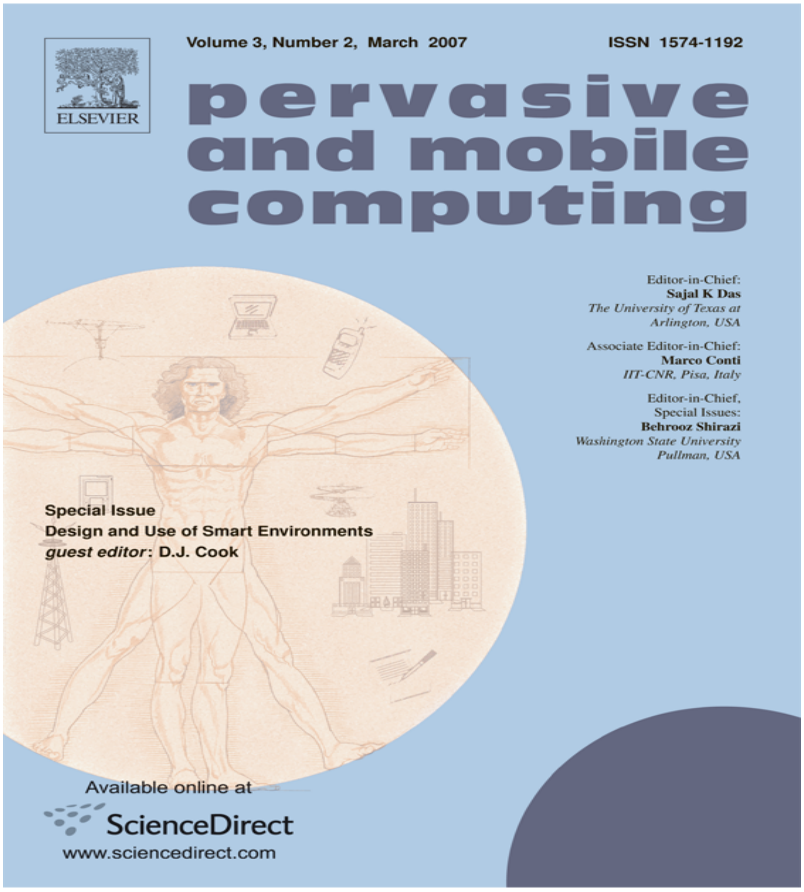

Volume 3, Number 2, March 2007
ISSN 1574-1192
ELSEVIER
pervasive and mobile computing
Editor-in-Chief:
Sajal K Das
The University of Texas at Arlington, USA
Associate Editor-in-Chief:
Marco Conti
IIT-CNR, Pisa, Italy
Editor-in-Chief, Special Issues:
Behrooz Shirazi
Washington State University Pullman, USA
Special Issue
Design and Use of Smart Environments
guest editor: D.J. Cook
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect
Pervasive and Mobile Computing 3 (2007) 117–157
pervasive and mobile computing
www.elsevier.com/locate/pmc

Intelligent association selection of embedded agents in intelligent inhabited environments

Hakan Duman*, Hani Hagras, Vic Callaghan

Department of Computer Science, University of Essex, Wivenhoe Park, Colchester CO43SQ, UK

Received 23 November 2005; received in revised form 7 June 2006; accepted 15 June 2006
Available online 8 August 2006

Abstract

Recent advances in technology and manufacturing have resulted in more powerful and smaller processors to be embedded in the various artefacts within smart environments. Most of these artefacts are network enabled and thanks to pervasive networking such artefacts can communicate and collaborate together to support our daily lives. Furthermore, these artefacts can also be equipped with embedded agents to provide intelligent reasoning, planning and learning capabilities. However, the multitude of interconnected devices and artefacts can result in major network and processing delays as well as creating inherent complexities in programming and configuring smart environments to personalise themselves to suit the individual needs. Hence, a major challenge to the design and use of smart environments involves finding the best set of device associations and interconnections that are most suitable to the environment and user needs. In this paper, we will present a novel intelligent method for reducing the number of associations and interconnections between the various devices and artefacts within smart environments to minimise the network and processing overheads while reducing the cognitive load associated with configuring and programming smart environments.

