In Special Issue of Ambient Intelligence of the ACM Transactions on
Autonomous and Adaptive Systems, Vol.5, No.2 pp.40-74 May 2010.

Multi-Society based Intelligent Association Discovery and Selection for Ambient Intelligence Environments

HAKAN DUMAN, HANI HAGRAS, and VICTOR CALLAGHAN
University of Essex

Our environments are being gradually occupied with an abundant number of digital objects with networking and computing capabilities. After these devices are plugged into a network, they initially advertise their presence and capabilities in the form of services so that they can be discovered and, if desired, exploited by a user or other networked device. With the increasing number of these devices attached to networks the complexity to configure and control them increases which may lead to major processing and communication overheads. Hence, the devices are no longer expected to just act as primitive stand-alone appliances which only provide the facilities and services to the user they are designed for, rather they can offer complex services from unique combinations of devices; which in turn creates the necessity for these devices to be equipped with some sort of intelligence and self-awareness which enables them to be self configuring and self programming. Intelligence in devices is obtained by embedding intelligent agents into them which provides them with proactive control and learning capabilities. Self-awareness within agents enables capabilities to operate in with a minimum of cognitive loading of the user, thereby supporting the vision for cognitive disappearance or ambient intelligence. This paper presents a novel intelligent embedded agent technique for reducing the number of associations and interconnections between various agents operating within an AIE in order to minimize the processing latency and overhead caused by message flooding in a Publish/Subscribe (pub/sub) middleware whilst reducing the cognitive load of configuring these associations to personalize themselves to the user needs. The main goal of the proposed fuzzy based intelligent embedded agents (F-IAS) includes learning and adapting the network configuration and the system functionality to meet the user's needs based on monitoring the user behaviors in a lifelong non intrusive mode to preemptively control the environment on his behalf. In addition, the F-IAS agent aimed at reducing the agent associations and interconnections to the most relevant set in order to reduce its processing overheads and thus implicitly improving the system overall efficiency. Moreover, we employ embedded ambassador agents, namely embassadors, which limit the number of messages reaching the societies by performing an analysis and filtering routine to determine if the propagated events match the desired criteria of the member agents of the societies. Embassadors are also utilized with novel characteristics to discover and select associations among agent pairs residing in separate societies based on a concurrence analysis of published events. In order to validate the efficiency of the proposed methods we will present two set of unique experiments. The first experiments described the obtained results carried out within the intelligent Dormitory (iDorm) which is a real world test bed for AIE research. Here we specifically demonstrate the utilization of the F-IAS agents and discuss that by optimizing the set of associations, the agents increases efficiency and performance. The second set of experiments is based on emulation of an iDorm-like large scale multi society based AIE environment. The results illustrate how embassadors discover strongly correlated agent pairs and cause them to form associations so that relevant agents of separate societies can start interacting with each other.

Categories and Subject Descriptors:
General Terms:
Additional Key Words and Phrases: Ambient Intelligence, Fuzzy Control, Multi-Agent Systems

1. INTRODUCTION

A vision to deliver greater user friendliness, support for human interactions and more efficient services that aim to continuously improve the lifestyle in our living spaces was initially coined by the Advisory Group to the European Community's Information Society Technology Program (ISTAG) [Wilson 2004]. This vision is called *Ambient Intelligence* (AmI). In the last few decades, the advent of wireless technologies and the achievements in miniaturization of electronic devices have proven that the realization of AmI will not remain fiction but will soon to become real.

An Ambient Intelligent Environment (AIE) consists of a multitude of interconnected embedded systems which are tangible everyday objects and artifacts embedded with computational and networking capabilities which form a ubiquitous, unobtrusive and seamless infrastructure that surrounds the user. These objects are often augmented with intelligent processes to assemble intelligent *embedded agents*, which provide intelligent reasoning and decision making. These intelligent agents are then seamlessly integrated into AIEs to form an intelligent "presence" allowing the AIEs to *identify* the users and be *sensitive* and *attentive* to their particular needs by autonomously learning from their behavior and thus configuring and preemptively controlling the user's environment on their behalf. They also need to provide adaptive learning mechanisms that allow the system to deal with uncertainties and adapt to the changing environment and user preferences over short and long term intervals [Duman et al., 2007a]. Moreover the applied mechanisms should result in transparent representations in the form of human readable rules so that they are interpretable and accessible by the end users.

In an AIE, the multitude of interconnected embedded agents would enrich the user's environment and provide more effective support, however with the increasing number they may also result in major processing latencies accumulating from the computational and communication overheads as well as creating inherent complexities in programming and configuring the AIEs. Thus, the agents are no longer expected to just monitor and learn the habits of the occupant but also provide self-awareness aiming at enabling them to manage themselves in the most dynamic, efficient, economic and reliable way. Self-aware intelligent agents are *dynamic* and capable of configuring themselves to changes in the network structure which makes them *resilient* and *fault-tolerant*. They are *efficient* as they aim to reduce their processing latencies by utilizing mechanisms to find the *most relevant associations* among agents necessary to meet the environments and the user's needs.

Consequently they remove associations to less important agents thereby decreasing the processing overhead, as only relevant messages are communicated and processed.

The current state of AIE research in the literature mostly demonstrates the use of single intelligent agents, mostly in the form of a PC running a software agent, aiming to learn and predict the preferences of the user as well as to operate the environment at the most efficient and effective level. Only a few describe the use of multi-agent system architectures for AIEs with less focus on integrating some sort of intelligent mechanisms to manage the interconnections, essential for AIEs.

This paper proposes a novel framework for environments, like AIEs, based on embedded intelligent agent technologies capable of increasing their functional performance as well as decreasing the processing latencies originating from the large number of transmitted messages. The framework leads to a decentralized multi society-based architecture where agents discover each other based on advertisement messages and share context information. They facilitate service invocations and perform reasoning and adaptation on their behavior dynamically in order to create ambient intelligence in the environment.

The proposed framework, the Intelligent Association System (IAS), integrates large numbers of agents into society based divisions depending on common objectives, proximities and/or other user-related characteristics and intensions. The intelligent agents operating within the IAS comply with the notions and requirements of AmI environments by utilizing novel techniques for 1) reducing the number of associations and interconnections between the various agents in order to minimize the processing overheads to become more efficient and 2) reducing the cognitive load of programming these associations to personalize themselves to the user needs by learning and adapting to the user's behavior in a non-intrusive mode to preemptively control the environment of his behalf during the lifetime of the system. The proposed intelligent agents are based on Fuzzy Logic Controllers (FLC) which has been credited with providing an appropriate framework for generating human readable models for complex systems [Hagras et al., 2004] [Duman et al., 2007a]. FLCs provide an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance when contending with the uncertainty, noise and imprecision attributed to real world environments as in AIEs. The agent learns the fuzzy model while being able to adapt and self configure itself in a long term learning mode. The fuzzy-based intelligent agents (F-IAS agents) besides learning the behavior of the user perform an online intelligent association evaluation based on a special hebbian-learning algorithm to calculate the association weights (which indicate the relevance between the corresponding agents).

Since the agents within the IAS are organized into societies, they can only discover and associate with agents within their own society. However through the use of *embassador agents* (which are F-IAS agents, mostly with higher computational power and memory storage) the societies are glued together so that associations among different agents of different societies can be discovered and established. In order to find the most relevant associations among the agents, the proposed online intelligent association evaluation mechanism has been extended to suit the society level interaction and calculation of the agents.

The proposed approaches have been tested and verified by two unique experiments. The first experimental set describes how the F-IAS agents learn and adapt to the behavior of a user spending five days within the intelligent Dormitory (iDorm), which is real world test bed for AIE research located at the University of Essex. Moreover, the agents, while learning the behavior of the user, have also successfully limited the number of associations to only those agents of its own society that are relevant and important to their operation, without dropping the overall system's performance. The second part of the experimental work is based on an emulation of a larger scaled AIE with multiple agents and societies. Here the embassadors performed an intelligent association discovery and selection routine to identify potential associations among strongly correlated agents of different societies. Once found, the embassadors notify the agents to establish an association. With this, associations between agents of different societies could have been discovered and selected.

The rest of the paper is organized as follows. Section 2 introduces the Intelligent Association System (IAS) framework and architecture, presents its notions and definitions and explains how the UPnP-Pub/Sub infrastructure can be and is utilized for IAS. In section 3 we introduce the intelligent fuzzy based embedded agent that is capable learning and adaptation to the user behavior in an online and non intrusive fashion during the lifetime of the system. Section 4 explains the proposed intelligent association weight calculation mechanism used by the intelligent agents to select a more relevant set of important associations to maintain (by dropping others) so that the agent processing latencies and overhead can be reduced and the performance increased. This section also describes an intelligent association weight calculation for a novel capability of an embassador that seeks to identify relevant association candidates among separately located agent pairs of different societies. Section 5 discusses the experiments and results. In this section we introduce also the intelligent Dormitory (iDorm) which forms the real world AIE experimental test bed for proposed intelligent agent techniques. Section 6 presents a discussion on related work and section 7 finally presents the conclusion.

2. THE INTELLIGENT ASSOCIATION SYSTEM (IAS)

The Intelligent Association System Framework (IAS) resides on top of a physical network and combines all devices and their services to a decentralized service-oriented overlay network [Duman et al., 2007b]. Service-oriented architectures have been proposed for AIEs [Masuoka et al., 2003] due to their most notable advantage that they facilitate a modular design strategy where applications are built using independent, loosely-coupled pieces of software (here devices and services) that achieve a specific, coarse-grained functionality [Issarny et al., 2005]. It aims to improve the overall effectiveness and efficiency of the agent integration process to automate activities normally performed manually. More information on SOAs applied to AIEs can be obtained from [Issarny et al., 2005] but for the work presented here, it is sufficient to know that any member of the IAS framework is an agent that is composed of services and uses services that are interfaced and presented to the network.

2.1 The IAS Framework

Within an AIE different types of agents may exist and can be explained by looking at their individual purpose and supplied services. A categorization of an agent that is used within an AIE is defined as follows [Duman et al., 2007a]: (1) *Input* embedded agent which is an agent that only provides sensory information (2) *Smart* embedded agent which is proactive by means of executing a set of predefined rules that are stored in the computational logic, e.g. security systems. (3) *Intelligent* embedded agent, which is an agent that is autonomous and includes some sort of reasoning, planning, learning and adaptation processes.

