In Special Issue of Ambient Intelligence of the ACM Transaction on Autonomous and Adaptive Systems, 2008

A 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 compbinat ions 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 present 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 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 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 pre-emptivelycontrol 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 embassador agents which limit the number of messages reaching the societies by performing an analysis and filtering routine to determine if the propagated events mat ch 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

---

Authors' addresses: Department of Computing and Electronic Systems, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom; email: hakan.duman@ieee.org

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) [ISTAG]. 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 pre-emptively controlling the user’s environment on their behalf. They also need to provide adaptive lifelong 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 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.

This paper proposes a novel framework for environments 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 utilize a novel technique for reducing the number of associations and interconnections between the various agents in order to minimize the processing overheads whilst reducing the cognitive load of programming these associations to personalize themselves to the user needs by learning and adapting to the the user's behavior in a lifelong non-intrusive mode to pre-emptively control the environment of his behalf. The 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 in a non intrusive manner while being able to adapt and self configure itself in a lifelong learning mode. The fuzzy-based intelligent agents (F-IAS agents) besides learning the behaviour of the user, performs an online intelligent association evaluation based on modified hebbian-learning 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 behaviour 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 presents a discussion on related work. In section 3 we introduce the intelligent Dormitory (iDorm) which forms the real world AIE experimental test bed for proposed intelligent agent techniques. This section also covers a discussion on the limitations of the iDorm's UPnP infrastructure and proposes the hybrid of the UPnP and Pub/Sub infrastructures to achieve scalability and resilience for a multi society based AIE environment. Section 4 introduces the Intelligent Association System (IAS) framework and architecture, presents its notions and definitions and explains how the UPnP-Pub/Sub infrastructure can be utilized for IAS. In section 5 we introduce the intelligent fuzzy based embedded agent that is capable of lifelong learning and adaptation to the user behavior in an online and non intrusive fashion. Section 6 explains the proposed intelligent association weight calculation mechanism used by the intelligent agents to select an optimum set of the more important associations to maintain (dropping others) so that the agent processing latencies and overhead can be reduced and the performance increased. This section also describes an adjusted version of the 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 7 discusses the experiments and results while section 8 finally presents the conclusion.

- - - - - -

2. RELATED WORK

2.1 Intelligent Environments

The following projects demonstrate the use of intelligent agents within intelligent environments. Each of these project share its own way of deploying and analyzing. The number of intelligent agent-based environments increases instantly and the notation of 'intelligence' in agents vary depending on the application targeted. Thus, this section will only list the projects that are most notable and relevant to the paper.

The Adaptive House [Mozer 1999] uses a soft computing approach based on neural networks, which was focused mainly on the intelligent control of lighting within a building. The aim is to enable an environment that does not require any kind of user interface or explicit control such as a touch-screen or speech recognition. Instead, the user should interact with the home exactly as they would interact with any ordinary home. The adaptive house monitors actions to anticipate and take over many manual tasks as it becomes better trained [Mozer 1999]. The Adaptive Home, a.k.a. the Neural Network Home, can be regarded as the pioneer project which explored the 'learning user's habits' aspects of an AIE. It was aiming to predict occupancy of rooms, hot water usage and the likelihood that a zone is entered in the next few seconds, using trained feed-forward neural networks running on a centralised computer architecture. The context information in the project was again mainly comprised of location, but additional state information from rooms like the status of lights or the temperature set by inhabitants was used. Although learning and prediction was done via feed-forward multi-layer perceptrons with the known limitations, it showed that prediction of user locations can help to save resources and support users by learning their behavior and automating simple tasks [Mozer 1999].

The Gator Tech Smart House aims to create a 'smart house in the box', i.e. off-the-shelf assistive technology for the home that an average user (especially disabled and elderly) can buy, install, and monitor [Helal et al., 2005]. The care agent included into an intelligent environment use the following: cameras to monitor the living space aided with motion sensors; automated blinds that work with the air conditioning to help control temperature; a 'smart floor' that can detect motion and if someone falls; and even a sensor in the mailbox that alerts the resident when mail has arrived. The operating system which carry out functionalities enables its occupant to program the devices in an independent or collective level. The Plug-and-Play feature allows new devices to be added to the environment.

The MavHome smart home project focuses on the creation of an environment that acts as an intelligent agent, perceiving the state of the home through sensors and acting upon the environment through device controllers [Youngblood et al., 2005]. The environment is represented using a Hierarchical Hidden Markov Model and a reinforcement learning algorithm is employed to predict the environmental preferences based on sensors within the environment. Desired actions are proposed for the control of lights within the environment primarily based on motion detection sensors and if the actions are within the bounds of acceptable safety and security policies, they are invoked within the environment. The agent aims to maximize the comfort and productivity of its inhabitants while minimizing the operation cost.

Including reasoning, planning and learning in devices and artifacts and hence embedding intelligence was suggested by researchers at the University of Essex [Callaghan et al., 2004]. These concepts were explored in various projects such as the intelligent Dormitory (iDorm) which at the same time forms a test bed for the IIE research. Many approaches have been presented and analyzed for online learning and adaptation of the agents embodied in the iDorm such as the ISL [Hagras et al., 2004] and AOFIS [Doctor et al., 2005]. Both ISL and AOFIS employed fuzzy logic based embedded agents that seek to particularize (rather than generalize) to the specific user needs and respond immediately to whatever the end user demands (providing it does not violate any safety constraints). Moreover, reports of ongoing projects consist of the development and experimentation with new types of autonomous embedded agents such as a new type-2 fuzzy logic based embedded agents that are more robust to uncertainties and are capable of learning and adapting in a non intrusive way using a lifelong approach [Doctor et al., 2005].

The aforementioned intelligent agent based approaches focus more on the lower-level intelligent adaptation and learning features of the individual embedded agents for AIEs. The emphasis is on the use of a single 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.

2.2 Association Management Systems

Different approaches have been proposed to address the relevancy evaluation of devices for a given problem and domain. The most similar ones to our approaches are listed next.