Keywords: Intelligent association selection; Smart environments; Embedded agents; Fuzzy systems; Machine learning

* Corresponding author. Tel.: +44 1206874880.
E-mail addresses: hduman@essex.ac.uk (H. Duman), hani@essex.ac.uk (H. Hagras), vic@essex.ac.uk (V. Callaghan).

1574-1192/$ - see front matter
doi:10.1016/j.pmcj.2006.06.001

1. Introduction

Over the last several years there has been an accelerated growth in embedded computational artefacts fuelled by the recent advances in microelectronics, networks and internet technologies. Recent figures show that about 8 billion microprocessors were produced with only 2% of them going into PCs [33]. The remaining 98% ended up as part of the pervasive fabric of computing that's being woven around and through our lives via a wide range of tiny embedded computers integrated within various artefacts. Such artefacts include mobile phones, home entertainment systems, fridges, washing machines, kitchen appliances, security systems, cars, transport systems and even our clothes and furniture [34]. Most of these artefacts are network enabled and thanks to pervasive networking such artefacts can communicate and collaborate together to support our lives. These embedded computational artefacts can contain intelligent agents to form *embedded agents* which provide intelligent reasoning and decision making. The embedded agents operate in a non-intrusive manner to personalise themselves to the user's needs and preferences by learning from their behaviour and thus configuring and controlling the user's environments on their behalf. The intelligent mechanisms used within the agents should be able to operate on the limited computational platforms of the tiny embedded devices that populate the smart environments. There is a need also to provide an adaptive lifelong learning mechanism that will allow the system to handle the uncertainties and adapt to the changing environment and user preferences over short and long term intervals [7,18]. Furthermore it is important that these intelligent mechanisms represent their learnt decisions in a form that is transparent and can be easily interpretable and analysed by the end users [7,18].

Intelligent Inhabited Environments (IIEs) constitute an important subset of smart ubiquitous computing environments [34] where IIEs are living spaces (generally, any accessible and habitable area for people such as homes, offices, hospitals, cars, etc.) decorated with a vast number of intelligent embedded agents. An IIE is characterised by its ubiquity, transparency and intelligence. It is ubiquitous because the user is surrounded by a multitude of interconnected embedded systems which form a pervasive infrastructure. Its transparency is due to the invisible nature of the computing based artefacts being seamlessly integrated into the surrounding environment [3,17,18]. Embedded agents can provide the intelligent 'presence' as they are able to recognise the users and can autonomously program themselves to the users' needs and preferences by learning from their behaviour to control the environment on their behalf. Thus these agents can reduce the cognitive load associated with configuring and programming intelligent environments.

In IIEs, the myriad of individual intelligent embedded computers would enrich the user environment, however, it is the interconnections and associations between these various embedded devices that will enable these devices to cooperate and complement each other to empower the user through creating an IIE that is 'aware' of his presence and context and is *sensitive, adaptive* and *responsive* to his needs in an unobtrusive way. The interconnections and associations between the various embedded devices should personalise themselves and adapt to the changing user needs and preferences. However, even for a small number of embedded computational artefacts, it is very difficult to find the best possible associations and interconnections between the various devices and artefacts that will result in satisfying the system and user needs. In addition, handcrafting associations and hardwiring them will be unsuitable for dynamic environments like IIEs where unforeseen changes of the ensemble can occur such as disconnecting older or defective devices or bringing in new devices for embellishing the environments; moreover the user preferences might also change with time.

As the number of interconnected devices, networks and agents increase and populate everywhere within the IIE (which is implied in the ubiquitous computing environments vision [34]), it will prove almost impossible to program and set up the interconnections and associations between the various devices to personalise the IIE to suit the individual needs. In addition, increasing the number of associations will cause major network and operational delays which will cause big problems for the embedded agents which operate on the limited hardware platforms of the computational artefacts within IIEs. These platforms have limited computational power, network connectivity and memory storage requiring the agents to operate economically and efficiently by reducing the associations and interconnections to other agents thus minimising the network overhead, computing processing as well as storage for learnt rules within the rule base. Therefore there is a need for an intelligent non-intrusive system that is capable of finding the best and most relevant interconnections and associations between the devices and agents operating within the uncertain smart environment that have large numbers of potential interconnected devices. This system should operate within embedded agents in a non-intrusive, flexible and adaptable manner to satisfy the user needs. This necessity has been regarded as one of the Grand Challenges for Computing Research within the next century by the UK Engineering and Physical Sciences Research Council (EPSRC) [21]. This work aims to be a step towards addressing these challenges.

