A Soft-Computing based Approach to Intelligent Association in Agent-Based Ambient-Intelligence Environments

Hakan Duman, Hani Hagras, Vic Callaghan
Intelligent Inhabited Environments Group
University of Essex, Colchester, CO4 3SQ, U.K.
Email: hduman@essex.ac.uk

Abstract*:* *The synergistic and complementary use of fuzzy logic and neuro-computing has initiated the development of soft computing methodologies. In this paper, we investigate the research issues related to Intelligent Association in multi-embedded agents environments in respect to soft computing techniques. We introduce a new system that learns the importance of associations represented by fuzzy weights that are dynamically calculated using different Hebbian Rule methodologies. By doing this, redundant associations can be removed and new ones can be discovered. The technique is particularly suitable for the field of pervasive computing where there is a need for agent-based artifacts to operate within time-variant agent communities.*

Keywords*:* *Embedded Agents, Intelligent Associations, Fuzzy Control, Hebbian Learning, Pervasive Computing, Ubiquitous computing, Ambient Intelligence*

1. Introduction

Anthony awoke to the sound of his alarm clock, but the sound did not come out from a regular alarm clock beside the bed. The clock just displayed the time and the alarm sounded from the speakers of a hi-fi on the opposite wall. He had set the alarm time using his PDA; alternatively he could have used his TV or desktop computer. The alarm stopped and the blind opened as Anthony got out of the bed. The smart electric blind knew that he liked daylight in the morning and after checking the outside light level, its motor had powered the blind to open. It is checking the light condition because if it was dark outside it would have switched on the room lights. The aroma of fresh coffee spread out almost instantly. The coffee maker knew that Anthony liked drinking coffee while he was checking his emails. The coffee maker was also a smart one that used the time setting from the clock, data from several activity monitors and an internal model of his past behaviour to optimise the brewing start time. Needless to say the smart central heating and hot water system had intelligently learnt to turn on so as to minimise energy wastage whilst maximising Anthony's comfort.

Are these kinds of devices and environments already becoming a part of our lives, or is it just science fiction? How hard is it today to imagine that billions of networked devices interacting with each other to support our everyday activities?

Only ten years ago, Mark Weiser introduced the term *ubiquitous computing* which described a vision for living environments populated with "computerized" objects where the emphasis was on greater user-friendliness, more efficient service support, user-empowerment, and support for human interactions [7]. It seems as though this vision is already becoming reality. Networked wearable devices (eg mobile phones or PDAs) have already found their way into everyone's briefcase whilst internet appliances (eg smart fridges), are starting to appear on the market [6]. However, ubiquitous computing takes the vision of the future environments even further. It envisages integrating large number of smart/intelligent appliances and putting electronics into hitherto "silicon free" items (e.g. garments, chairs, cups etc). In general, this new vision might be described as enhancing, everyday objects with sensing, effecting, computing and communication capability, resulting in computationally complex and highly dynamic environments. An obvious barrier to achieve this is the inherent complexity of the technology and the large number of devices. It cannot be assumed that people who purchase these devices understand or are able to deal with the underlying technology. An important aspect of enabling collections of artifacts to collaborate together is a mechanism to determine which artifacts are associated together. Industrial and building automation institutions have been working for decades to develop network protocol standards but to date, at best, the tools only support some limited automation of association (e.g. a LonWorks controlled light might flash to request a switch action, resulting in an association of the two devices). However, with current systems, it is not possible to perform fully automatic (without the help of a user), or even intelligent, associations between devices. It is expected that artifacts in intelligent environments will self-organize into functional clusters to perform a higher-level coordinated action (albeit with some user imposed constraints to reflect security, safety or personal needs). Currently, in existing device networks, when a new artifact enters or leaves a community (or sub net), it becomes the user's responsibility to reconfigure the network accordingly, otherwise the whole system would become unstable and fails to operate in the way it was designed to work. Another constraint is that computer based artifacts have limitations in respect to memory size, processing capacity and most importantly communication bandwidth. This means, that the devices can only be connected to a limited number artifacts at the same time. If an artifact requires sensory information from other artifacts to perform its control function, it has to communicate with them to request this information. However, in a domain with hundreds (and possibly thousands) of artifacts this would increase the communication overhead and lead to a system delay or even crash. If the artifact would know only the inputs from artifacts that it is interested in, it would only communicate with these instead. Also, during the lifetime of the system a service can become less important and another artifacts' service more important for a device. If it is of less importance (e.g. device removal from the system or breakdown) the association should be removed and new services can be discovered and associated. Clearly producing a robust and reliable design for such a complex artifact association system, involving hundreds of thousands of heterogeneous fixed and mobile artifacts, is a most challenging problem. A possible solution to this is to employ some form of embedded intelligence to work on behalf of the user to dynamically configure this inter-artifact communication infrastructure. Solving this problem can be seen as addressing two issues; the system's ability to accurately determine a user's task and intention, and its ability to autonomously develop associations between artifacts to assist the user in undertaking is task and meeting his intentions.