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 et al., 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 et al., 2003] the agent, which oversees 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 large amount of data over a long period to achieve a good result from the clustering mechanism. For this, the agent needs to be manually associated to every sensor and actuator in the environment, which makes the system computationally expensive and causes a network overload. Another drawback is that the algorithm seeks to find a general representation of the building's functional-dependent structures whereas the embedded agents' approaches in this thesis find the particular user-related structure of associations between the devices.

A significant body of work is also emerging within the multi-agent system community which studies the integration of agent technologies to intelligently learn and exploit relevancy between associations [McCann et al., 2004] [Dulay et al., 2005]. The learning employed in [McCann et al., 2004] was accomplished by task-specific predefined policies which enable the agents to specifically associate with available devices in the search space and allow some degree of adaptation e.g. to have policies for new devices joining the domain or a existing ones leaving, which is commonly the case for mobile devices. However, it cannot always be assumed that the devices and services present in an AIE are semantically described or predefined with operational policies. Hence, to allow the full realization of AIEs, the devices need to learn their associations and adapt their policies or rules during operation to changes and failures occurring within the environment.

In addition to the above mentioned application-level approaches, there is an increasing popularity among the networking and middleware community which aim to improve the service discovery and matchmaking. Therefore, a wide variety of different lookup architectures have been proposed and implemented by research and industry. The most prominent industrial standards are Jini [JINI], UPnP [UPNP], and Salutation [SALUTATION]. Jini was designed to support nomadic, Java-enabled environments where mobile devices join an existing network and use services in an ad hoc manner. A Jini lookup service is discovered using multicast. It may offer Java-based service interfaces that are downloaded to a mobile device in order to interact with a given service. Jini lookup services can form a hierarchy and requests may be passed up this hierarchy for resolution. Universal Plug and Play (UPnP) is a framework defined at a much lower level than Jini. It offers IP address allocation and DNS name assignment for mobile devices and builds, for example, on DHCP. UPnP's Simple Service Discovery Protocol (SSDP) supports registration and discovery of devices. This may involve dedicated directory services but does not rely on them. In Salutation, devices use a Salutation Manager (SLM) for the lookup process. SLMs may exchange registration information and support clients by mediating data transport that covers different transport protocols. A client queries a near-by SLM in a similar way as is done in the ASG Lookup Service. However none of these standards support discovery and selection of services (provided by devices) based on a user-driven relevancy analysis. Most of them conduct a service and matchmaking on functional and/or non functional properties, where the selection is mostly driven by semantically similar attribute matching (e.g. same type of devices or they are located at the same proximity). However in an AIE, services have to be discovered based on the user's needs while trying to satisfy the system's overall constraints and performance.

From the above discussion and to the authors' knowledge, no work in AIEs has investigated creating a method to reduce the associations and interconnections between the various devices, services and agents to a minimal number required to satisfy the user's personalized needs.

The next section describes the intelligent Dormitory (iDorm), which is a unique test bed for AIEs research.

3. THE IDORM – A LIVING EXAMPLE OF AN AIE

3.1 The iDorm and iSpace

The intelligent Dormitory (*iDorm*), as shown in Figure 1, forms one of the main test beds for AIEs research at the University of Essex. 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 has a standard multi-media PC that combines a flat screen monitor and a multi-media video projector which can be used for both working and entertainment.

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 embedded 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:

- Agents embedded within the sensing devices within the iDorm; they are termed *passive agents* as they deliver purely sensory information. The passive agents are embedded in the following sensing devices: 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).
- Agents embedded within the actuating devices within the iDorm; they are termed *intelligent agents* as they are autonomous entities enhanced with intelligent reasoning and decision making. The intelligent agents are embedded in the following actuating devices: 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 intelligent agents are also embedded in other devices such as *mobile robots* that can navigate within the user environment to serve the user needs such as getting drinks and medicine [Callaghan et al., 2004].

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]. UPnP provides an architecture for pervasive peer-to-peer network connectivity of intelligent embedded agents, wireless mobile devices and artifacts and PCs of all form factors. It is designed to bring easy to use, flexible, standards based connectivity to ad-hoc or unmanaged networks whether in the home, car, public spaces, or attached to the Internet. Moreover, UPnP is a distributed, open networking architecture that leverages

TCP/IP and the Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office and public spaces.

The UPnP technology which is based on an event based communication middleware allows agents to plug and play in an ad hoc fashion thus enabling *automatic discovery* and *configuration* [UPNP]. Within the iDorm UPnP distinguishes between two types of embedded agents: (1) UPnP Devices *(Sensing devices with embedded passive agents)* and (2) UPnP Control Points (*Actuating devices with embedded intelligent agents)*. The *UPnP services* define the functionality offered by the device where the control points use the services to control the device and monitor their status [Duman et al., 2007a]. However, the control points must be connected to devices first to make use of these services. The service subscription procedure of UPnP is responsible for establishing and disestablishing associations between the embedded agents. Once linked, the events coupled to the associated agent's services and control messages are exchanged on a unique channel assigned by UPnP. These events can be in the form of state changes, new arrival or removal notification of devices. Thus, the UPnP infrastructure is well suited for highly dynamic distributed systems like AIEs.

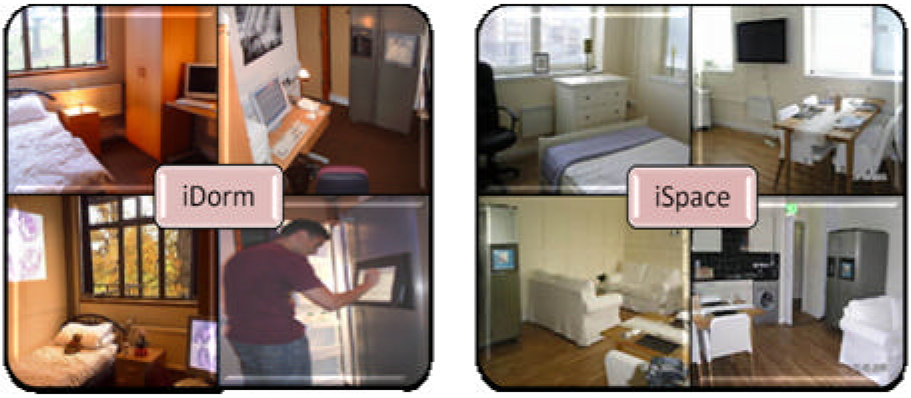


Fig. 1. The University of Essex intelligent Dormitory (iDorm) and intelligent Space (iSpace)

