, where agents share rewards and penalties of successes and failures. The last type will be the one used for the proposed architecture since each IHU processing subpart is evolved on its own population and interacts and cooperates with the others to solve the problem, contributing its best to the final solution.

The co-evolution algorithm used here is a version of the Enforced Sub Populations algorithm (ESP [13], [14]) with some modifications to obtain a distributed controller. It is a neuro-evolution method to evolve sub-populations of neurons forming a global neural network. In this process, groups of single neurons are created (called sub-populations), and a neuron is selected in a certain generation from each sub-population to form a hidden layer unit of a global neural network[3]. Every neuron inside one of the groups of sub-populations encodes with real values the state of the connections of such neuron within the global ANN. The information coded on every neuron is called the genome and it is the information that is evolved by the genetic algorithms.

To score the performance of the evolved neural network on the domain, a fitness function is created. The fitness function has to be calculated (and often, fine tuned) experimentally. It mathematically defines the global behavior required for the system. The fitness is an evaluation score of the neuron's performance inside the global ANN. During co-evolution, the nets of all IHUs must have a reward provided by the same fitness function. In [15], it is showed that rewarding the whole team with the same fitness produces cooperation between agents, while rewarding each agent with its own fitness value induces more competitive behavior, because every agent tries to maximize its own reward without paying attention to the group's goal.

The ESP algorithm implemented here follows the description in [16], including delta-coding to prevent premature convergence [14].

[3] In fact, the global neural network is each local neural network associated to an IHU.

## 3 Experimentation

From the human-factors perspective, a key constraint to creating ubiquitous technologies for monitoring health status into a smart home is that the technology must be unobtrusive. If people must take active participation in collecting the data about their health status, the data collected will be unreliable. Furthermore, if people remains aware of the presence of the technology, they are apt to change their behavior.

Robotic pets are being nowadays exploited as a successful option to insert technology into the home environment. In the perspective of our study a mobile robotic platform is very interesting because: (i) it is a current element being inserted into the homes; (ii) it allows to use sensors and actuators in a limited number; (iii) the common goal to be reached can easily be tested; (iv) autonomous robots are usual testing platforms for MAS and control architectures. So, experiments performed in this platform can be directly exported to the home environment, and it can be re-used for the autonomous robot research area.

The goal is to obtain an autonomous robot, a robotic pet, able to find a person on its space and then start orbiting around her in an endless loop: the pet will be randomly placed in some free point of the home where a person stands; first, it will look for the person using a random search algorithm or some RFID element; once it finds her, it will start orbiting around her at a *close* distance. In this form, it is possible, for example, to obtain health monitoring of a person arriving to his home, meanwhile the pet orbits around him, or through a certain variable scheduling.

This behavior will have to emerge by making cooperate four IHUs inside the pet. The simulated robotic pet used for the experiments consists of a squared platform where two infrared sensors and two motors are attached[4]. Three wheels control the movement of the robot: two wheels at the bottom, controlled by one motor each one, and a free wheel at the front, which gives stability but no control. Infrared sensors are placed at the upper-left corner of the robot, one pointing to the front and another pointing to the left, being able to detect objects from a range between 3 and 20 cm. The detection values of the sensors have been quantized in order to keep the whole system simpler. It will be considered that an object is *far* when distance is greater than 20 cm; *half* distance when it is between 20 and 10 cm; *close* when 10 and 6 cm; and *very close* when it is between 6 and 3 cm. Motors are placed at both lower corners of the robot. Their range of velocities has been also quantized, only allowing 4 different values: *full forward*, *half forward*, *stop*, and *half backward*.

Four different IHUs are required to represent the whole system (Fig3), one for each IR sensor and one for each motor, implementing a feed-forward artificial neural network with sigmoid activation function and one hidden layer with 12 neurons. All the nets have only one output neuron. Communication between

[4] 2 Sharp GPD202 IR sensors and 2 standard DC motors were simulated.

IHUs is performed by connecting the outputs of the neural networks to the inputs of the other nets, including itself.

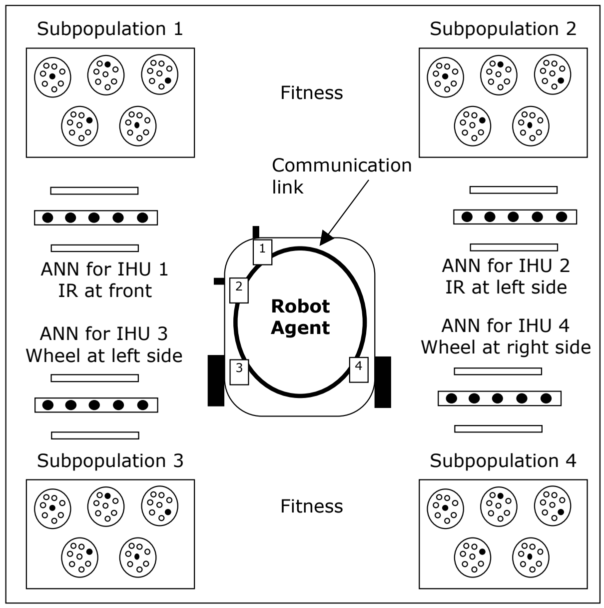


**Fig. 3.** Architecture for a robotic pet controlled by four IHUs.

Three types of experiments were developed to cope the features of the proposed architecture. First an experiment using ESP with a central controller, a second one using the distributed controller version of the ESP, and finally a third experiment, specifically devoted to analyze the intelligent hardware sensors.