Although an agent is capable of acting on its own to execute tasks, the agents perform more useful and complex behaviors if they collaborate together towards a common objective, thus forming an *embedded agent society* [Duman et al., 2007a], [Duman et al., 2007b]. Grouping agents into societies aims to reduce the complexity of associating and managing large number of embedded agents within AIE, where the IAS will deal with a manageable number of agents within each society.

The embedded agents (hereafter agents) within a society are interconnected via links referred to as *associations*. An association is a physical or virtual communication channel and link between agents, sending information, which is expected to be useful and vital for the corresponding agent. The direction of this exchange is either single or bi-directional. Single-directional associations are mostly set from an input agent to a smart or intelligent agent. If both agents are of a smart type then the association may be set to bi-directional. The lifetime of an association is either permanent or temporary (e.g. mobile or portable agents can join and/or leave the societies at runtime). Furthermore, the associations are selected and established either a) *manually*, where agents are interconnected explicitly by the user (e.g. using a PDA), b) *automatically*, where agents can reason about type compatibility and find agents that are manufactured to work together and c) *intelligently*, where agents are capable of learning and managing their associations autonomously by using intelligent techniques to achieve self-configuration and fault tolerance.

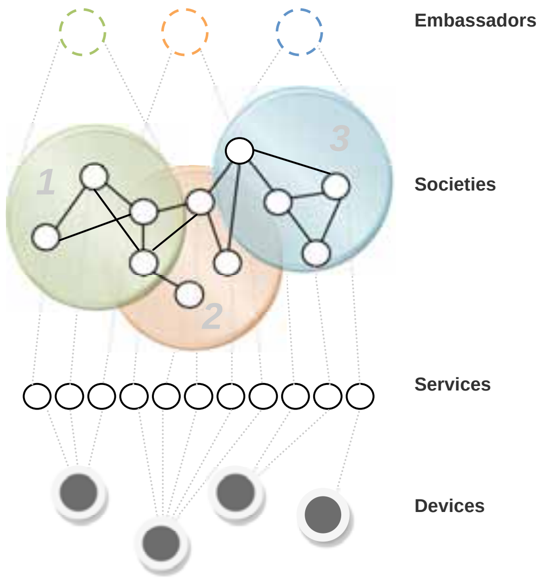


Fig. 1. The Intelligent Association System (IAS) Framework.

The agents associated with each other share services which may be available for several societies at the same time leading to overlapping societies. Every society requiring an inter society communication may have a leader assigned which is termed as the *embassador agent*. An embassador agent may also act as a limited data repository of its society getting information from every agent it is associated with within the same and/or different societies (however it is not like UDDI a registry database that keeps track of the availability of agents). It should be noted that an embassador agent in this context is not a separate unit the tasks of an embassador can be assigned to an existing intelligent agent. The appropriate intelligent agent assigned to be an embassador agent can simply be based on a user selection or an automatic selection procedure can be applied to get the agent with the best computational specifications in comparison to other available agents. As soon as an embassador agent disappears (e.g. break down) a search is initiated to assign a new one. A detailed description of the function of an embassador agent can be found at [Duman et al., 2003] and [Duman et al., 2007a]. The IAS architecture integrates the above components into four layers as shown in figure 1.

2.2 Publish/Subscribe for Society-based Communication

A publish/subscribe (Pub/Sub) system consists of the following components: producers, consumers (subscribers) and events as means of communication between producers and consumers, subscriptions as a standing request and indication of interest in certain notifications, and the event notification service as mediator between producers and consumers of notifications [Eugster et al., 2003]. The mediator (also referred to as message broker within the Pub/Sub community) is responsible for guaranteeing the delivery of events to the consumers that are interested in these events. The delivery of the events depends mainly on the Pub/Sub model that the mediator is part of.

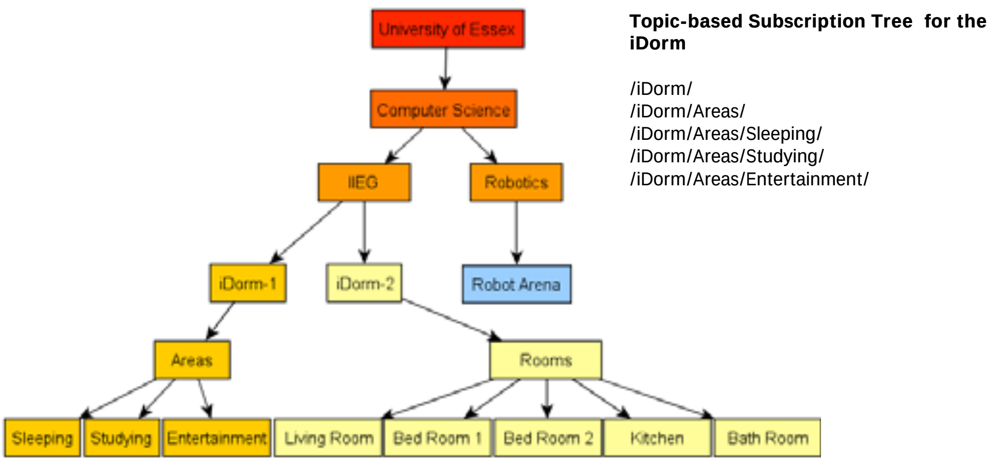


Fig. 2. Topic-based Pub/Sub subscription structure.

2.2.1 *Topic-based Subscription.* The classical Pub/Sub model is based on the notion of topics and subjects [Eugster et al., 2003], which resembles groups or societies (hereafter the term society will be used throughout this paper). An agent *A* subscribing to a topic *T* is regarded as becoming a part of a society *S* labeled as *T*. After a successful subscription, the agent receives all events pertaining to that topic. Topics are annotated with a character string, describing the position relative to the hierarchy this data item belongs to [Eugster et al., 2003]. For example, an agent might publish status information of a ceiling lamp located in the iDorm under the topic "/University of Essex/Computer Science/IIEG/iDorm/Areas/Studying/Ceiling Lamp1". Of course, the topic structure could be reduced to "/iDorm/Areas/Studying/Ceiling Lamp1" if for instance the events are only published at the iDorm level. Each level "down" in the hierarchy describes a finer granularity of notifications and thereby a smaller subset of all event notifications of the system. An example of a topic-based hierarchical structuring of the Department of Computer Science is presented in figure 2. Although the topic names have to be known in advance, the members of each topic are enjoying full *anonymity* so that an agent subscribing to a topic *T doesn't need to be aware* which other agents are member of this topic but would receive every event published to *T* automatically. In other words, the agent automatically subscribes to all agents of the topic *T* without being conscious about their names, types etc. Through this, the overall system becomes *scalable* and *resilient* to changes in the network structure. The justification to use a topic-based Pub/Sub communication infrastructure for AIEs such as the iDorm can be summarized with the following (which also fully conform with the requirements and notions of the IAS):

- Agents do not have to keep a directory of all other agents' presence, location and other functional or non-functional attributes. Many existing communication protocols need to be have these information available beforehand, i.e. existence, address etc. of the agents (and/or their attributes) so that they can request the right information on request. In contrast here, the agents only subscribe to the topics (request membership to a society) they wish to acquire information from and automatically receive the events disseminated from all the other agents subscribed to the same topic.
- Events and messages are routed to the interested agents transparently.
- Pub/Sub middleware infrastructures eliminate the client/server paradigm so that single point of failures are avoided and hence become more resilient and robust, e.g. in the event of an agent failure, the overall system continues operating

without any interruptions and substitutions can be obtained from other agents submitting events to the same topic.

- Overlapping societies (defines different segments of interacting agents of the environment) can be generated through subscribing to multiple topics at the same time, meaning that these agents are part of many societies simultaneously.

2.2.2 *Content-based Subscription.* The content-based Pub/Sub subscriptions abolish the restrictions of a topic-based Pub/Sub model that an event belongs to a particular society and thus to a specific topic. Instead, the decision on how to route the event to the interested agents is mainly done on a message-by-message basis, e.g. based on a query or predicate issued by a subscriber [Eugster et al., 2003]. The advantage of this model in comparison to the topic-based approach is its flexibility. The content-based subscription is regarded as a filtering mechanism based on event properties. Such properties can be internal attributes of data structures carried by event messages in form of e.g. meta-data associated to events, similar as provided in the Java Message Service (JMS) framework [Eugster et al., 2003]. The main disadvantage of this model is the burden it places on the underlying system to match huge amounts of messages to the subscriptions [Eugster et al., 2003]. With this model, every agent of a Pub/Sub messaging infrastructure receives *every* message published in the network and need to perform an analysis and filtering routine to determine if the event matches the desired criteria.

2.3 Embassadors as Pub/Sub Mediators

Agents with more than one *unique* topic subscription are considered as multiple society members. This leads to the creation of overlapping societies as illustrated in figure 3. As a result, these agents publish events to many topics at the same time. Among these intersecting agents, the IAS selects the embassadors for each society so that the societies are interconnected with each other.

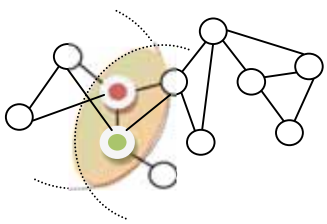


Fig. 3. Overlapping societies as a result of multiple topic subscriptions

The embassadors (besides operating as intelligent agents that seek to optimize the number of their own associations) are also assigned to be the mediators between multiple societies in order to avoid local message flooding of the societies. An embassador agent has filtering and routing mechanisms installed which only forwards events coming from external societies to the agents of its own society, if and only if they are requested, required and/or assumed to be useful. The embassador makes use of both the topic-based and content-based Pub/Sub subscriptions. The message filtering is performed using the content-based Pub/Sub subscription and the event forwarding handled by the topic-based subscription of the Embassadors. In addition to the filtering and forwarding capabilities, embassador agents are capable of generating a topic *on-the-fly* and informing the corresponding agents to subscribe to the newly created topic so that a "private" and personalized communication area can be established. The main purpose of this is to decrease the number of messages multicast to all agents of a certain society. Embassadors are the most crucial entities of an IAS-like system as they are expected to reduce the number of messages reaching their societies by filtering out "redundant" information and blocking their routes. At the same time they need to discover agents from other societies that might be of high interest to the agents of their own society.