Another unique test bed, also located at the University of Essex, is the Essex intelligent space (iSpace) which is also shown in figure 1. The iSpace is a spacious two bedroom flat with a kitchen and a bathroom. The iSpace provides a flexible test-bed for research into AIEs and it offers the possibility for examining the deployment of embedded agents and sophisticated user interfaces within the intelligent environments of the future. Although the iSpace looks like a domestic flat at first glance, it actually extends the ideas and terminologies of the iDorm into a *multi-room* and *multi-user intelligent environment*. There are numerous networks in place ranging from wired and power-line, thorough wireless to broadband and high bandwidth multi-mode fibre connections to the outside world [Callaghan et al., 2004]. As in the iDorm, the low-level UPnP control architecture enables communication between the services that the agents provide which can be controlled wherever desired.

The occupants may freely decorate the iDorm and iSpace with embedded agents making them to become dynamic and open environments where numerous new agents may enter or existing leave in an ad hoc fashion. Generally, there is no limit on the amount of agents the user can bring in and no restrictions (except safety regulations) on how he/she uses them.

3.2 The Limitations of the UPnP Infrastructure

UPnP which forms the middleware of both, the iDorm and iSpace, uses the Simple Service Discovery Protocol [UPNP] in order to discover network devices and their services. The protocol is rather simple and has a number of limitations. 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 devices (embedded agents) have a somewhat local view of the network, meaning that they only can advertise their presence and discover other devices of the same subnet. Hence, devices attached to other (sub-) networks are invisible to these devices 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 devices 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.

The next section introduces the contrary of this client/server-style messaging paradigm, namely the "*publish-subscribe*" which provides a better infrastructure for agents of different societies.

3.3 Publish/Subscribe (Pub/Sub)

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. The most popular models are listed as follows.

3.3.1 *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. In practice, topic-based Pub/Sub subscriptions use string matching for event delivery where every event is part of a hierarchy of topics. 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. The closer to the root, the more general the selection criteria will get. E.g. an agent subscribed to "/iDorm/Areas/*" (note the wildcard '*') would receive all of the events published in every subsequent topic of the hierarchy. An example of a topic-based hierarchical structuring of the Department of Computer Science is presented in figure 2. Despite facilitating wildcards this Pub/Sub model remains and represents a static scheme, which offers limited expressiveness [Eugster et al., 2003]. In addition, the topic hierarchy (incl. topic names) needs to be utilized in the same way by all of the publishers and subscribers (i.e. embedded agents) which require prior knowledge before operation. However, 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. This feature of the topic-based Pub/Sub middleware is especially essential for the proposed multi-society based associative discovery and selection and is not or not fully provided by UPnP as described in the previous section. The reasons to use a topic-based Pub/Sub infrastructure for AIE can be summarized with the following:

- Agents do not have to keep a directory of all other agents' availability, location and other functional or non-functional attributes. The agents only subscribe to the topics 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.

3.3.2 *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]. Boolean expressions in combination with comparison operators (=, <, <= , >, >=) analyse the content of the event and forward them accordingly to the subscribers. 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.

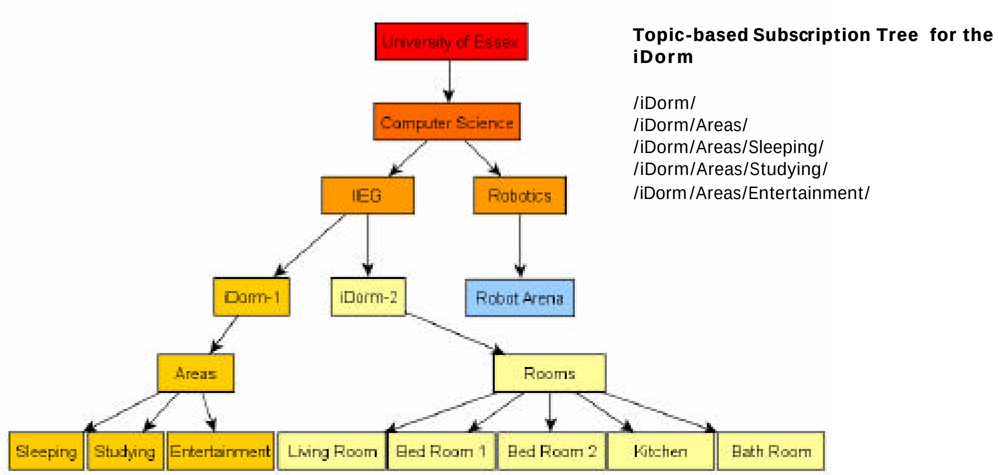


Fig. 2. Topic-based Pub/Sub subscription structure.

3.4 Merging UPnP with Pub/Sub

The UPnP is a straightforward communication protocol that is reliable in general but fails to accommodate structure dynamism as described in the previous sections. However it defines a good event monitoring and triggering facility which is mostly efficient and reliable. For instance, whenever a change in the UPnP devices occurs, the UPnP protocol triggers an event and multicasts it to the agents previously subscribed to them. As mentioned previously subscribing to a service of an UPnP agent requires a prior knowledge about their existence and properties. Also, these subscriptions are handled in a *point-to-point* fashion which reduces the overall scalability of the system. In order to overcome both technologies limitations, the following describes briefly how the UPnP protocol can be extended to allow topic- and content-based Pub/Sub style messaging so that the necessity of prior knowledge of agents abolishes. The only knowledge they have to be equipped with is the society they belong to in the form of a topic name. After subscribing to this topic, the agents will start receiving events dispatched to the very same topic so that they can make use of this information e.g. to learn the associations to existing more relevant agents and "unlearn" the interconnections to agents that are producing redundant information for them. It should be noted, that Pub/Sub can built up on any other service discovery protocol however due to the reason that the iDorm is using UPnP as the middleware to combine its various different networks, we focus on integrating Pub/Sub into the UPnP infrastructure.