2. The Intelligent Association System (IAS)

2.1 The IAS Architecture

We define three types of embedded-computing artifact: (1) *Passive* – open-loop systems that collect, process and provide sensory information (2) *Automatic* – closed-loop systems that execute a set of predefined control rules stored in the computational logic, and (3) *Embedded-A*gents – closed-loop system that incorporate mechanisms for adaptive control and are capable of learning new behaviours/rules rather than simply executing pre-programmed rules. A further characteristic of an embedded-agent is that it is inseparably integrated into a computer-based artifact (as against a conventional intelligent agent, which is more often a soft process running on a conventional computer) [1]. Examples for passive artifacts include light level, and pressure sensors; intelligent agents have a functionality and can reason and infer and can be embedded into heaters, mobile robots etc. Attaching an agent to an artifact means that each agent offers a particular low-level service. In order to perform a higher-level coordinated action, many agents can be clustered together into *Agent Societies*. For example, agents attached to lighting elements of a light panel may coordinate to dynamically vary their overall intensity based on environmental factors. An intelligent agent within a society can become the local leader of its society, a so-called *Embassador Agent*. The main role of the Embassador is to communicate with other societies on behalf of its members (in a bandwidth efficient way) and combine overlapping agent societies. This is not a centralized architecture as the embassador role is dynamic and can be readily transferred to other agents (all agents being seen as "embassadors in-waiting". Detailed description on the characteristics and functionality of an Embassador Agent can be found in [2]. It is now common for such embedded-agents to have a network connection thereby facilitating multi embedded-agent systems. In a fully distributed multi embedded-agent systems these agents cooperate by means of either structured or ad-hoc associations with its neighbours. An association is a soft or hard link between two or more agents. The associations can be in one of the three ways specified as follows: (1) *Manual Association*, where agents are associated manually (eg the user making the equivalent of physical connections), (2) *Edited Associations*, where the associations are managed using an association editor (e.g. selection via a PDA based tool) and (3) *Intelligent Associations*, where the associations are managed by intelligence in each of the agents (or the Embassador agent, at a given instant, in the case of inter-society association). The latter is the main target of this paper. The associations describe the strength (hereafter: *weights*) of the connections between the agents. The associations are represented as either discrete (0,1) or continues across a range $[0, w_{max}]$. The discrete weights illustrate only the binary activation of the associations (active or inactive). In contrast, associations that are continuously adjusted through learning the impact of agents to others can determine the real strength of the associations and thus it is easier to judge which associations are more important for the agent and which ones are regarded as “redundant”. This information becomes especially significant for systems containing agents with bandwidth and memory limitations. Figure 1 illustrates the IAS architecture comprising of the Embassador (here: A_2) and agent societies.

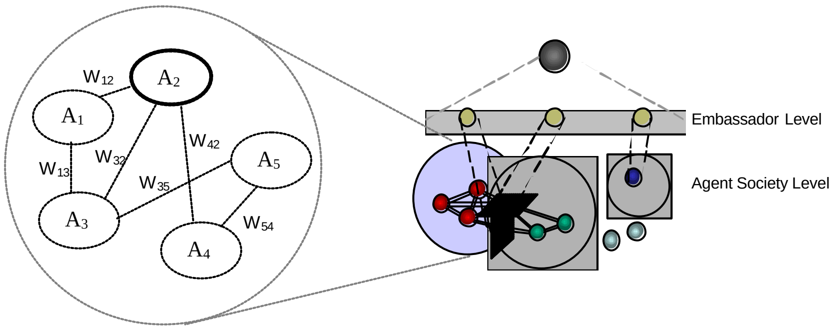

Figure 1: The IAS Architecture

As proof-of-concept we have implemented a tool, the IAS Editor, which can visually display available agents, the adjustments of their association weights and enable the user to create/dissolve agent societies and associations. It should be noted that in this paper the main focus is set to dynamically calculate the association weights between interconnected agents in order to identify redundant connections.