3. THE INTELLIGENT FUZZY-BASED IAS AGENTS (F-IAS)

The F-IAS agents are based on the Fuzzy Logic Controller (FLC) provide an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance when contending with the uncertainty, noise and imprecision attributed to real world environments as in AIEs. The agent learns the fuzzy model and its corresponding rules in a non-intrusive manner while being able to adapt and self configure itself in a lifelong learning mode.

3.1 The Fuzzy Controller

The F-IAS agent *perceives* the environment through the information provided by the associated agents subscribed to topic *T* and it affects the AIE through its actuator based on its learnt fuzzy logic controller that approximate the particularized preferences of the user [Duman et al. 2007b]. The F-IAS agent comprises of two stages: 1) offline monitoring and learning and 2) online adaptation and actuation. In 1) the F-IAS monitors the interaction of the user with the environments and collects data of *K* input-output data pairs. Each vector datum $(\vec{x}^k, y^k)$ can be expressed as $(x_1^k, x_2^k, ..., x_N^k; y^k)$, with $\vec{x}^k \in \Re^N, y^k \in k = 1,2,...,K$.

The fuzzy system rule base comprises of a set of L IF-THEN fuzzy rules where the i^{th} rule is having the following form:

$$R^i : \text{ IF } x_1 \; is \; A_1^i \text{ AND } x_2 \; is \; A_2^i \text{ AND ... AND } x_N \; is \; A_N^i \text{ THEN } y \; is \; B^{i*} \tag{1}$$

where N is number of the input variables of the agent where each variable x_j is represented by V fuzzy sets. The variable y represents the output of the agent and it is represented by a Gaussian fuzzy set B^{i*}.

The F-IAS Agent controller use singleton fuzzification, *max-product* inference method and the height defuzzification, so the crisp output of this controller can be written as follows [Kosko 1992]:

$$y = \frac{\sum_{i=1}^{L} w_i \overline{B}^{i*}}{\sum_{i=1}^{L} w_i} \tag{2}$$

where $\overline{B}_i^*$ is the centre of the output fuzzy set of the i^{th} rule and w_i is the rule firing strength which is equal to the product of the membership functions for each rule inputs.

During 2) F-IAS Agents requires the agents to have an effective, fast and reliable adaptation method that can generate new, adjust and/or remove existing rules in the rule base. The rule induction method of the F-IAS for learning and adaptation is described next.

3.2 The F-IAS rule induction method

The F-IAS agents' rule extraction method is adopted from the [Wang 2003] which enables every F-IAS agent within the AIE to learn the model and its behavior through interacting with the user. The procedure involves the following steps to obtain the model of the system [Wang 2003] of a F-IAS agent aimed at subscribing and obtaining information from one or more Pub/Sub topics (societies). A more detailed description can be obtained from [Duman et al. 2007a]

I. Establish associations to N agents which are member of the desired *topic* T or societies $\vec{T}$. Here N can vary depending on the capability and resources and F-IAS agent can provide. In this paper, it is assumed that N is provided by the hardware manufacturer of the F-IAS agents which at the same time describes the limit of the number of association an agent can have.

II. Monitor the user's interaction with the associated agents which can be inferred from the information published to $\vec{T}$. Cache this information in a local storage.

III. Assign for each input supplied by a input agent a set of fuzzy membership functions. The fuzzy membership functions for each agent within the environment are obtained through [Doctor et al. 2005].

IV. For each data pair (x^k, y^k), compute the membership values $\mu_{A_j^q}(x_j^k)$ for each fuzzy set $q = 1,...,V$, and input $j = 1,...,N$, find $q \in \{1,...,V\}$, such that $\mu_{A_j^q}(x_j^k)$ is maximum. The following is the rule generated by (x^k, y^k).

$$\text{IF } x_1^k \textit{ is } A_1^q \text{ AND } \ldots \text{ AND } x_N^k \textit{ is } A_N^q \text{ THEN } y \textit{ is } y^k \tag{3}$$

V. Repeat *Step V* for all *k* from *1,...,K* to obtain *K* data generated rules in the form of Equation 3. Divide the resulting rules into groups (*conflicting rules group*) sharing the same IF part (antecedents) and having different consequents. Combine the group *l* with K_l rules into a single rule in the form of Equation 1 where B^{i^*} is a Gaussian fuzzy set. The antecedent and consequent of the obtained rule becomes the following form

$$\text{IF } x_1^k \textit{ is } A_1^q \text{ AND } \ldots \text{ AND } x_N^k \textit{ is } A_N^q \text{ THEN } y^{k(av^l, \sigma^l)} \tag{4}$$

with the *consequents' average* av^l and *variance* σ^l computed as follows:

$$av^l = \frac{\sum_{k=1}^{K_l} y_k^l w_k^l}{\sum_{k=1}^{K_l} w_k^l} \tag{5}$$

$$\sigma^l = \frac{\sum_{k=1}^{K_l} \left| y_k^l - av^l \right| w_k^l}{\sum_{k=1}^{K_l} w_k^l} \tag{6}$$

where w_k^l is the rule weight of each conflicting rules within group *l* and is computed as

$$w_k^l = \prod_{j=1}^{N} \mu_{A_j^q}(x_j^k) \tag{7}$$

VI. Repeat this combination for all conflicting groups l to obtain the final rule set which contains L rules in the form of Equation 1 and store it in F-IAS Agent's *Rule Base*.

The rule base will ultimately consist of different types of rules: *static rules* and the *dynamic rules*. The static rules encode the fixed requirements of the system that should not be changed which is mostly specified by an administrator (or expert user) of the AIE. Furthermore, static rules can also be a substantial part of the function of an agent that is provided by the manufacturer to meet its purpose. The fixed rules are of special importance for safety and privacy issues e.g. a smoke detector should in any case of smoke detection activate the alarm. This rule is of life importance and shouldn't be allowed to be adjusted.

Dynamic rules are manly related to the comfort and preferences of the user and directly learnt from the user as described above. The number of rules extracted is limited to the number of training input-output data pairs K and does not depend on a fuzzy partition resolution level (i.e. the number of fuzzy sets) [Wang 2003]. During the online actuation and adaptation, the rule induction procedure of F-IAS agent's allows the rule base to be adaptive in a lifelong learning mode so that new rules may be inserted or existing rules may be modified or deleted.

Before the F-IAS can extract rules with this method, it has to have all the information about its associations and collected data. However, the objective of an F-IAS agent is also to reduce the large number of possible associations to a subset of the most important and effective associations without significantly reducing the agent's ability to model the user behavior.

4. THE INTELLIGENT ASSOCIATION DISCOVERY AND SELECTION IN MULTIPLE SOCIETIES

Many methods exist for learning and calculating the significance and importance of associations between input-output pairs e.g. in NN [Haykin 1998] and GA [Trajan 2006], however most of them are computation intensive and mainly rely on huge data sets. Furthermore, they require a long learning period and thus are not suited for online real time intelligent agents.

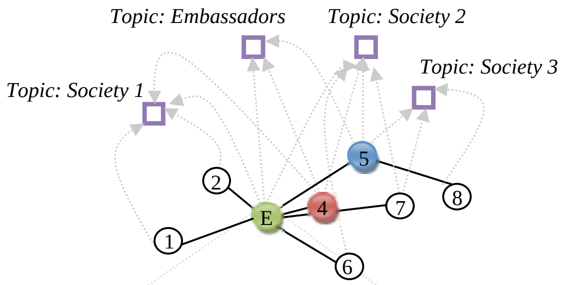
The proposed method used to evaluate the importance of associations among agents in this paper is similar to the notions of directed graphs which are also applied in Fuzzy Cognitive Maps (FCM) [Kosko 1992]. The calculation of the causality/association strength is conducted using a function similar to hebbian-learning [Haykin 1998]. The proposed method is *event-driven* and can be applied in an online fashion. The association strength increases when a simultaneous event (interaction) of associated agents occurs. The higher the weight of an association the stronger is the importance of the agent.

4.1 The intelligent association weight calculation at the individual agent's level

An agent *A* subscribed to a society (or topic *T*) is described as a directed graph where edges indicate the association of *A* to other agents of *T* and the direction of the edge indicates the message flow (e.g. from an input agent to a F-IAS agent). The edges strength expresses the causality which can be calculated, updated and evaluated real time. In addition in the event of a new agent joining or an existing one leaving *T*, the graph can adjust itself in an ad hoc fashion to accommodate the change without the need to suspend the operation of the system. These characteristics show that directed graphs in general are well suited for determining the relevant associations among agents and thus comply with the high-level concept and notions of the IAS architecture.

Since the concepts of the directed graph in this paper embody fuzzy-based agents (whether input or F-IAS agents) and the edges to other agents of the society *T* are referred as associations with weight indications, hereafter we will use the term FCM for the directed graphs which fully conforms with the structural notions of IAS. As in FCMs, here the associated agents seek to answer questions like: what happens to agent *A* if an event is performed by Agent *B*, which implicatively describes the causality between *A* and *B*.

Figure 4 illustrates the use of FCMs within the IAS framework. Here, the *embassador E* (*Agent 3*) is a member of *Society 1* and *Society 2*. This means that *Agent 3* has established explicit associations to the other agents subscribed to the same topics which include *Agent 1, Agent 2, Agent 4, Agent 5, Agent 6* as well as *Agent 7*. The causal strength of the associations indicates a value between 0 and 1, where 0 indicates no association and 1 a maximum causal effect on the agents.