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 3 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) [JAVA] and can be put on top of many industry messaging and Pub/Sub products, including Corba, Jini etc.

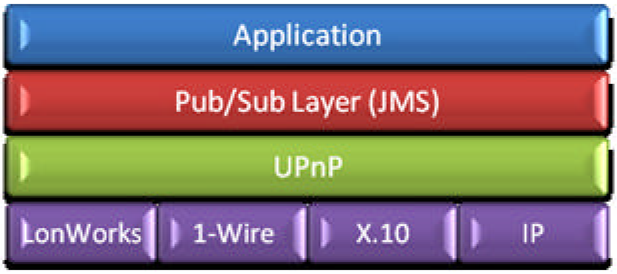


Fig. 3. The extended communication infrastructure based on combination of UPnP and JMS.

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

- Each agent is interfaced to the iDorm's UPnP middleware as a UPnP Device
- Each 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. If the topic name doesn't co-exist within the message brokers of other agents, this topic can be regarded as a new topic; in contrary if the user for instance has knowledge about existing topics, he/she can assign it to the agent, so that events dispatched with this topic are forwarded to the newly joined agent.

- 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.

3.5 The dilemma of advertisements in Pub/Sub

As with most technologies, Pub/Sub also comes with some limitations and problems, where the message flooding is perhaps the most well known within the Pub/Sub community. In both Pub/Sub models (topic-based and content-based), agents must broadcast advertisements periodically to notify about their existence and availability so that this information is then stored in soft state of the brokers. Since the advertisements are "flooded", it can be argued that scalability does become an issue. Also the messages would require "forcing" the agents to process them one by one to evaluate the significance of the information for their operation. This cannot be accepted as it would lead to drastic increase of the agents processing. Especially in an AIE where agents have to interact with the user in real-time and have limited computational resources and memory storage available, they are expected to run in the most efficient and reliable fashion aiming to reduce the delays caused by flooded "redundant" messages. This paper therefore addresses this necessity and describes a method specific to our approach to significantly reduce the messaging and agent processing overhead. We deploy a novel hybrid Pub/Sub infrastructure where agents are clustered into societies (represented by topic labels). With this, messages are broadcasted only to the desired topics avoiding flooding of other societies of the environment. In addition, every intelligent agent of a society performs intelligent tasks to increase their efficiency by eliminating redundant associations and finding more relevant associations to other agent of the same or different societies. Nevertheless, to make suitable associations more discoverable, especially in multi-society environment there is an immediate need for an intelligent mechanism that can detect the relevancy of events published within different societies and notify agents about their coexistence.

The next section first introduces the Intelligent Association System (IAS) which forms the framework for multi agent and society based AIE.

4. THE INTELLIGENT ASSOCIATION SYSTEM (IAS)

4.1 The IAS Framework

This section introduces the architectural framework of the Intelligent Association System Framework (IAS), which resides on top of a physical network and combines all devices and their services to a decentralized *service-oriented overlay network* architecture [Duman et al., 2007b]. Service-oriented architectures have been proposed for ubiquitous computing and 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.

Within an AIE different types of agents may exist and can be explained by looking at their individual purpose and supplied services. A granulated categorization of an agent that is used within an AIE is defined as follows [Duman et al., 2007a]: (1) *Passive* 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.

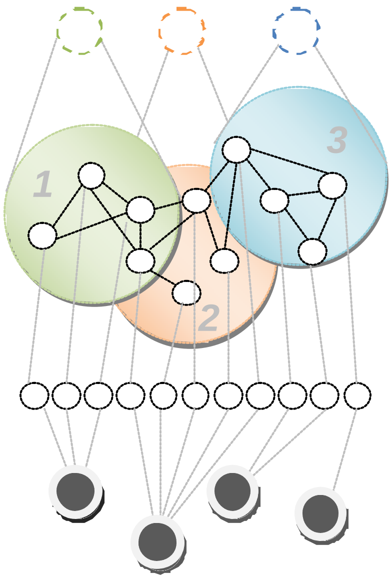
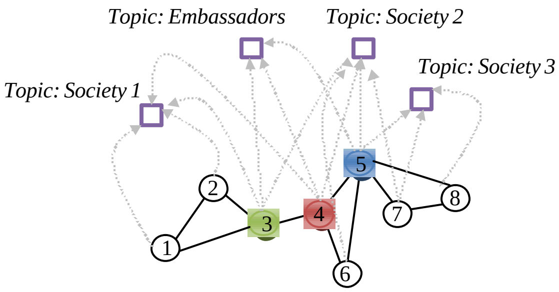


Fig. 4 The Intelligent Association System (IAS) Framework.

The embedded agents (here after agents) within a society are interconnected via links referred to as *associations*. 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. It should be noted that an embassador agent isn't a separate unit and 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 *Intelligent Association System (IAS)* architecture integrates the above components into three layers as shown in figure 4.

4.2 Pub/Sub for IAS

The IAS framework utilizes a society-based segmentation of the environments (e.g. iDorm) and accommodates agents that aim to optimize themselves (i.e. to reduce the agent communication and processing latencies) by exploring the best possible associations to other agents. In addition, the agents should remain resilient to agent failures and be able to discover other replacement agent in order to provide a robust and persistent (fault-tolerant) operation. This, however, mainly relies on the *discoverability* of the other agents of an AIE. Most traditional discovery protocols fail to provide this flexibility as soon as the societies become independent units. The agents of these societies become invisible for other agents so that associations automatically dissolves and previously associated agents fail to receive important information which is required for proper operation. Thus, a good solution is provided by the Pub/Sub messaging paradigm, where societies can be labeled as topic names, the *embassador agents* taking up the role of a messaging broker (or *Mediators*) among different societies and the events filtered based on their context and other desired criteria specified by the user or the system.