2.2 The IAS Editor

We used the iDorm at the University of Essex [5] as our test bed. The iDorm is a multi-use space (i.e. contains areas with different activities such as sleeping, working, entertainment etc) and can be compared in function to a room for elderly or disabled people or an intelligent student or hotel room. The room consists of normal furniture that will allow the user to live comfortably as it has a bed, a working desk, bedside cabinet, and wardrobe, a multi media PC that the user can use for working or entertainment. Besides this, the room contains a large number of passive/active and intelligent embedded-computing artifacts, such as indoor and outdoor light level, temperature and humidity sensors, chair and bed pressure sensors, bed lamp, desk lamp, controllable vertical blind etc. The iDorm is populated with different kinds of computing artifacts, which can be static, or mobile (e.g. mobile robots) or “wearable” (e.g. PDAs).

The IAS Editor is a tool to facilitate users to manually cluster agents to agent societies and create/edit associations between them. The main role of the IAS Editor is to provide the following services to the user: (1) Discovery of agents and their services, (2) Visualize information of discovered agents, incl. their services and constraints, (3) Creation of Agent Societies and assigning a high-level function for them, (4) Support the user in creating and removing associations and rules in natural language (fuzzy values), and (5) Offer the user to monitor performance and abnormalities.

During our experimentations, the IAS Editor was used to configure the agents. In the first experiment described in this paper, we have associated every agent with other existing agents. In addition, initial (expected) fuzzy behaviour rules were assigned for the agents, e.g. *IF light switch is ON the ceiling lamp is HIGH*. The IAS Editor sends these rules to the corresponding agents to add them into their behaviour rule bases. Please refer to [2] for a more detailed description on the functionality and operation of the IAS Editor. As soon as the rules have been sent to the agents, the agents start to operate and the IAS Editor simultaneously displays the learnt/calculated association weights to visualize the importance of the created associations for the user.

3. Association Weight Calculations

People living in their homes rarely change their habits and actions. Although there are daily fluctuations, our work has shown that there is a distinct and discernable habitual behaviour pattern, which can be obtained by monitoring a person's everyday activities. However people are essentially non-deterministic and highly individual, therefore there is a need for a system that particularizes for individual users rather than generalizing for a group of users [3]. Thus, the system needs to provide an online, life-long, non-intrusive method for learning behaviours and anticipatory adaptive control for these highly dynamic physical environments.

The mechanism to learn user behaviours used during our experimentations within the iDorm is called *ISL* (Incremental Synchronous Learning). The online learning, adaptation and control algorithm is based on a hierarchical fuzzy genetic-like system that learns rule bases (behaviours) of intelligent agents in without explicit human interaction [3]. The learning cycle is initiated when a behaviour, or collection of behaviours, is over-ridden a number of times e.g. 3 (the number being the learning inertia and is chosen to stop the system reacting to erratic behaviour) meaning the existing rule no longer satisfies them. Any new rules are stored in a fuzzy rule base according to the user's new actions. The system requires a relevant representation of the environment state so that it can learn the necessary rules. The rule base consists of every state variable that is known by the system (eg sensors data, state of other agents). It is obvious that artifact associations will have differing value to the agent rule base. For example, the desk lamp room light level is almost certainly more important than temperature. In addition, an association that was important at the first place can become of less important during the lifetime of the system so that the agent can remove it. This is especially essential for agents with computational limitations, which, for example, need to keep connections within a given number (e.g. only 5 associations). For this, the agents need to continuously manage, and find the best possible associations, by creating new or dissolving redundant associations dynamically. How can redundancy of some association be identified? One approach would be to find the causal relationship and calculate the weights of the associations between the agents or, in other words, which agent caused another agent to change its state, and to what degree?

There are many ways to "learn" the weights between the agents. One of the mostly used associative learning mechanisms is the Hebb learning rule. In this an association weight is increased with a simultaneous occurrence of pre-associative and post- associative agent activities [4]. The learning and calculating the association weights is initiated with every new rule that ISL learns. The algorithm is as follows:

Step 0: Initialize all weights: $w_{Ai} = 0$ *(i=1 to n)*
Step 1: Initialize the learning rate ? ?= 0.1
Step 2: For each pre-associative and post-associative agent pair (i,j=1 to n), a : p, do following:

$$w_{ij}(q) ?? ?? w_{ij}(q-1) ?? ?? a_i(q)\, p_{ij}(q) \qquad (1)$$

Step 3: On a regular basis apply threshold, t=0.1, to determine redundant associations; if redundant associations > 0 then do Step 4, otherwise go to Step 2.