A1	A2	A4	A5	A6	A7
0.1	0.4	0.3	0.4	0.9	0.7

Fig. 4. The structural concept of fuzzy cognitive maps for F-IAS agents in a Pub/Sub infrastructure

The following illustrates the steps for the intelligent association selection and learning which is based on the weight calculation:

1. Initiate $FCM(Y)$ for the F-IAS Agent with the agents and establish associations to N input agents ($X_j \rightarrow Y$). These agents may have been selected by the user, randomly or automatically, e.g. based on proximity within the same agent societies or topics $\vec{T}$.
2. Set the association weights $\alpha_{X_j \rightarrow Y}$ of $FCM(Y)$ to zero, for all associated agents j=1…N so that the association matrix of F-IAS becomes $\Lambda_{X_j \rightarrow Y} = [X_1 = 0, X_2 = 0, X_3 = 0 \ldots X_N = 0]$.
3. Set the learning rate δ = 0.1.
4. Initialize the pre-associative (ξ_{X_j} = 0) and post-associative (ξ_Y = 0) flags for the agent pair $X_j \rightarrow Y$.
5. In the case of a published state change to topic T of an associated agent X_j the information x_j^k is forwarded to the F-IAS.
    a. For each event update the pre-associative flag of the corresponding agent ξ_{X_j} to 1
    b. Calculate the resulting output of the F-IAS Agent caused by the event x_j^k by applying Equation 2

i. Update ξ_Y to 1 only if the F-IAS has adjusted its output state due to event x_j^k

c. Calculate the new association weights $\alpha_{X_j \to Y}$ for each agent pair $X_j \to Y$ at time *k* by applying the following equation:

$$(\alpha_{X_j \to Y})^k = (\alpha_{X_j \to Y})^{k-1}(1-\tau) + \delta \xi_{X_j} \xi_Y \quad (8)$$

where $(\alpha_{X_j \to Y})^{k-1}$ is association weight and τ is the decay value which is set to 0.01. The reason for adding a decay value is to prevent the association calculation increasing endlessly. It is clear that an F-IAS Agent where its associations can only increase is bound to be useless and misinterpreted. An association which was important for the F-IAS at the beginning might become redundant over time and without decrease in the association weights this would never be noticeable. The decay value 0.01 has been derived by trial-and-error experimentation and suited the F-IAS Agent most.

d. Reset ξ_{X_j} and ξ_Y to 0

e. Repeat *Step 5* continuously until a given time *k*. Time *k* for the F-IAS Agents is set to *forever* until the agent suspends.

The algorithm presented calculates the importance of associations between the F-IAS and the interconnected input agents. With the increasing number of simultaneous state changes, the association weight also increases.

Another major issue that needs to be addressed is the frequency of use of the agents. It is obvious that the use of various agents differs according to their functionality and purpose. Additionally different agents provide different information. For example a chair pressure sensor only publishes an event if someone sits on it or stands up, whereby a temperature sensor continuously measures the temperature of the environment and regularly multicast the events. By using the algorithm above it is clear that the fewer events a service transmits, the less likely it will be regarded as a strong association. To overcome this situation, the following procedure aims to normalize the association weights according to their frequency of use, so that a better judgment of the importance between all associated agents can be performed.

- *Step I*: While running the above intelligent association weight calculation, count the total number of events $Count\left(x_j^k\right)$ for each agent X_j.
- *Step II*: For each X_j, apply the following equation to obtain the normalization constant value

$$\gamma_{X_j} = \left(\frac{Count(x_j^k) - \overline{Count(x_j^k)}}{1/\dfrac{K}{Count(x_j^k)}} \right) \tag{9}$$

- *Step III*: Assign γ_{X_j} to the following equation which forms a sigmoid function with a normalized constant value for each service

$$f_{sig}(\alpha_{X_j \to Y}) = \frac{1}{1 + e^{-\alpha_{X_j \to Y} * \gamma_{X_j}}} \tag{10}$$

- *Step IV*: The above equation generates normalized association weights so that an equal and fair judgment on the importance of association between the agents and the F-IAS can be performed.
- *Step V*: Insert the obtained $\alpha_{X_j \to Y}$ into the association matrix Λ of the F-IAS Agent.

If $f(\alpha_{X_j \to Y}) \geq \Theta$ (where Θ is a predefined threshold e.g. 0.15) then the association is of great importance to the F-IAS agent. On the other hand if the association weight of a F-IAS agent is smaller than the threshold Θ than this agent may be considered as irrelevant or redundant and would become a candidate to be removed.

During the operation of the F-IAS agents, the intelligent association weight calculation mechanism constantly seeks to reduce irrelevant associations to agent and simultaneously evaluates new and potentially more relevant and significant agent that will maintain the fuzzy model's quality while decreasing the overall agents computational loads.

4.2 The multi-society based intelligent association discovery and selection

So far, the methods presented in this paper on intelligent association weight calculation assumes that the F-IAS agents know the topics that they need to be subscribed to in order to receive the events published by the agents which are member of these topics. Multiple topic subscriptions increase also the chance to find relevant associations to agents of other societies. However the questions that is legitimate in this case is which societies contain the most relevant agents and how can they be discovered without experiencing messaging flooding caused by too many topics as this would immensely increase the F-IAS agent's processing overhead. The answer resides in how embassador agents are employed in the IAS infrastructure. The embassadors' can be used for 1) filtering and forwarding information between societies and 2) 'interconnect' agents of different societies and make them discoverable and selectable for agents residing in various societies. Without the embassadors described capabilities societies can be regarded as isolated regions of the environment where agents have a limited view of the happenings as shown in figure 5.

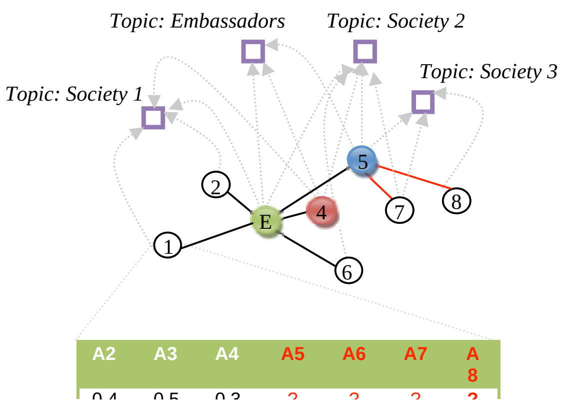


Fig. 5. Agent 1's limited view of the environment

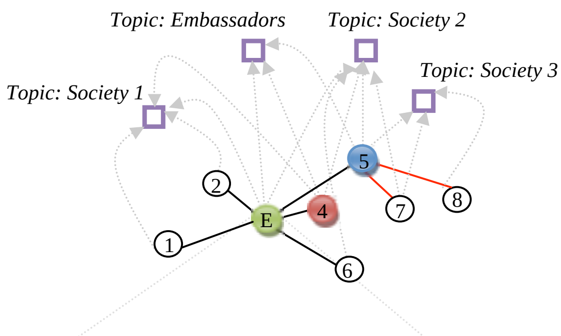
As it can be seen, *Agent 1*'s association matrix would only contain agents of its own society (here *Agent 2*, *Agent 3*, and *Agent 4*). The only possibility to extend their "view" is to know the topics which disseminate events from agents subscribed to them or through the embassadors.

The embassador enables multi society based intelligent association discovery and selection which uses in principle the same methodology for intelligent association weight calculation among potential association candidates as described in section 4.1. The main differences are incorporated in the following steps:

1. Initiate a *two*-dimensional $FCM(E)$ for the embassador E and subscribe to the topics $\vec{T}$ (e.g. *Society 1* and *Society 2*)

2. For every agent X_j of $\vec{T}$ add a column and a row in $FCM(E)$. Figure 6 shows an example of the $FCM(E)$ with Agent 2, Agent 4, Agent 5, Agent 6, and Agent 7 as members.
3. Set the association weights $\alpha_{X_j \to X_m}$ of $FCM(E)$ to zero, for all associated agents *j*, *m* =1…*N*
4. Run *step 3 to 5 continuously* of the intelligent association weight calculation as described in *section 4.1* to calculate the association weight $\alpha_{X_j \to X_m}$ for all agent pairs $X_j \to X_m$.
5. At time *t* apply the Θ (which is here a predefined value of 0.5) to obtain the most significant association among agents of separate societies. If the calculated weight is greater than the threshold Θ (as which is the case for $X_2 \to X_6$, $X_4 \to X_6$ and $X_4 \to X_7$) than these association are considered as important. In contrast, if the weights after a certain period remains less than Θ the embassador can regard these potential associations as irrelevant so that it can stop monitoring and evaluating of this specific agent pair.
6. For each agent pair $X_j \to X_m$ above the giving threshold Θ, generate a new topic $T_{X_j \to X_m}$ (e.g. $T_{X_2 \to X_6}$ labeled as "Society $X_2 \to X_6$") and invite the pair to subscribe to this newly generated topic so that an association among them can be established.
7. For each agent pair $X_j \to X_m$ below the threshold start a process to remove them from the $FCM(E)$ and also as them to unsubscribe from the $T_{X_j \to X_m}$.

The proposed algorithm operates in real time so that the embassadors can initiate discoveries for relevant agent pairs of located in different societies and as a result select and recommend the most appropriate ones by generating custom topics where selected agents can communicate. With this approach, agents even in the most "distant" societies can be discovered and presented for the intelligent agents. For instance, in order to propagate events from $X_4 \to X_7$ many embassadors needed to cooperate so that the communication route of the events can be identified as $X_4 \rightarrow E(X_3) \rightarrow E(X_5) \rightarrow X_7$. The assumption here is that the societies that the embassadors are representing intersect so that every embassador agent is interconnected with its "neighboring" embassador. If this is however not provided (e.g. isolated societies with no coexisting members) than the embassador or the intelligent agents need to have a prior knowledge of the various agents and societies so that matching topics can be generated.



	X2	X4	X5	X6	X7
X2	1	0.2	0.1	0.8	0.2
X4	0.2	1	0.3	0.7	0.6
X5	0.1	0.3	1	0.1	0
X6	0.8	0.7	0.1	1	0.4
X7	0.2	0.6	0	0.4	1

Fig. 6. The association matrix of embassador *E*

5. EXPERIMENTS AND EVALUATION

This section describes the experimental systems used to evaluate the proposed concepts and methods. The experiments are based on two different platforms. The first set of experiments was conducted in a real world AIE, the iDorm. Agents (in form of input and intelligent agents that are part of the same society, called the *iDorm* society) were constructed to interact with the user to learn his/her behavior while optimizing their set of associations to other agents to only include the most relevant associations. The main ambition was to demonstrate and validate the significance of the intelligent association weight calculation algorithm applied to the individual intelligent agents as presented in section 4.1.

The second part of this section presents experimental results obtained from an emulation of an iDorm-like AIE environment. Since the amount of agents placed within the iDorm is limited and currently based on a single room (represented as a single society), the emulation allowed us to create a more sophisticated AIE with multiple societies (e.g. rooms) where more associations for agents can be discovered and evaluated using the intelligent association weight calculation algorithm. The emphasis of this set of experiments is set to the embassadors unique capabilities to monitor and evaluate the events multicast to the societies to extract potential candidates for creating associations among agents of different societies.