Type	ID	Topic
Intelligent Agent	1	Society 1
Passive Agent	2	Society 1
Embassador	3	Embassadors, Society 1, Society 2
Embassador	4	Embassadors, Society 1, Society 2
Embassador	5	Embassadors, Society 2, Society 3
Passive Agent	6	Society 2
Intelligent Agent	7	Society 2, Society 3
Passive Agent	8	Society 3

Fig. 5. The Intelligent Association System (IAS) Framework.

Figure 5 illustrates an example of this topic labeling. For example, *Agent 1* which is an intelligent agent is subscribed to topic *Society 1* as *Agent 2*, *Agent 3* and *Agent 4*. All of these agents publish events to the same topic which means that they form an individual society with the name "*Society 1*".

Depending on the type of the agent, they can either send and/or receive events. In the context of Pub/Sub, a passive agent is only a provider as the service it provides is to deliver purely sensory information. An intelligent agent and an embassador agent are providers but can also be subscribers, which means that they can publish as well as receive information from specified topics.

4.3 Embassadors as Pub/Sub Mediators

Agents with more than one *unique* topic subscription (such as *Agent 3*) are considered as multiple society members. This leads to the creation of overlapping societies as illustrated in figure 6. 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.

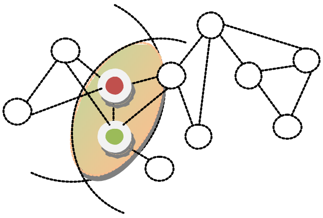


Fig. 6. Overlapping societies as a result of multiple topic subscriptions

The embassadors are mostly agents with higher computational power, memory storage and networking capabilities and as such become a good candidate to take up the role of mediators of the multi society based Pub/Sub middleware infrastructure. In other words, the embassadors (besides operating as intelligent agents that seek to optimize the number of their associations) are also expected to be the mediators between multiple societies in order to avoid local message flooding of the societies. The 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. The embassador agents also acting as intelligent agents are concerned with optimizing their associations in that to dissolve connections to unrelated agents and find more appropriate ones for them to operate more effectively and efficiently. The following sections introduce the proposed fuzzy-based intelligent agent also employed as an embassador which utilized the aforementioned functionalities for *association discovery*, *learning* and *selection*.

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

The F-IAS agents are based on the Fuzzy Logic Controller (FLC) which has been credited with providing appropriate framework for generating human readable models for complex systems [Hagras et al., 2004]. 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 and its corresponding rules in a non-intrusive manner while being able to adapt and self configure itself in a lifelong learning mode. A more detailed description of the F-IAS agent's main components is described next.

5.1 The Fuzzy Controller

The FLC is composed from the following processes: fuzzification, rule base, fuzzy inference engine and defuzzification. The fuzzification interface measures the input variables and maps crisp numbers into suitable fuzzy sets. The fuzzy rule base comprises the knowledge of the domain. The fuzzy inference engine is the kernel of an FLC where it has the capability of simulating human decision making based on fuzzy concepts and of inferring fuzzy control actions employing fuzzy implication and the rules of inference in fuzzy logic. In other words, the fuzzy inference engine handles the way in which rules are combined and conducts the fuzzy reasoning process. The resulting fuzzy set from the fuzzy inference engine is then converted into a crisp output value using the defuzzification process. More information about fuzzy logic systems can be found in [Zadeh 1965], [Kosko 1992], [Zadeh 1996].

The F-IAS agent *perceives* the environment through the information provided by the associated agents and their services 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 use of fuzzy logic as a controller is of particular importance as it provides robustness to uncertainties, noise and imprecision attributed to real world systems such as AIEs.

It is assumed that each F-IAS embedded agent has a *N:1* relationship meaning that *N* possible passive agents can be associated to *1* output. It should be noted that the approaches can be easily extended to a *multiple inputs associated to multiple outputs* relationships but for the sake of simplicity, we will consider that the F-IAS agent has only to control one output actuator. In addition, it should be mentioned that a F-IAS agent output can also be used as an input association for another F-IAS agent, however throughout this paper we will consider that the inputs for F-IAS agents will be mainly the passive agents that are embedded within the sensing devices (see section 3).

For an AIE, after collecting the data set of K input-output data pairs. Each vector datum $(\vec{x}^k, y^k)$ can be expressed as $(x_1^k, x_2^k, \ldots, x_N^k; y^k)$, with $\vec{x}^k \in \mathbb{R}^N, y^k \in \mathbb{R}$, $k = 1,2,\ldots,K$. 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:

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

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-min* 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.

The lifelong learning and adaptation capabilities of the F-IAS Agents requires the agents to have an effective, fast and reliable learning method that can generate new rules as well as adapting, changing and removing the existing rules that are stored in the form of Equation 1 in the rule base. The rule induction method of the F-IAS which operates in an online and lifelong learning mode is described next.

5.2 The F-IAS rule induction method

The rule induction method adopted by the F-IAS Agents is adopted from the enhanced version of the Wang-Mendel (WM) method using a one-pass technique to extract fuzzy rules from a sampled data set [Wang 2003]. The algorithm enables every F-IAS agent within the AIE to learn the model and its behavior through interacting with the user. It is a simple, reliable and fast data mining approach to extract fuzzy rules based on a collection of events. The procedure involves the following steps to obtain the model of the system which at the same time characterizes the user's behavior [Wang 2003]:

I. Subscribe to *topic* T or multiple topics $\vec{T}$. The selection of T can be user-driven or automatically suggested.

II. Establish associations to N agents which are member of the subscribed societies $\vec{T}$. Here N can vary depending on the capability and resources and F-IAS agent can provide. F-IAS agents with very high processing facilities and storage may accept more associations than agents with limited resources. In this paper, it is assumed that n is provided by the hardware manufacturer of the F-IAS agents.

III. 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.

IV. Once enough data (in form of events) have been collected assign for each input supplied by a passive agent a set of fuzzy membership functions. A double-clustering approach combining fuzzy-C-means and hierarchical clustering is applied for obtaining these fuzzy membership functions, more information about the fuzzy membership functions generation can be found in [Doctor et al. 2005].

V. Expert rules are allowed and may be combined with the rules induced from the events. This can be freely added by the user or supplied by manufacturers in form of accompanied operation policies (e.g. safety rules which may not be altered).