Step 4: Adjust every agent's rule base according to the importance of the associations

In the following experiments, we have applied, online, the above algorithm to every newly created fuzzy rule generated by the ISL. For this experiment, an occupant resided in the iDorm for 24 hours.

We have used 13 embedded-computer artifacts for this experimentation, with one ISL agent-based artifact acting as an Embassador and which learnt the associations described in these experiments. It should be noted that the ISL agent created over 70 user related rules during this time. However, here we have applied Hebbian learning to only 20 rules and assume that only these rules have been learnt of the duration of the experiment. This is done for graphical representation purposes, as described next.

A_1 - Light Level	A_2 - Occupancy	A_3 - Light Switch	A_4 - Bed Pressure	A_5 - Chair Pressure
A_6 - Temperature	A_7 – Blind	A_8 - Ceiling Lamp 1	A_9 - Ceiling Lamp 2	A_{10} - Ceiling Lamp 3
A_{11} - Ceiling Lamp 4	A_{12} – Blower	A_{13} - Heater		

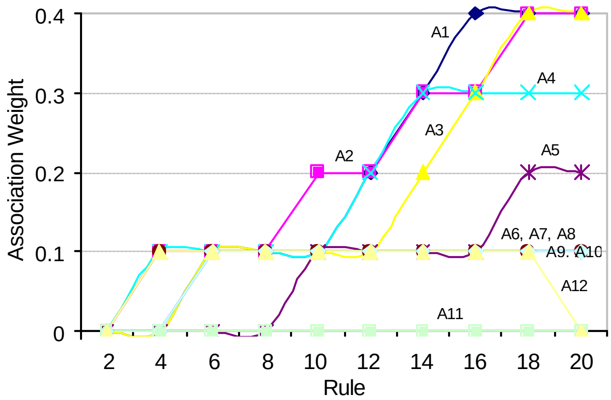

Figure 2: Simple Hebbian Learning

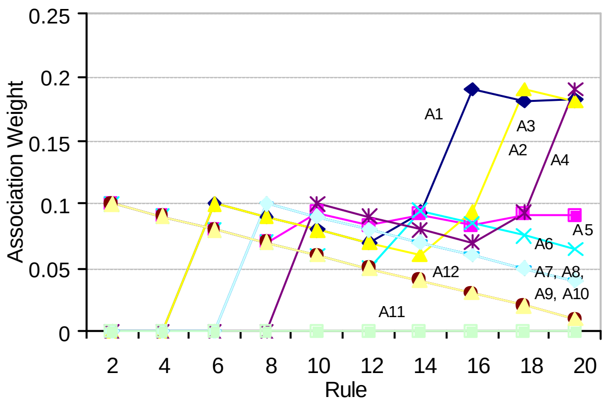

Figure 3: Hebbian Learning with Decay

Figure 2 illustrates the learnt weights of the blinds' association with the remaining 12 computer-based artifacts. The horizontal and vertical axes represent rule and the association weights respectively. The weights were calculated using the simple Hebbian Learning Equation (Eq. 1). After applying the threshold value, as stated in *Step 3* of the proposed algorithm, it shows the blinds associations with A1, A2, A3, A4 and A5 are more "important" than the rest of the agent. It is however obvious that repeated application of the Hebbian learning rule leads to increase in post-associative agent $a_i(q)$ and therefore exponential growth that finally drives the association to saturation, regardless of the learning rate ? . After this point, learning becomes almost impossible and the weights are not reflecting the "real" strength of the associations anymore [4]. In other words, the agent associations that become less important over time, would not be recognized and decrease in strength. If we consider association weights that have more than 10% of the maximum weight as important, then the weight has to be in a rage of [0, w_{max}]. In order to avoid this saturation, we modified the Hebbian Learning Rule and applied a decay value (Figure 3) to our algorithm. The equation for this learning method is as follows:

$$w_{ij}(q) ?? ?w_{ij}(q-1)(1-?) ?? ?\, a_i(q)\, p_{ij}(q) \qquad (2)$$

As it is clear from Figure 3, the Hebbian Learning with Decay differs only from the result of the Simple Hebbian Learning rule in respect to the association between A5 and the blind when the threshold is set to 0.1. Here, A5 doesn't pass the importance barrier and the agent would regard this specific association as unimportant. After removing redundant associations through deleting the corresponding inputs and outputs from the rule base, the performance of the agent remained the same and satisfactory for the control of users actions with less specified number of association. Obviously, for this case, the Hebbian Learning with Decay performs as good as the simple Hebbian learning but with fewer associations. The definition of the threshold is mainly depending on the computational limitations of the artifacts. As mentioned before, the number of associations any device can handle at one time will be dependent on its computational capabilities; therefore the threshold has to be determined dynamically to match the device capabilities (Figure 3 would be appropriate for an device able to handle 4 associations at a time). In addition, the maximum weight that an association can get is defined by w_{ij}^{max}=? ???were ? ?is the learning rate and ??is the decay value. By increasing the decay value ?, the association weights are literally "forced" to decrease more quickly than increasing the association weights. However, association weights that increase as a result of active artifacts are as important as association weights that decrease because of inactive artifacts. Thus, the variables are set to 0.1, so that the association weights remain between [0,1]. Another purpose keeping the association weights within this range is that we can represent them in fuzzy values. For this we have generated a membership function with the fuzzy sets *Very low*, *Low*, *Medium*, *High*, *Very high*. By doing this, the threshold can be set to be a fuzzy value rather then using discrete values, e.g. IF weight is *very low* THEN *delete this association*. Of course, the membership functions need to be adjusted according to the association constraints of the agents, as described for the threshold, but this work is ongoing and is not part of the research reported in this paper.

4. Conclusion and Future Work

The experimental results confirm that, for the environment in question (the iDorm, many of the associations that were initially set were not essential to operate a system. The benefit of learning/calculating association weights proves to be an important issue in solving problems that comes with systems, which simply associate "everything with everything". The problems arising from this include (1) Bandwidth and communication overhead, (2) storage limitations and dimensionality problems in rule bases, (3) expert/technical knowledge required, and (4) unstable and non-dynamic agent societies.

In this paper, we have introduced the IAS and its GUI, the IAS Editor. The tool supports user-driven inter-device and inter-society association. While the system learns the occupants' behaviour via an ISL fuzzy learning mechanism, simultaneously associations are derived by an independent Hebbian rule learning mechanism. After applying the threshold value to the learnt weights, redundant associations can be identified and removed via removing the corresponding inputs and outputs in the rule-base. We have also described Hebbian Learning with Decay, which provides a more reliable way of calculating the association weights in contrast to the traditional Hebbian Learning.

The calculation of association is a part of an ongoing research project that currently investigates the possibility of applying Fuzzy Cognitive Maps (FCM) to IAS. In a similar to the approach described herein, FCMs encode rules in a networked causally connected structure where rules fired based on both a given set of initial conditions and on the dynamics of the FCM. We are also, intending to explore variations, including causally and fuzzily interconnected agents system. The main target of this approach is to automatically construct FCM that will represent individual behaviours.

Acknowledgements: We are pleased to acknowledge the support of our colleagues in the University of Essex IIEG group (http://iieg.essex.ac.uk), and the EU Disappearing Computer programme for the funding that made this research possible.

References

[1] V. Callaghan, G. Clarke, M. Colley, H. Hagras (2001), Embedding Intelligence: Research Issues for Ubiquitous Computing, In *Proceedings of the International Conference on Ubiquitous Computing in Domestic Environments*, Nottingham, UK, September, 2001.

[2] H. Duman, H. Hagras, V. Callaghan, G. Clarke, M. Colley (2002), Intelligent Association in Agents Based Ubiquitous Computing Environments, In *Proceedings of the International Conference on Control, Automation, and Systems*, Muju, Korea, October 2002.

[3] H. Hagras, M. Colley, V. Callaghan, G. Clarke, H. Duman, A. Holmes (2002), A Fuzzy Incremental Synchronous Learning Techniques for Embedded-Agents Learning and Control in Intelligent Inhabited Environments, In *Proceedings of the IEEE International Conference on Fuzzy System*, May 2002.

[4] S. Haykin (1999), Neural Networks: a comprehensive foundation, *Second Edition, New Jersey, Prentice-Hall Inc.*, ISBN 0-13-27330-1, 1999.

[5] A. Holmes, H. Duman, A. Pounds-Cornish (2002), The iDorm: Gateway to Heterogeneous Networking Environments, In *Proceedings of the International ITEA Workshop on Virtual Home Environments*, Paderborn, Germany, February 2002.

[6] LG Electronics, *http://www.lge.com/c_product/h_network/products/product.shtml*.

[7] M. Weiser (1991), The Computer of the 21st Century, In *Scientific American*, Vol. 265, No. 3, pages 66-75, 1991.