5.1 The intelligent Dormitory – A test bed for AIE research

The intelligent Dormitory (*iDorm*), as shown in figure 7, forms the main test bed for the AIE research described in this paper. The iDorm is fitted with a plethora of embedded sensors, actuators, processors and heterogeneous networks that are cleverly concealed (buried in the walls and underneath furniture) so that the user is completely unaware of the hidden intelligent infrastructure of the room. The iDorm looks and feels like an ordinary study/bedroom environment containing a mix of furniture such as a bed, work desk and wardrobe. This splits the room into areas of different activity such as sleeping, working and entertaining [Holmes et al., 2002] [Callaghan et al., 2004].

The iDorm provides an AIE that is ubiquitous, transparent and intelligent. The iDorm is ubiquitous because the user is surrounded by a multitude of interconnected embedded systems and transparent since the artifacts are seamlessly integrated into the environment [Duman et al., 2007b]. The agents will provide the intelligent ‘presence’ as they are able to recognize the users and can autonomously program themselves to the users’ needs and preferences by learning from their behavior to control the environment on their behalf [Duman et al., 2007b].

The agents are embedded within the various artifacts and listed as follows:

- The input agents are embedded in the following sensing devices of the iDorm: Internal Light Level sensor (ILL), External Light Level sensor (ELL), Internal Temperature sensor (ITEMP), External Temperature sensor (ETEMP), Chair Pressure sensor (CHAIR), Bed Pressure sensor (BED) and Clock (HOUR).
- The intelligent agents are embedded in the following actuating devices of the iDorm: Desk Lamp (Desk Lamp), Bed Lamp (Bed Lamp), Dimmable Ceiling Lamp 1 (DIM1), Dimmable Ceiling Lamp 2 (DIM2), Dimmable Ceiling Lamp 3 (DIM3), Dimmable Ceiling Lamp 4 (DIM4).

The iDorm combines four networks platforms which are LonTalk, Tini 1-wire, IP and X10. This provides a diverse infrastructure and allows the development of network independent solutions. It also gives an opportunity to evaluate the merits of each network. The iDorm gateway server creates a common interface to the iDorm and its devices that are based on the Universal Plug & Play (UPnP) which is an event based communication middleware for allowing devices to be plug & play enabling automatic discovery and configuration [Holmes et al., 2002]. These events can be in the form of state changes, new arrival or removal notification of devices.

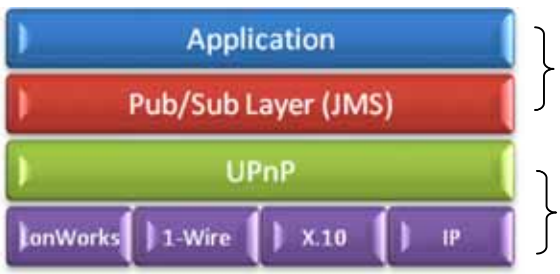
Fig. 7. The University of Essex intelligent Dormitory (iDorm)

5.1.1 *The Limitations of UPnP*. In UPnP, network devices periodically advertise their presence by multicasting to the subscribed devices. Multicasting is also used for discovery requests. It is obvious that such a system is *neither scalable nor effective* as the network has to deal high numbers of multicast traffic. In addition, the flat structure of UPnP limits the presence and discoverability of the agents so that they can advertise themselves and discover other (known) agents of the same subnet only. Hence, agents attached to other (sub-) networks or societies are invisible to these agents and cannot be explored. A solution to overcome this problem is to employ a mediator or bridges between each subnet so that advertisement and discovery messages are being passed automatically to each other. However, these subnets and their agents need to *be known in advance* so that subscriptions to services of different subnets can be established and as a result messages communicated.

This however cannot be assumed nor provided especially in dynamic ad hoc environments such as AIEs where agents may join or leave a subnet (or a society of agents) in an unpredictable manner. For this reason, the conventional *request-receive* style communication as offered by UPnP is not sufficient and is likely to fail due to the lack of knowledge of existing agents about the environment.

5.1.2 *Merging UPnP with Pub/Sub.* To provide both UPnP-based event notification and topic/society-based subscription, the following simple changes need to be done to enhance the UPnP framework with Pub/Sub capabilities. Figure 8 illustrates the layers of the extended layered communication infrastructure based on combination of UPnP and JMS.

The first two layers (bottom-up) of the communication architecture remain unchanged. The third layer replaces the UPnP point-to-point subscription layer with the Pub/Sub layer. Here, JMS has been chosen to provide the Pub/Sub infrastructure. JMS is an API specification, being part of the Java2 Enterprise Edition (J2EE) [Thomas 2007] and can be put on top of many industry messaging and Pub/Sub products, including Corba, Jini etc.



This layer defines the topic-based Pub/Sub overlay network where agents of the upper layer can establish or dissolve subscriptions to topics (societies). The agents only need to be aware of the names of the topics so that they can subscribe to them. Any lower-level connections to the physical devices are handled by the UPnP layer.

The UPnP layer provides the interface to the physical device and presents them as agents to the Pub/Sub (JMS) layer. This layer handles also the event triggering of the devices, so that any change of state of the devices are published as event into the upper layer which then is multicast to the associated societies/topics.

Fig. 8. The extended communication infrastructure based on combination of UPnP and JMS.

The following simple steps and changes are conducted to integrate JMS to the UPnP architecture:

- Every agent needs to be interfaced to the iDorm’s UPnP middleware
- Every agent produces a local message broker and subscribes to a topic (e.g. society name, device name, functional description). This subscription *differs* to the previous UPnP-based subscription since the agents now do not have to conduct *discovery and subscription* routine rather they subscribe directly to a topic where other agents are considered to become member of. However, the agents need to be informed about topic names beforehand or can perform a search to discover them.

- The eventing procedure of the UPnP protocol is used without any changes. Whenever a state change within the agents occurs (e.g. a switch is set to on), UPnP generates and triggers an event that is then multicast to the *topic* so that every member agent of this topic receives it as well.

With this simple but efficient transformation of an UPnP infrastructure to support Pub/Sub-style subscriptions and event dissemination mechanism the overall system becomes more scalable as it alters UPnP’s flat point-to-point communication into a hierarchical structure. It becomes more robust and resilient due to the dynamic nature of Pub/Sub enabling *push*-based ad hoc device discovery and event broadcasting. If an agent disappears it no longer will publish its presence to the topic so that it removes itself automatically without blocking the systems operation.

5.2 Intelligent association learning in the iDorm

We conducted several experiments within the iDorm. In this paper we will present a subset of these experiments where a user stayed within the iDorm for 5 consecutive days. It is worth mentioning that all the experiments and results reported in this section are repeatable for the various users that stayed in the iDorm. We will demonstrate that after the F-IAS agent learns from the user’s interaction and adapts to his behavior within the iDorm and reduces the associations to other agents that even if a small number of associations are omitted the agent processing overheads will decrease significantly. In addition, we will also explain that the system performance increases as the number of fuzzy rules of the F-IAS agent decreases thus leading to a more robust and efficient operation with less processing and rule storage requirements.

5.2.1 Experimental Setup. The experiments presented in this subsection have been carried out in the iDorm using the following embedded agents:

- *Input Agents* which were embedded in the following devices: Internal and external light level sensors (ILL, ELL), internal and external temperature sensors (ITEMP, ETEMP), chair and bed pressure sensors (CHAIR, BED), and clock (HOUR)
- *F-IAS Intelligent Agents* which were embedded in the following actuating devices: desk lamp (DESKLAMP), bed lamp (BEDLAMP), 4 independent ceiling lamps (DIM1, DIM2, DIM3, DIM4).

It should be noted that during the experiments, all the input agents were run on separate and independent hardware processing units whilst the UPnP stacks and intelligent agent mechanisms were run on a single PC as software multi-processes. The hardware processors used in this experiments were small (20MHz, 0.5MB RAM processors) that could not support complex agents and so we utilized the PC as a proxy for these agents (a common technique in distributed embedded-architectures). This approach provides a more flexible experimental structure.

The agents are fuzzy-logic based, where the membership functions of the inputs and outputs of the various embedded agents were obtained from [Doctor et al., 2005], the number of the fuzzy sets and the linguistic labels are listed as follows:

- *ILL, ELL, ITEMP, ETEMP, HOUR, DIM1-4* each consisting of 7 fuzzy regions and labeled as "*vvlow, vlow, low, med, high, vhigh, vvhigh*" respectively
- *CHAIR, BED, DESKLAMP and BEDLAMP* each consisting of 2 fuzzy regions and labeled as *"on, off"* respectively

All of the agents were organized to form the *iDorm* society. For this every F-IAS agent initiated an association request using UPnP to subscribe to every other agent within the same society in the form of *e(sub(∂), iDorm),* where *e* is the *subscription event* sent out from by the agents, ∂ the obtained subscription ID after a successful association and *iDorm* indicating the label of the society they belong to. After the agents submit their requests, a *new* Pub/Sub topic *iDorm* representing the *iDorm* society is initiated where all subscribed agents of this topic become interconnected with each other so that every event published within this society *e(pub(∂), iDorm)* is multicast to the agents.

5.2.2 Results. For simplicity, the DESKLAMP F-IAS agent (hereafter only F-IAS agent) is used to explain the results. During, the first 3 days the F-IAS agent monitored the user and collected the data based on events *e(pub(∂), iDorm)* published to the *iDorm* topic. At the end of the third day the F-IAS agent extracted a total of 297 rules from 400 collected data sets using the presented rule induction method as described in section 3.2. This formed the initial fuzzy rule base which resulted in a fuzzy model that approximated the user behavior with a Normalized Mean Squared Error (NMSE) of 0.0108. The processing time obtained for each F-IAS agent to go through a single control cycle for the 400 data sets was in average 4797ms. An agent processing latency criterion (APLC) was introduced to measure the F-IAS agent's processing load caused by published events *e(pub(∂), iDorm)* and F-IAS agent's own local processing of the fuzzy controller. Initially, the APLC for this experiment was 3220ms.

The F-IAS agent is then transformed into the actuation mode where it seeks to support the user by acting on behalf of them based on the knowledge acquired during the monitoring stage while trying to improve APLC by evaluating the effectiveness and significance it's associations.

The following example initially explains the results obtained from the intelligent association learning of the F-IAS agent (DESKLAMP) and 3 intelligent agents, the internal light level (ILL) and temperature sensors (ITEMP) and the chair pressure sensor (CHAIR) based on the *iDorm* society, in order to present a better understanding of the system's functionality.