VI. Start reading events from the cache. For each data pair (x^k, y^k), compute the membership values $?_{A_j^q}(x_j^k)$ for each fuzzy set $q ? 1,\ldots,V$, and input $j ? 1,\ldots,N$, find $q ? ?1,\ldots,V?$, such that $?_{A_j^q}(x_j^k)$ is maximum. The following is the rule generated by (x^k, y^k).

$$\text{IF } x_1^k \; is \; A_1^q \text{ AND } \ldots \text{ AND } x_N^k \; is \; A_N^q \text{ THEN } y \; is \; y^k \qquad (3)$$

VII. Repeat *Step V* for all k from $1,\ldots,K$ 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

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

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

$$av^l \; ? \; \frac{\overset{K_l}{\underset{k?1}{?}} y_k^l w_k^l}{\overset{K_l}{\underset{k?1}{?}} w_k^l} \qquad (5)$$

$$?^l \; ? \; \frac{\overset{K_l}{\underset{k?1}{?}} \left| y_k^l \; ? \; av^l \right| w_k^l}{\overset{K_l}{\underset{k?1}{?}} w_k^l} \qquad (6)$$

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

$$w_k^l \; ? \; \overset{N}{\underset{j?1}{?}} \; ?_{A_j^q}(x_j^k) \qquad (7)$$

VIII. 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. 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. The next section introduces the novel non intrusive approach for intelligent association discovery and selection which takes places at the individual agent's and embassadors (multiple societies) levels.

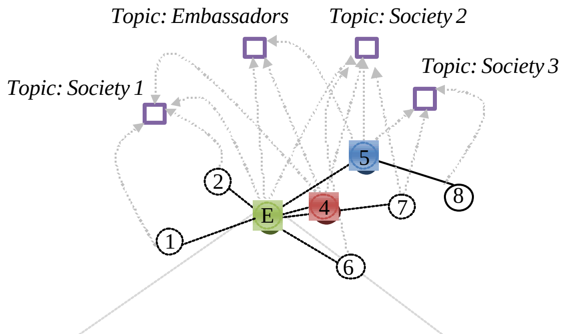
6. THE INTELLIGENT ASSOCIATION DISCOVERY AND SELECTION

Many methods exist for learning and calculating the causal weights of associations between input-output pairs e.g. in NN [Haykin 1998] and GA [TRAJAN], however most of them are computation intensive and mostly rely on huge data sets. Furthermore, they require a long learning period and thus are not suited for online real time intelligent agents. The section describes a method similar to hebbian-learning [Haykin 1998] which is widely used in Fuzzy Cognitive Maps (FCM) [Kosko 1992] to calculate the associative strengths between the concepts. This method is *event-driven* and can be applied in an online fashion to compute the association strength between the agents and the embassadors. The association weight 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. The next subsection presents the proposed intelligent association weight calculation for learning association at the individual agent's level.

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

As mentioned before, the F-IAS agents use the structural notions and descriptions of Fuzzy Cognitive Maps (FCMs) [Kosko 1992]. FCM allow causal evaluation among the associated agents member of the same topic T or multiple topics $\vec{T}$. Fuzzy cognitive maps (FCM) are an extension of the cognitive map which is a collection of nodes with some causal links or edges. The nodes are concepts and the edges represent the links between the nodes which can be illustrated as directed arrow to depict the direction of the influence. The main reason that FCM have been chosen as the structural concept for the work presented in this paper is that it complies well with the notion and requirements of the IAS framework. The FCMs in general represent societies (or topics $\vec{T}$) for the intelligent agents and provide a good framework for adaptation and dynamism of the system. New agents can join and existing ones can be removed in an ad hoc fashion in real-time without the need to suspend the operation of the system. These characteristics show that FCMs 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.

Figure 7 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. 7. The structural concept of fuzzy cognitive maps for F-IAS agents in a Pub/Sub infrastructure

The algorithm of the association weight calculation of an F-IAS agent is described next. It should be noted that the intelligent association weight calculation is computed continuously and updated in the occurrence of an event. The association weight increases when the state of the interconnected agents simultaneously changes. In contrast, the weights decrease over time when the state change of an agent cannot be correlated. 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 passive agents ($X_j ? \quad Y$). 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 $?_{X_j ? Y}$ of $FCM(Y)$ to zero, for all associated agents j=1…N so that the association matrix of F-IAS becomes $?_{X_j ? Y} ? [X_1 ? 0, X_2 ? 0, X_3 ? 0 ... X_N ? 0]$.
3. Set the learning rate $?$ = 0.1.
4. Initialize the pre-associative ($?_{X_j}$ = 0) and post-associative ($?_Y$ = 0) flags for the agent pair $X_j ? \quad Y$.
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 $?_{X_j}$ 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 $?_Y$ to 1 only if the F-IAS has adjusted its output state due to event x_j^k
   c. Calculate the new association weights $?_{X_j ? Y}$ for each agent pair $X_j ? \quad Y$ at time k by applying the following equation:

$$(?_{X_j ? Y})^k ? (?_{X_j ? Y})^{k?1}(1 ? ?)? ?_{X_j} ?_Y \tag{8}$$

   where $(?_{X_j ? Y})^{k?1}$ is association weight and $?$ 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 $?_{X_j}$ and $?_Y$ 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 passive 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 $Count(x_j^k)$ for each agent X_j.
- *Step II*: For each X_j, apply the following equation to obtain the normalization constant value

$$?_{X_j} ? \left(1 ? \left(\frac{Count(x_j^k)}{K}\right)\right) \tag{9}$$

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

$$f_{sig}(?_{X_j ? Y}) ? \frac{1}{1 ? e^{??_{X_j ? Y} ??_{X_j}}} \tag{10}$$

- ✍ *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 $?_{X_j ? Y}$ into the association matrix ? of the F-IAS Agent.

If $f(?_{X_j ? Y}) ? ?$ (where ? 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 ? than this agent may be considered as irrelevant or redundant and would become a candidate to be removed.

During the life-time 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.