### 3.1 ESP Learning for a Central Controller

For the central-controller approach, only one neural network is required, being the inputs the quantized data from motors and sensors, and outputs the responses for the motors; no output were required for sensors since they are passive elements. In order to obtain the required behavior the following fitness function was defined,

$$F = \begin{cases} +1, \ if \ S_x = close \wedge S_y = far \wedge SpeedL > 0 \wedge SpeedR > 0 \\ -1, \ else \end{cases} \tag{1}$$

rewarding neurons only when the robotic pet is running with both wheels, detecting an object on his left at *close* distance, and detecting nothing in front of him. When the previous condition is not met the group of neurons is punished.

By punishing in this way the pet tries to find the person as soon as possible. The required behavior was obtained after an average of 140 generations, being the maximum fitness obtained of 239 out of 300 steps. Another interesting result is that even when evolution was left running for several hours until generation 1500 is reached, fitness never improved, having a maximum of 240.

### 3.2 ESP Learning for a Distributed Controller

The ESP learning algorithm designed by the UTCS Neural Nets Research Group [17] was modified to train one neural network for every IHU, in a distributed form. All the networks were evolved at the same time. The multi physical-agent system was obtained by using parameters similar to those employed for the centralized controller, and the same fitness function in order to obtain the same evolved behavior.

Significant differences with respect to the central control experiment can be observed: first, the average number of required generations to obtain the same behavior was drastically reduced (78). Secondly, when evolution was left running, the obtained maximal fitness reached values of about 260. Both results empirically show that a distributed controller works and it can learn faster and better than a central one, as was stated in [7] for the predator-prey game.

### 3.3 Intelligent Hardware Sensors

When using four IHUs to control the pet, the question about the necessity of using neural networks in the sensors arises. Since its only job is to receive the value from the sensor and share it with the rest of agents, is it really necessary to use those processing units in such passive elements? Is the neural network attached to the sensors doing any active job? To answer these questions, a third experiment in two phases was performed .

First, a distributed controller with only two IHUs was designed, one IHU controlling each motor and directly connecting the output of the sensors to the ANNs of those actuators, hence no neural nets were attached to the sensors. Results show that the final robotic agent is able to acquire the required behavior, but his fitness dropped down a little bit (an average of 232 out of 300, against 254 out of 300 for the four agents distributed controller). From the result of this first experiment, it can be concluded that sensor agents are really doing some helpful job, but maybe the improvement is not enough significant if the higher processing cost is considered.

The second phase of the experiment was concerned with the achieved learning of the sensor agents. Learn intelligent hardware sensors some abilities from the relation internal-external world? or are they sharing only their sensed value, without any interesting processing on it? To ask this question, function learned by the ANN associated to a IR sensor X was plotted under two different circumstances. Fig.4, on the left, shows the function learned by sensor X according to the distance sensed when Motor L and Motor R value *half-speed* and sensor Y senses nothing.

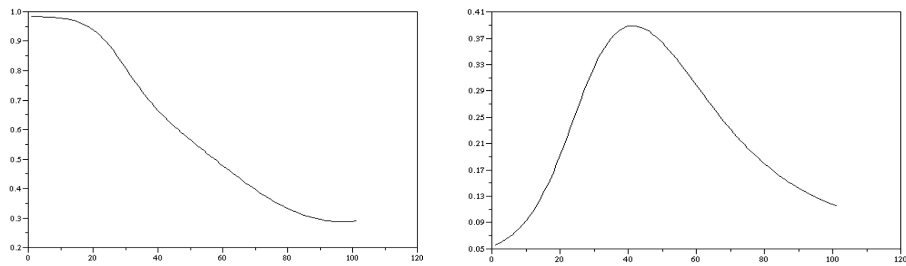


**Fig. 4.** ANN output learned by the IR sensor X. Because function depends on 4 variables, sensor Y, Left motor and Right motor have been fixed, showing only the dependence of the network output in front of sensed X values.

Then, the code emulating the sensor X was modified to insert a failure. Failure consists on allowing to sense objects only at *close* and *very-close* distances. After this modification, the 4 IHUs were evolved again, and Fig.4 (right) plots the resulting processing function of the sensor X ANN when Motor L and Motor R value *half-speed* and sensor Y senses nothing.

Comparing both figures, it can be seen that the IHU is learning how to process the input signal, and this processing highly depends on the sensor behavior. First of all, plots in Fig.4 show that the sensor agents are actives and they are not only taking the value sensed and sharing it to the rest of IHUs. Second, their job depends on the situation. So, it is possible to use ANNs on the sensors able to adapt by themselves to a noisy environment and learn what kind of treatment would be necessary to take the most of the environment to perform the required task.