At every simultaneous change of the F-IAS agent and the associated agents, the association weight $\alpha_{X_j \to Y}$ increases. In contrast, the association weight decreases if the occurred event doesn't have an impact on the F-IAS agent. The association weight was calculated over a period of 2 consecutive days where the F-IAS agent operated in an online fashion. The resulted association weight of the F-IAS agent and the CHAIR was calculated as 0.629. The strength of the association between the F-IAS agent and the ITEMP and ILL are 0.034 and 0.15 respectively. Depending on the threshold Θ (which was initially set to 0.15 for this experiment), the F-IAS agents would request the removal of the association with the ITEMP agent.

With the removal of the ITEMP agent, the rule base of the F-IAS Agent decreases to 195 rules which is a 34% saving on the memory storage. With the removal of the association to the irrelevant ITEMP agent by sending an unsubscribe event *e(unsub(∂), iDorm)*, the F-IAS agent doesn't need to listen to events coming from the ITEMP anymore and put it in the "ignore list". There is no need to process them any longer which saves computational resources and increases the robustness and efficiency of the F-IAS agent. The results of the intelligent association weight calculation for simultaneously running F-IAS agents within the iDorm are listed in Table I. The F-IAS agents include DIM1, DIM2, DIM3, DIM4, DESKLAMP and BEDLAMP.

	ILL	ELL	ETEMP	ITEMP	Chair	Bed	Hour
DIM1	0.24	0.25	0.02	0.09	0.35	0.21	0.28
DIM2	0.27	0.34	0.04	0.03	0.23	0.27	0.32
DIM3	0.16	0.12	0.02	0.02	0.23	0.13	0.19
DIM4	0.42	0.38	0.07	0.04	0.32	0.28	0.25
DESKLAMP	0.15	0.31	0.04	0.03	0.63	0.26	0.29
BEDLAMP	0.23	0.12	0.02	0.02	0.16	0.56	0.34

Table I. The association weight matrix for all of the F-IAS Agents in the iDorm.

After applying the threshold Θ (0.15), which can be different for each F-IAS agent depending on the limitations of resources in processing, network connections and memory, the most relevant associations are selected and irrelevant ones removed. The threshold is set to 0.15 for all F-IAS agents in this experiment. Table II depicts the association weight table of all of the F-IAS agents after removing the associations. It is clear that all of the F-IAS agents have made a reduction in their associations and formed their own society of agents (e.g. DIM1 F-IAS agent's society as depicted in figure 9). The most drastic change happened to the DIM3 F-IAS agent which initially started with 7 associations and gradually omitted 5 of them during the intelligent association calculation process.

	ILL	ELL	ETEMP	ITEMP	Chair	Bed	Hour
DIM1	0.24	0.25	--	--	0.35	0.21	0.28
DIM2	0.27	0.34	--	--	0.23	0.27	0.32
DIM3	0.16	--	--	--	0.23	--	--
DIM4	0.42	0.38	--	--	0.32	0.28	0.25
DESKLAMP	0.16	0.31	--	--	0.63	0.26	0.29
BEDLAMP	0.23	--	--	--	0.16	0.56	0.34

Table II. The F-IAS association weight matrix after applying the threshold.

The impact on accuracy after removing the above associations from the F-IAS agents in respect to model accuracy was determined by using the NMSE. After removing the 5 association of the DIM3 F-IAS agent the system's prediction accuracy drops from 0.0541 to 0.1339 while at the same time the number of fuzzy rules reduces from 297 to 24. It is obvious that through the removal of the associations already used and perhaps important rules were deleted however the continuous adaptation process of the F-IAS agent allows relearning them. To avoid long adaptation processes the user can set a threshold to prevent a major prediction decrease. The 91% reduction in the rule base results in less need for memory and faster processing for the F-IAS Agent.

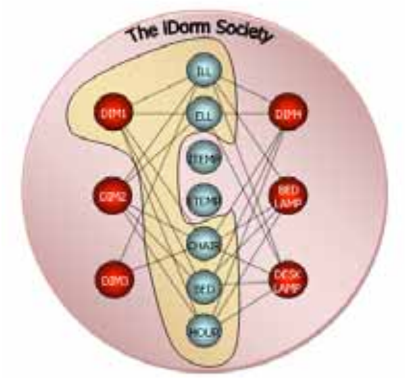


Fig. 9. The DESKLAMP F-IAS Agents Society within the iDorm

5.3 Intelligent association discovery and selection based on an emulation of a multi society based AIE

The emulated environment reflects a multi society based structure (e.g. multiple rooms in a building) where societies (here the amount of them is randomly selected) containing various types of agents (in form of input and intelligent agents) are interconnected through the utilization of embassador agents. Every embassador besides satisfying the needs to improve its own efficiency is concerned to *intelligently* discover and select potential associations between relevant agents that are member of different societies. The emphasis of the following experiments is especially set to demonstrate these unique capabilities of embassadors.

5.3.1 Experimental Setup. The experimental environment is based on a large-scale emulation of an AIE which was developed using Java. Furthermore, the Pub/Sub middleware is constructed using the UPnP communication and Java Messaging Service (JMS) stack. The emulated Pub/Sub-enabled AIE environment is sketched out in figure 10 using the JUNG visualization tool [Jung 2007].The regions indicate randomly selected overlapping societies and each dot represents an agent, which can be either input or intelligent. The societies are “glued” together via the embassador agents who are selected from the available intelligent agents of the societies.

Fig. 10. A multi society based AIE emulation.

Figure 11 show a simplified version (depicting only a section) of the experimental setup. Hereafter, for the sake of clarity, this structure will be used as a proof-of-concept for the proposed intelligent association discovery and selection methodology for embassador agents.

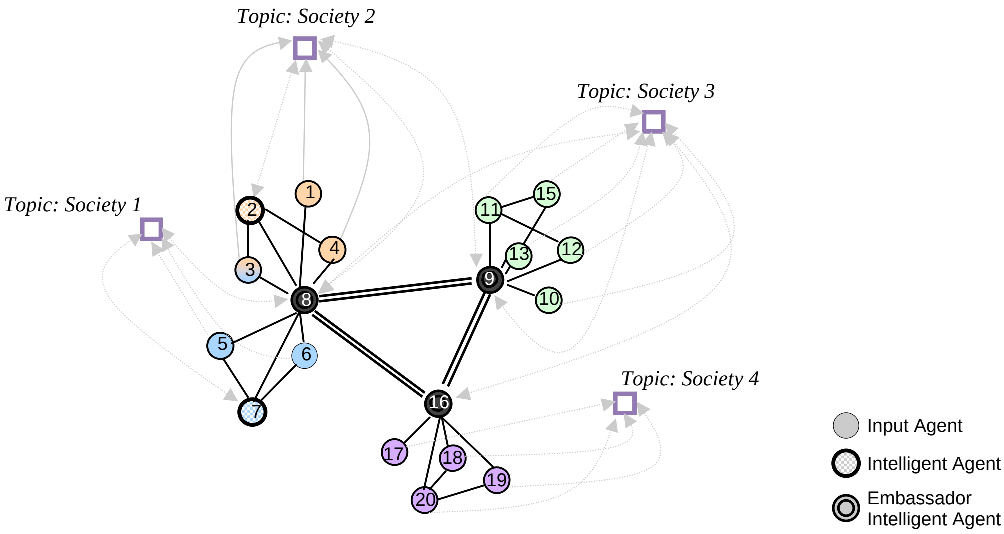


Fig. 11. The simplified representation of the emulated AIE.

The environment is segmented into *four* societies with *three* embassador agents (where *embassador Agent 8* is member of the topic *Society 1* and *Society 2* thus which lead to form overlapping societies). Each society consists of either one or two intelligent agents that are associated with every input agent of its society. The following steps have been initiated to obtain the described experimental setup:

- For every randomly selected *input agent* (which in terms of the Pub/Sub infrastructure is a producer which publishes events) initiate a subscription *e(sub(∂), T)* to a desired topic *T*. For instance, here *Agent 4* subscribes to topic *"/Society2/"* by sending the event *e(sub(∂), "/Society2/")* and accordingly *Agent 3* subscribes to both *"/Society2/"* and *"/Society3/"* after sending out the following events *(e(sub(∂), "/Society2/") and (e(sub(∂), "/Society3/")* respectively.
- As in the previous step, once the intelligent agents have indicated their interest in a certain topic by subscribing to it e.g. *(e(sub(∂), "/Society2/")*, it will, in contrast to input agents, not only be allowed to send events but also receive them. For this, they also need to subscribe to the topic as consumers, *(e(cons(∂), "/Society2/")*
- Each Embassador acts like an intelligent agent, i.e. subscribing and publishing to specific topics, however, they also can talk to each other in a separate topic, e.g. *"/Embassadors/"*. By doing this, different events published by agents of spate societies can be communicated through this channel and directly forwarded to the agent that might be interested in them. Table III illustrates the subscriptions of the *Agent 1- Agent 8* of the given example.

	1	2	3	4	5	6	7	8
Society	Society 2	Society 2	Society 1 & 2	Society 2	Society 1	Society 1	Society 1	Society 1 & 2 & 3
Agent Type	Input	Intelligent	Input	Input	Input	Input	Intelligent	Embassador
Pub/Sub	Publish	Publish/ Subscribe	Publish	Publish	Publish	Publish	Publish/ Subscribe	Publish/ Subscribe
Topics	/Society2/	/Society2/	/Society1/ /Society2/	/Society2/	/Society1/	/Society1/	/Society1/	/Society1/ /Society2/ /Society3/ /Embassadors/

Table III. The topic-based pub/sub subscriptions of Agent 1 – Agent 8.

5.3.2 Results. The main ambition of this experiment was to validate the discoverability of agents operating in a multi society based environments through the use of embassador agents which normally have a more global view of the societies. In other words, every embassador of a society was concerned with finding correlated agent pairs located in separate societies which may ultimately lead to establish association among each other in order to increase their performance and efficiency. For simplicity, the performance and efficiency metrics were defined as follows: Every agent (input or intelligent) triggers an event at random. The objective of the embassadors here was set to find associations among agents that were *simultaneously* multicasting events in different societies. E.g. after observing figure 12, it can be conclude that *Agent 1* is closely associated with *Agent 3* as both of them publish the same amount of events in a given time intervals. Thus, the calculated percentage of simultaneously publishing of a pair of agents is the metrics for calculating the significance of relevant associations among each other.