6.2 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. In other words, the embassadors filtering and forwarding capabilities as well as Embassador agents glue together agents from different societies and make them discoverable and selectable without a prior knowledge of their existence. Without the embassadors described capabilities (see section 4.3) societies can be regarded as isolated regions of the environment where agents have a limited view of the happenings as shown in figure 8. 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.

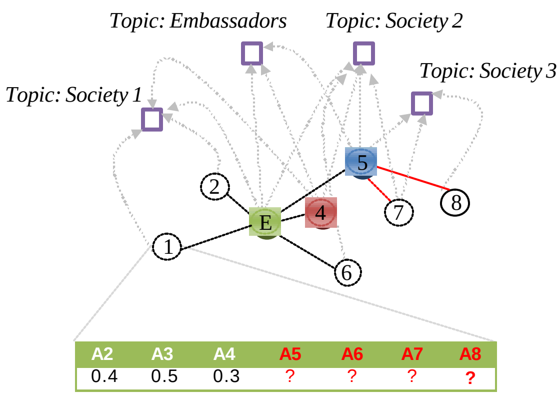


Fig. 8. Agent 1's limited view of the environment

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 6.1. The main differences are incorporated in the following steps:

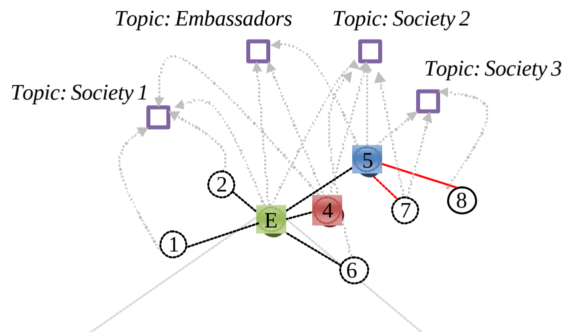
1. Initiate a *two*-dimensional $FCM(E)$ for the embassador E and subscribe to the topics $\vec{T}$ (e.g. *Society 1* and *Society 2*)
2. For every agent X_j of $\vec{T}$ add a column and a row in $FCM(E)$. Figure 9 shows an example of the $FCM(E)$ with Agent 2, Agent 4, Agent 5, Agent 6, and Agent 7 as members.
3. Set the association weights $?_{X_j ? X_m}$ of $FCM(E)$ to zero, for all associated agents $j, m = 1 \ldots N$
4. Run *step 3 to 5 continuously* of the intelligent association weight calculation as described in *section 6.1* to calculate the association weight $?_{X_j ? X_m}$ for all agent pairs $X_j \; ? \; X_m$.
5. At time t apply the $?$ (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 $?$ (as which is the case for $X_2 \; ? \; X_6$, $X_4 \; ? \; X_6$ and $X_4 \; ? \; X_7$) than these association are considered as important. In contrast, if

the weights after a certain period remains less than ? 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 ? X_m above the giving threshold ? , generate a new topic $T_{X_j ? X_m}$ (e.g. $T_{X_2 ? X_6}$ labeled as "Society X_2 ? 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 ? X_m below the threshold start a process to remove them from the $FCM(E)$ and also as them to unsubscribe from the $T_{X_j ? X_m}$.

The above mention approach operates in real time and lifelong 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 ? X_7 many embassadors needed to cooperate so that the communication route of the events can be identified as X_4 ? $E(X_3)$? $E(X_5)$? 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. 9. The association matrix of embassador *E*

7. EXPERIMENTS

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 passive 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 6.1.

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 (see section 3) 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.

7.1 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.

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

- *Passive 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 passive 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.

6.1.2 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 5.2. 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 passive 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 $?_{X_j ? Y}$ 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 ? (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 ? (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 10). 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.

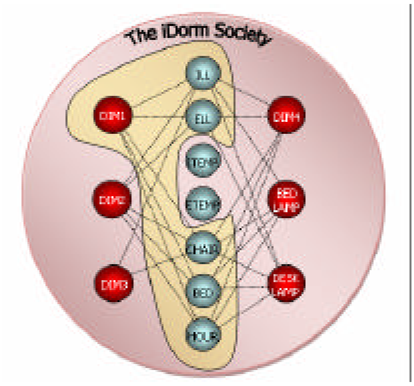


Fig. 10. The DESKLAMP F-IAS Agents Society within the iDorm

7.2 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 passive 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.

7.2.1 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 11 using the JUNG visualization tool [JUNG].The regions indicate randomly selected overlapping societies and each dot represents an agent, which can be either passive or intelligent. The societies are “glued” together via the embassador agents who are selected from the available intelligent agents of the societies.

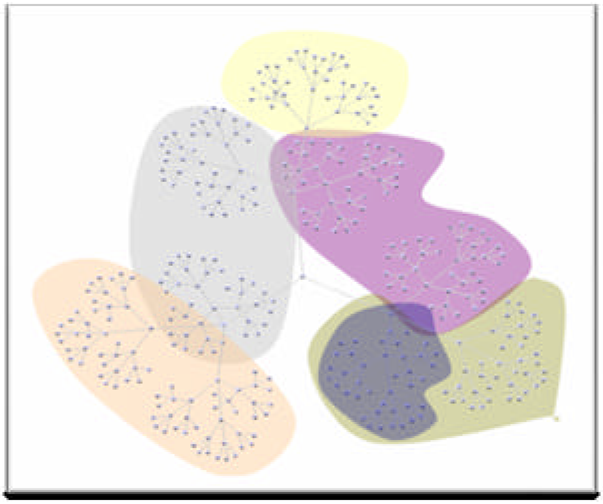


Fig. 11. A multi society based AIE emulation.

Figure 12 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.

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 passive agent of its society. The following steps have been initiated to obtain the described experimental setup:

- For every randomly selected *passive 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 passive 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	Passive	Intelligent	Passive	Passive	Passive	Passive	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.