## 4 Conclusions

Automation systems in smart homes are concerned with sensors and actuators that monitor the occupants, communicate with each other, and intelligently support the occupants in their daily activities. Collective intelligence technology will be essential to analyze data from these distributed sensors. Research through this article focus on achieving the adaptation of soft-computing algorithms, developed usually as software modules in conventional computers, being implemented into specific hardware to obtain adaptation to the users. The introduced approach to the control problem of the complex system use a division into agents in a very lower complexity level that it is usual in MAS, allowing a higher implication between the physical and computing components if it is compared with standard literature.

A main feature of the proposed architecture is that it is not even necessary to codify a priori the task to be developed for each agent to reach its objective; moreover, it is not necessary at all to specify the particular goal for each agent, only the desired global goal task is indicated by the designer. This very interesting feature is obtained because the reduced computing complexity associated to each IHU. Implicitly, system suppose that it there exists some reactive connection between sensors and actuators, so that information generated for these physical elements is translated according to the necessities of the ensemble of agents and the degree of goal task accomplishment for each IHU. No IHU is in charge of any specific activity, that means, a certain sub-goal of the global task. Implication of any IHU in the general process is self-recognized for them during the not supervised learning phase.

Experimentation on a simulated robotic pet platform shows that the novel architecture improves its centralized counterpart, it learns faster and better. Increase of complexity associated to the collaborative organization is rewarded with a more efficient control of the system. Meanwhile robustness has been demonstrated thanks to the processing capability of the intelligent hardware sensors, scalability of this architecture is a open research problem being dealt following MAS approaches.

## 5 Acknowledgments

This work is partially supported by the Spanish ACCUA project "Arquitectura Cooperativa de Computacin Ubicua basada en elementos hardware inteligentes en el entorno de una vivienda Asistida" (TIC2003-09179-C02-01).

## References

1. Crabtree, A., Rodden, T., Hemmings, T., Benford, S.: Finding a place for ubicomp in the home. In Dey, A.K., Schmidt, A., McCarthy, J.F., eds.: UbiComp 2000. Volume 2864 of Lecture Notes in Computer Science., Seattle, WA, USA, Springer (2003) 208–226
2. Anderson, R.N.: Method for constructing complete annual u.s. life tables. Vital and Health Stat **129** (2000) 1–28
3. Erickson, P., Wilson, R., Shannon, I.I.: Years of Healthy Life. Number 7 in Statistical Notes. US Department of Health and Human Services (1995)
4. Haigh, K.Z., Kiff, L.M., Myers, J., Guralnik, V., Krichbaum, K., Phelps, J., Plocher, T., Toms, D.: The independent lifestyle assistant (i.l.s.a.): Lessons learned. Technical Report ACS-P03-023, Honeywell Laboratories, 3660 Technology Drive, Minneapolis, MN 55418 (2003)
5. Warren, S., Craft, R.L., Bosma, J.T.: Designing smart health care technology into the home of the future. In: Workshops on Future Medical Devices: Home Care Technologies for the 21st Century, Rockville, MD (1999)
6. Krco, S.: Bluetooth based wireless sensor networks implementation issues and solutions. In: Proceedings of Telfor 2002, Belgrade, Serbia and Montenegro (2002)
7. Haigh, K.Z., Yanco, H.A.: Automation as caregiver: A survey of issues and technologies. In: AAAI Workshop on 'Automation as caregiver', July, American Association for Artificial Intelligence (2002) 39–53

8. Vlassis, N.: A concise introduction to multiagent systems and distributed AI, Informatics Institute, University of Amsterdam (2003) http://www.science.uva.nl/~vlassis/cimasdai.
9. Minsky, M.: The Society of Mind. Simon & Schuster, Inc., New York, NY, USA (1986)
10. Werner, G., Dyer, M.: Evolution of communication in artificial organisms. In Langton, C., Taylor, C., Farmer, D., Rasmussen, S., eds.: Artificial Life II, Redwood City, CA, Addison-Wesley Pub. (1992) 659–687
11. Yong, C.H., Miikkulainen, R.: Cooperative coevolution of multi-agent systems. Technical Report AI01-287, Department of Computer Sciences, University of Texas at Austin (2001)
12. Schrott, G.: A multi-agent distributed real-time system for a microprocessor fieldbus network. In: Proceedings of the 7th Euromicro Workshop on Real-Time Systems, Odense, Denmark, IEEE Computer Society Press (1995) 302–307
13. Gómez, F., Miikkulainen, R.: Incremental evolution of complex general behavior. Technical Report AI96-248, Department of Computer Sciences, University of Texas at Austin (1996)
14. Gómez, F., Miikkulainen, R.: Solving non-markovian control tasks with neuroevolution. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, Denver CO, Morgan Kaufmann (1999) 1356–1361
15. Balch, T.: Learning roles: Behavioral diversity in robot teams. Technical Report GIT-CC-97-12, Georgia Institute of Technology (1997)
16. Vikram Aedula, C.D.: Collective behavior in robots using evolutionary neural networks. Abstract, Smart Engineering Systems Laboratory, University of Missouri-Rolla, Rolla, MO 65409-1060 (2002)
17. Group, U.N.N.R.: (Neuro-evolution software)