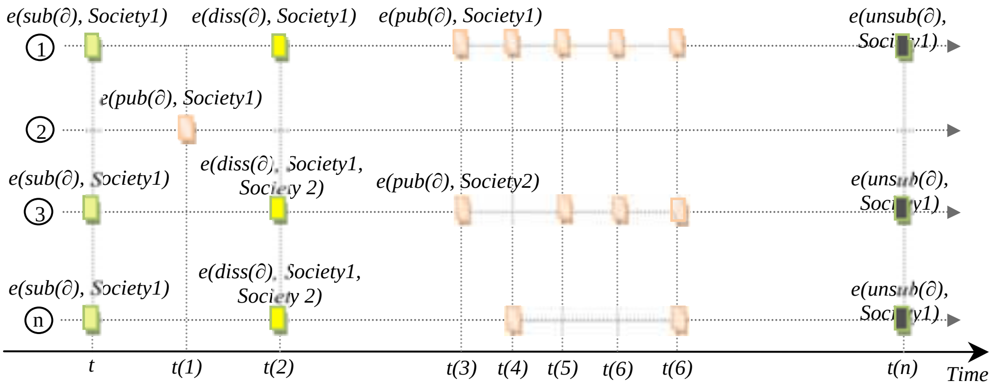


Fig. 12. Events published simultaneously by various agents

After the embassadors have been selected and associations created to the agents of the societies they are subscribed to they start applying the online adaptive intelligent association calculation mechanism, as proposed in section 4.2, to measure the relevancy of the simultaneously publishing agents.

After the agents start sending out messages (here in form of advertisement messages that are randomly multicast), every embassador subscribed to the societies which disseminate these messages as events among the member agents and start evaluating the co-occurrence of the messages and accordingly calculated the association weights. From figure 13 one can see that the embassador (denoted as *Agent 8* in figure 11) over a given time (here 5000 seconds) was able to find some relevancy of about 0.85 (or 85%) between the agent pair $X_2 \rightarrow X_6$. On the other hand, agent pair $X_7 \rightarrow X_{12}$ (see figure 14) although high at times, is regarded less relevant since the obtained metrics after 5000 sec reaches a maximum relevancy of just below 0.5. Depending on the set threshold Θ (e.g. 0.5) this agent pair would be regarded as less relevant and thus dissolved. After an agent pair is removed from the list of the embassador new pairs can be acquired from other societies e.g. thorough the propagation of the events between several embassador agents.

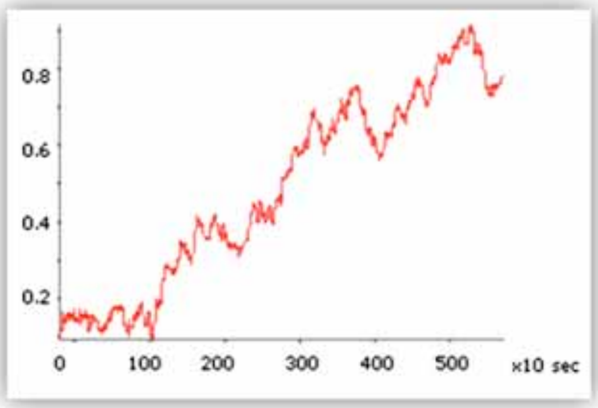


Fig. 13. Relevancy calculation between agent pair $X_2 \rightarrow X_6$.

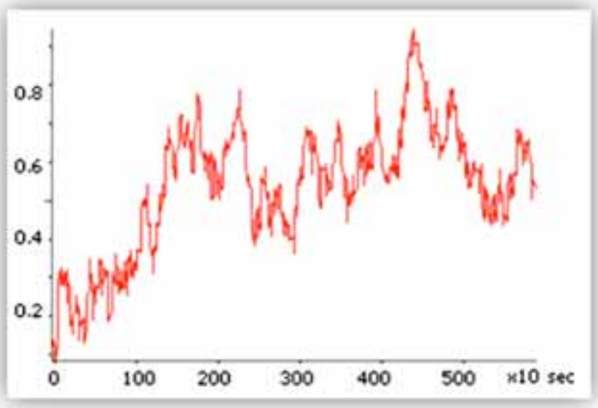


Fig. 14. Relevancy calculation between agent pair $X_7 \rightarrow X_{12}$.

After the embassador recognizes the relevancy among the agent pair it generates a new topic $T_{X_2 \rightarrow X_6}$ for the agents so that they become "visible" to each other and may start exchanging information accordingly.

6. RELATED WORK

The following projects demonstrate the use of intelligent agents within AIEs. The emphasis of most of them is on the individual as numerous projects operate as an AIE consisting of a single embedded agent, mostly in the form of a PC running a software agent, aiming to learn and predict the preferences of the user as well as to operate the environment at the most efficient and effective level. Only a few describe the use of multi-agent system architectures for AIEs with less focus on integrating some sort of intelligent mechanisms to manage the interconnections, essential for AIEs.

The work of [Davidsson 2005] uses multi-agent based approaches for the management and control of devices in AIE and uses methods of recognizing user activity and learning relationships between user preferences and states of the devices in the environment. The approaches do not implement a control system based on the learned particularized behaviors and preferences of the user of an AIE. The approach assume that the preferences are pre-learned or predefined and describe a closed and static multi-agent architecture, where adding new devices requires the reconfiguration of the overall system by an administrator. This is in most cases for the realization of truly AIE with a large number of devices being almost impossible.

In [Mozer. 1999b] an approach was applied to investigate energy reduction in the Adaptive Home. The method uses neural networks based on traditional machine learning theory to control the user's environment. The approach uses objective functions that aim to either derive a minimal control function that satisfies the 'averaged' needs of the occupants or are aimed at prioritizing a number of competing needs such as energy efficiency and user comfort [Callaghan et al. 2004]. The main drawback of this approach is that it requires an iterative offline learning cycle and thus is less eligible for online adaptation since the neural networks require to be retrained. This is very time consuming. In addition, the methods are not appropriate for devices with limited resources as neural networks require a large amount of training data for the classification and learning process, which contradicts with the vision of AIE where the artifacts consist of limited resources.

The particular difference of the ISL approach of the AIE [Hagras et al. 2004a], [Hagras et al. 2004a] to [Mozer. 1999b] and [Youngblood et al. 2005] is that the used agent truly represents a hardware-based embedded agent which is capable of learning the particular behavior of the occupant of the AIE, the iDorm. The ISL implements an adaptive online learning methodology that learns from the user through interaction with the environment. The ISL agent is of incremental nature meaning that newly obtained data or rules can easily be added into the existing system in an online fashion. AOFIS [Doctor et al. 2005], [Doctor 2006] uses the same learning style as ISL and improve to make the system immune to noise and uncertainty by adding type-2 fuzzy logic functionalities to the agent.

All of the above-mentioned studies compose a fixed structure and require to be rearranged if the environments configuration has to be changed. To do this an administrator needs to shut down the system to add new devices or remove broken ones. Also the studies discussed ignore the true visions of AIE where devices with limited resources would operate in the most efficient and effective way to provide additional service to the user and assist them in their everyday activities. There is a little research conducted to investigate the interconnection aspects of AIE with the aim of classifying them into work groups and reduce the input space of devices to the most relevant and needed ones.

ABI ‘breaks’ the static structure paradigm of the systems and demonstrates an algorithm that dynamically discovers and hierarchically clusters functionally-related sensors and effectors [Trindler 2003] within an AIE. Based on the events generated by these devices, the system builds up a weighted directed graph where the weights are event-dependent and adapted with a hebbian-learning rule. One of the key issues which have not been properly addressed is the need for a distributed and autonomous operation of several agents in the environment. In [Trindler 2003] the agent, which overlooks a whole building, is mainly concerned with clustering the sensors and actuators into functional partitions and is not much concerned about the control aspects of the agents. Furthermore, it requires collecting and storing a huge amount of data over a long period to achieve a good result from the clustering mechanism. For this, the agent requires to be manually associated to every sensor and actuator in the environment, which makes the system computationally expensive and causes a network overload. In contrast to the presented work in this paper, the agents in [Trindler 2003] seek to find a more general representation of the building’s functional-dependent structures whereas the F-IAS agents find the particular user-related structure of associations between the devices which is the among the main purposes of AIEs.

The ANS [McCann et al. 2004] features a middleware for the integration of autonomic systems in ubiquitous computing environments. The ANS tool uses the OWL ontology to find replacement devices e.g. in the case of a failure, but it also facilitates an adaptive agent-like component (node), which can learn the user's preferences. This ability is also used to make a decision over the importance of the replacement devices. The ANS rewards those nodes moves along with the user's preferences. The component based software agents come with predefined sets of abilities that pertain to that agent and its role with the system. The disadvantage of the ANS is that it requires updating its' ontology descriptions every time a new type of a device needs to be integrated into the system, otherwise the service of that device would not be available for the system. Each component is described in terms of what it requires and what it provides at data structure level, although they become adaptive at a later stage. In addition, ANS requires a continuous exchange of a large amount of messages among the agents of the environment [McCann et al. 2004], which are also processed at the agent level.

SMCs use a policy-based mechanism for self-configuration and self-management in autonomic computing environments. When a device is assigned to a role, any policies applicable to that role are loaded onto the device and enabled. The policies are predefined for each device although some sort of autonomy for adapting the roles is provided. The policies represent the aimed objective of the devices and the SMCs 'intelligently' try to discover new nearby components that will contribute to their functionalities and create associations with the most applicable ones. The SMCs perform using a top-down approach which relies on predefined policies to accordingly reconfigure and reorganize itself accordingly.

From the discussion above, it can be summarized that there is extensive research conducted in the area of agent and artifact association for different purposes and at different levels. However, one of the key issues which have not been properly addressed by these approaches is how the devices and thus the system select its associations intelligently through user interaction without a pre-defined association selection paradigm, e.g. overall policies or semantic correlations. This paper focused exactly at this level of the problem and presented intelligent embedded agents that are competent to interact with the user and accordingly find the most relevant associations among a group of agents residing in multiple societies.

7. CONCLUSION

This paper presented a novel intelligent embedded agent technique for reducing the number of associations and interconnections between the various agents operating within an AIE in order to minimize the processing latency and overhead caused by message flooding whilst reducing the cognitive load of programming these associations to personalize themselves to the user needs. For this, we proposed the Intelligent Association System (IAS) which is a framework that integrates large numbers of agents into society based divisions depending on common objectives, proximities and/or other user-related characteristics and intensions. Within this framework, we defined three types of agents, input agents, which are used purely to provide sensory information, intelligent agents (F-IAS agents) which are equipped with monitoring, learning and reasoning capabilities as well as embassador agents which define a special type of F-IAS agents used as the representatives and mediators of the societies they belong to.