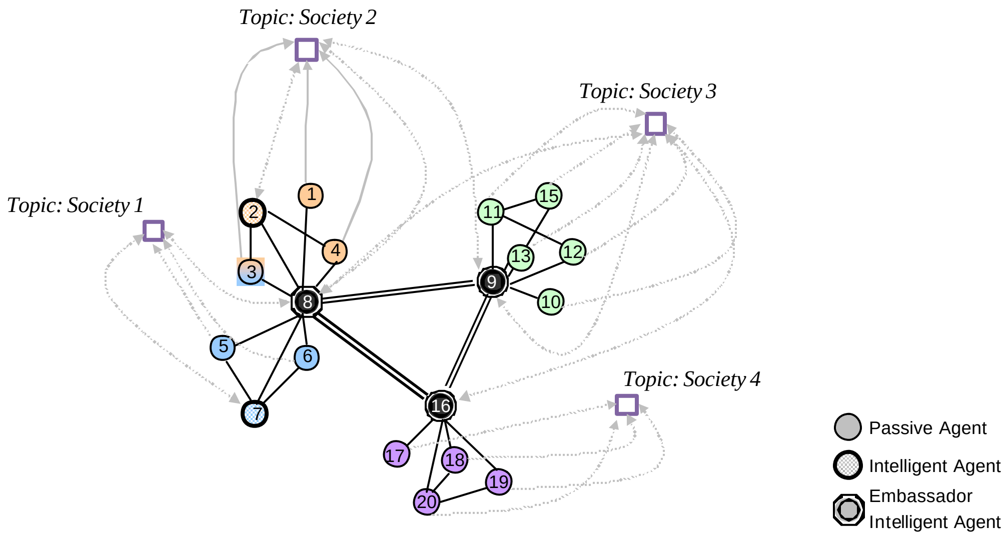


Fig. 12. The simplified representation of the emulated AIE.

6.2.2 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 (passive 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 13, 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.

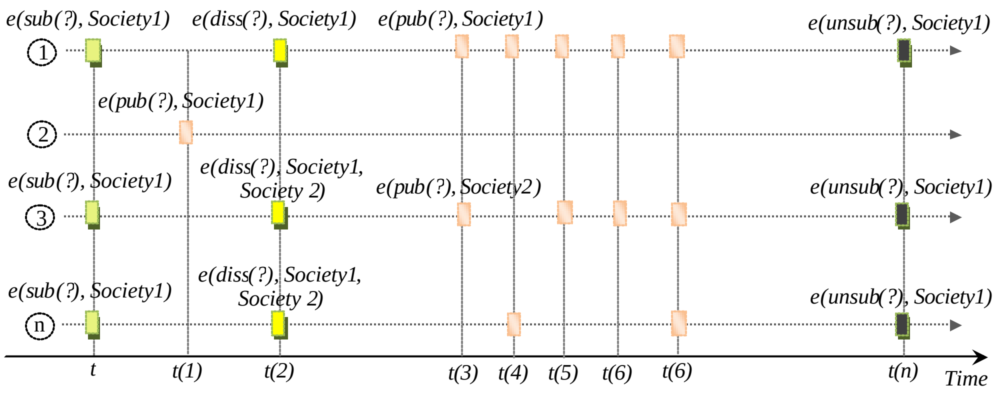


Fig. 13. Events published simultaneously by various agents

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 6.2, 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 14 one can see that the embassador (denoted as *Agent 8* in figure 12) 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 ? X_6. On the other hand, agent pair X_7 ? X_{12} (see figure 15) 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 ? (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.

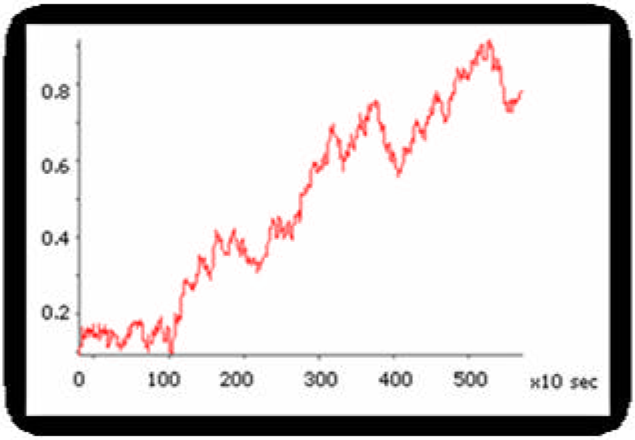


Fig. 14. Relevancy calculation between agent pair X_2 ? X_6 .

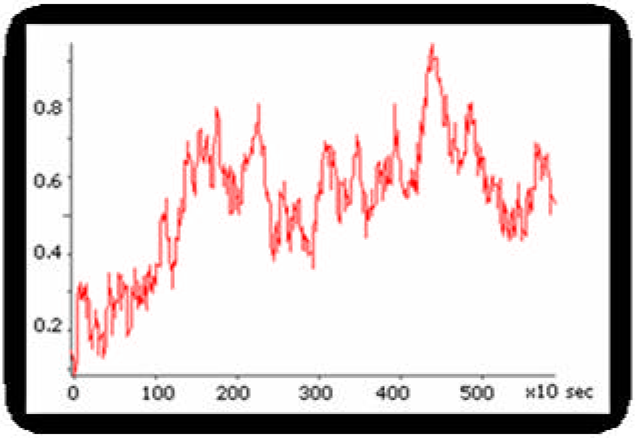


Fig. 15. Relevancy calculation between agent pair X_7 ? X_{12} .

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

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, passive 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 passive 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 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. (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.

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.

EASYNN-PLUS. *Neural Network Planner Software*, http://www.easynn.com, 2002.

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.

ISTAG. *Information Society Technologies Advisory Group*, http://cordis.europa.eu/ist/istag.htm

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.

JAVA. http://www.java.com/

JINI, http://www.jini.org/

JUNG, Java Universal Network/Graph Framework, 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.

In Special Issue of Ambient Intelligence of the ACM Transaction on Autonomous and Adaptive Systems, 2008

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.

SALUATION, http://en.wikipedia.org/wiki/Salutation_%28computing%29

TRAJAN. *Trajan Neural Network Software*, http://www.trajan-software.demon.co.uk/

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

UPNP. *Universal Plug and Play*, http://www.upnp.org/.

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, 1965.

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