The main goals of the proposed fuzzy based F-IAS agents operating within AIEs included learning and adapting to the user behaviors in a lifelong non intrusive mode to control the environment of his behalf. In addition, the F-IAS agent aimed at reducing the agent associations and interconnections to the most relevant set in order to reduce its processing overheads and thus implicitly improving the system overall efficiency.

The embassador agents extended the core functionalities of an F-IAS agent. Besides acting as an F-IAS agent, they also aimed at reducing the number of messages reaching their own society by performing an analysis and filtering routine to determine if the propagated events match the desired criteria of their member agents. This was achieved through the utilization of the proposed intelligent association discovery and selection mechanism which addressed the following questions: (1) *How is it possible to discover potential associations among agents residing in different societies without prior knowledge of their existence* and (2) *How is can these agents be selected and evaluated to determine what their true relevance and importance is?*

A solution for (1) was given by the Publish/Subscribe (Pub/Sub) middleware infrastructure, where agents can subscribe to topics and as a result publish to or receive messages from agents subscribed to the same topics. Since Pub/Sub facilities a push based messaging infrastructure there is no need to have prior knowledge of the existing agents, of their attributes, or IP address. The messages are just *pushed* to generated topics and agents listening to them can automatically receive them and obtain further properties of the publishing agents. Since UPnP provided the main communication infrastructure of the experiments within the iDorm, we presented a solution on how to merge UPnP with

Pub/Sub so that triggered events were forwarded by UPnP to the topic-based message brokers of the Pub/Sub model.

Once agents became more *discoverable* embassador agents tried to seek and evaluate potential candidates for suggesting associations of agents in different societies, which addressed question (2). The evaluation was performed with the intelligent association weight calculation routing as used within individual F-IAS agents. The difference however is mainly based on its application as for the multi society based intelligent association weight calculation the embassadors construct a two dimensional association matrix which includes all the possible combinations of discoverable agent pairs within the system. Then for each agent pair the association weight is calculated so that the agent pairs with high associations weights are selected to be suggested as a candidate for a *new* inter-society association among agents.

We have presented two sets of experiments in this paper. The first experiments, described the use of intelligent association weight calculation at the individual agent's level with the iDorm which form a real test bed for AIE research within the University of Essex. During the experiments a user stayed five consecutive days within the iDorm and interacted with the agents in a natural way. 6 F-IAS agents were simultaneously deployed along with 7 input agents and the results demonstrated that each of them was able to quickly learn the user's behaviour and recognize the required associations required to operate efficiently and economically. During the iDorm experiments the F-IAS Agents managed to reduce their association by up to 71% and their rule base by up to 91% while keeping the overall systems performance at an acceptable level.

The second part of presented experiments was obtained from an emulation of an iDorm-like large-scale AIE environment with multiple societies. The main motivation of this set of experiments was to demonstrate the embassadors unique capabilities to discover and evaluate potential associations among agents nested in different societies. The embassadors were listening to multicast events (in form of advertisement messages) that were randomly initiated by the agents and tried to find co-occurrence among messages. The experiment demonstrated that after a specific period the embassadors could discover agents of different societies and suggest new associations due to their high relevancies.

Since the latter approach is ongoing, we suggest the following future work. (1) To validate the results of the emulation and to investigate the proposed system in a truly distributed and real AIE with a richer set of sensors, actuators, F-IAS and embassador agents based on multiple overlapping societies, e.g. in the form of multiple rooms (like the newly established Essex iSpace) and extract results based on the user's interaction with the iSpace. (2) To explore more the tradeoffs between human intervention and accuracy and efficiency of the F-IAS agents as well providing the agents with the ability to make good judgment on the priorities of their functionalities. By reducing associations the system might lose rules that might be essential for its operation but would it a systems with more associations (and consequently more rules) mean that it is more accurate? The rule base might contain rules that are less important than other ones. A solution might be given by prioritising rules for instance, rules that are significant for a proper operation of an agent and rules that are used rarely but initially extracted. This categorization of rules would allow pointing out the associations that might have produced these less important rules so that they can be dissolved. As with the current agent model, any important rules that were deleted because of an association removal can be relearn by the agents during the adaptation process. (3) To assess the possibilities to improve capabilities of the embassadors in regards to functional and non-functional personalized discoverability of the agent among diverse societies.

ACKNOWLEDGMENTS

We would like to thank all the members of the Intelligent Inhabited Environments Group at the University of Essex for their indirect contributions and support.

REFERENCES

CALLAGHAN, V., CLARKE, G., COLLEY, M., HAGRAS, H., CHIN, J., AND Doctor, F., 2004. Inhabited Intelligent Environments. *BT Technology Journal*, 22(3), 233-247, July 2004.

DAVIDSSON, P., BOMAN, M., 2005. Distributed Monitoring and Control of Office Building by Embedded Agents. Journal of Information Sciences. Vol. 171, No. 4, pp. 293-3007.

DOCTOR, F., HAGRAS H., CALLAGHAN, V., 2005a. A Fuzzy Embedded Agent-based Approach for Realizing Ambient Intelligence in Intelligent Inhabited Environments. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 35(1), 55-65, January 2005.

DOCTOR, F., HAGRAS H., CALLAGHAN, V., 2005b. A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments. *International Journal of Information Science*, 171(4), 209-334, May 2005.

DUMAN, H., BIEN, Z., HAGRAS, H., CALLAGHAN, V., CLARKE, G., AND COLLEY M., 2004. A Multi-Agent Pervasive Computing Architecture For Geographically Dispersed Care Environments, In *Proceedings of the 10th International Fuzzy Systems Association World Congress (IFSA '03)*, Istanbul, Turkey, June 2003.

DUMAN, H., HAGRAS, H., AND CALLAGHAN, V. 2007a. Intelligent Association Selection of Embedded Agents in Intelligen Environments. *Pervasive and Mobile Computing 3(4)*, 117-157.

DUMAN, H., HAGRAS, H., AND CALLAGHAN, V. 2007b. A fuzzy-based Architecture for Learning relevant embedded agents associations in Ambient Intelligent Environments. In *Proceedings of the IEEE Conference on Fuzzy Systems*, London, UK, July 2007.

DULAY, N., HEEPS, S., LUPU, E., MATHUR, R., SHARMA, O., SLOMAN, M., AND SVETEK J., 2005, AMUSE: Autonomic Management of Ubiquitous e-Health Systems, In *Proceedings of the UK e-Science All Hands Meeting*, Nottingham, September 2005.

EUGSTER, P., FELBER, P., GUERRAOUI, R., AND KERMARREC, A. 2003. The many faces of publish/subscribe, *ACM Computing Surveys*, 35(2), 114-131, June 2003.

HAGRAS, H., CALLAGHAN, V., COLLEY, M., CLARKE, G., POUNDS-CORNISH, A., DUMAN, H., 2004. Creating an Ambient Intelligence Environment using Embedded Agents. *IEEE Intelligent Systems Magazine 19(6)*, 12-19.

HAYKIN, S., 1998. *Neural Networks – A Comprehensive Foundation*, Prentice Hall, 2nd Edition, 1998.

HELAL, S., MANN, W., EL-ZABADANI, H., KING, J., KADDOURA, Y, AND JANSEN, E., 2005. The Gator Tech Smart House: A Programmable Pervasive Space, *IEEE Computer Magazine*, pp.50-60, March 2005.

HOLMES, A., DUMAN, H., POUNDS-CORNISH, A., 2002. The iDorm: Gateway to Heterogeneous Networking Environments. In *Proceedings of the International ITEA Workshop on Virtual Home Environments*, Paderborn, Germany, February 2002.

ISSARNY, V., SACCHETTI, D., TARTANOGLU, F., SAILHAN, F., CHIBOUT, R., LEVY, N. AND TALAMONA, A. 2005. Developing Ambient Intelligence Systems A Solution based on Web Services. Automated Software Engineering. 12(1), 101-137, 2005.

JUNG, 2007. Java Universal Network/Graph Framework [Online], http://jung.sourceforge.net/

KOSKO, B., 1992. *Neural Network and Fuzzy Systems*, ISBN 0-13-611435-0, Prentice-Hall, 1992.

MASUOKA, R., LABROU, Y., PARSIA, B. AND SIRIN, E. 2003. Ontology enabled pervasive computing application. *IEEE Intelligent Systems 18(5)*, 68-72.

MILLER, B.A., NIXON, C., TAI, C., WOOD, M.D., 2001. Home Networking with Universal Plug and Play. *IEEE Communications Magazine*. Vol. 29, No. 12, pp. 104-109.

MCCANN, J., KRISTOFFERSON, J. , ALONSO, E., 2004. Building Ambient Intelligence into a Ubiquitous Computing Management System, *In Processings of the International Symposium of Santa Caterina on Challenges in the Internet and Interdisciplinary Research*, Amalfi, Italy, 2004.

MOZER M., 1999. An intelligent environment must be adaptive, *IEEE Intelligent Systems Magazine*, March/April 1999.

THOMAS, P., 2007. Open Source and Java EE development. *IEEE Power Engineering Society General Meeting*, Tampa, June 2007.

TRAJAN, 2006. Trajan Neural Network Software [Online], http://www.trajan-software.demon.co.uk/

TRINDLER, J., AND ZWINKER, R., Adaptive Building Intelligence, Research Work Report, University and ETH Zurich, September 2003.

WILSON, M., 2004. The EU Information Society Technologies Programme Advisory Group – ISTAG. *ERCIM News*, No 59.

WANG, L-X. 2003. The WM method completed: A flexible fuzzy system approach to data mining. *IEEE Transactions on Fuzzy Systems*, 11(6), 768-782, 2003.

YOUNGBLOOD, M., HOLDER, L., COOK, D., Managing Adaptive Versatile Environments, In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PERCOM'05)*, 351-360, 2005.

ZADEH, L.A., 1965. Fuzzy sets. *Information & Control*, 8, 338-353.

ZADEH, L.A., 1996. Fuzzy logic = Computing with words. *IEEE Transactions on Fuzzy Systems,* Vol. 4, 103-111, 1996.