In this paper, we will present a novel Intelligent Association System for embedded agents operating within IIE. The agents will reduce the number of associations and interconnections to the other devices and embedded agents to minimise the network overheads and the unneeded extra processing created by the redundantly associated devices while satisfying the environment and the user personalised needs. The intelligent embedded agents are based on the Fuzzy Logic Controller (FLC) which has been credited with providing an appropriate framework for generating human readable models for complex systems [17,18]. 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 IIEs. 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. Each agent is a dynamic and autonomous entity, capable of communicating and cooperating with the other existing embedded agents. We have performed unique experiments in which our intelligent embedded agents have learnt and adapted to the behaviour of a user spending five consecutive days in the Essex intelligent Dormitory (iDorm) which is a unique test bed for IIE research. Moreover, the agent, while learning the behaviour of the user, has also successfully limited the number of associations to only those agents that are relevant and important to it without dropping the overall system's performance.

The rest of this paper is organized as follows. In Section 2, we will review related work in the area of IIE followed by an introduction of our test beds for IIE research in Section 3.

In Section 4, we will describe the nature of associations and how they are used in this paper. The Intelligent Association System, its notions and definitions are presented and explained in Section 5. In Section 6, we will introduce our fuzzy based embedded agent that is capable of lifelong learning and adaptation to the user behaviour in an online and non-intrusive manner. In Section 7, we will introduce our Intelligent Association Selection Algorithms for the fuzzy agents. Section 8 discusses the experiments and results while Section 9 provides the conclusions and future work.

2. Background and related work in IIE

During the last few years, the interest in smart environments research has increased popularity and much work is being conducted in the area of IIE. The Slovak University of Technology in Bratislava [1] have developed EUNICA which is an intelligent household environment where the user is surrounded by a multitude of interconnected devices on a network that is invisible for the user. The system is intended to deliver various home related services to the users and exhibit some form of intelligence, e.g. it is able to recognize each individual in the household and adapt the behaviour according their needs. In addition, it is able to recognize specific events (such as time or movement of a user) and act upon various situations. The heart of EUNICA is the agent based control unit where any artefact can be connected to the control unit using various types of communication media, such as Bluetooth, cable, etc. Monitoring and controlling devices are visualised on simple mobile Java based user interface devices connected to the control unit using the Bluetooth wireless communications technology operating on physically compact devices, such as PDAs. In EUNICA, they display and allow browsing information received from the control unit (represented in the EUNICA Markup Language that is based on XML) and send the user requests back to the control unit.

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 [19]. The care agent included into the IIE uses 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 proposed operating system for the IIE which carries 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 Oxygen at MIT [26,27] centres around two rooms containing cameras, microphones, an X10 controlled lighting system and a multitude of computer vision and speech understanding systems that help the system interpret what people are saying, where they are saying it and what interactions and activities are taking place. The system responds accordingly using speech synthesis when spoken to. The vision system is able to intelligently train itself for a particular environment in less than five minutes using projectors displaying a simulation of someone performing the training. The agents operate in a rather independent intelligence level where each sensor resides in a particular place and uses various local resources [6]. The agents export selected functionality to the network and communicate with each other to share those functions. The agent interaction is ad hoc;

however the connection between the agents has to be established manually or according to the same type of sensors. The system architecture is quite open and flexible by means of adding new agents and creating new services. Each agent comes with a description and capabilities that are announced via broadcasting, discovered and used by other agents in the domain.

Microsoft's EasyLiving [2] project is concerned with the development of architecture and technologies for smart environments — with particular focus on a living room environment in the home. The environment contains a desktop PC, a number of displays (including one large display), speakers, sofas and a coffee table. Services are provided to augment this home environment, such as automated light controls, location-based playing of music (tailored to the user's preferences), and automated transfer of items of interest to multiple displays. The EasyLiving environment can dynamically aggregate network-enabled sensor and actuator devices, such as keyboards and mice, even if they belong to different computers in a space. In addition, by using a geometric model of the room and taking reading from sensors, it aims to provide context aware computing using video tracking and recognition.

The goals of the Aware Home [5,23] project are to investigate what kind of services can be built on top of an environment that is aware of the activities of its occupants. Besides building models of human behaviour to aid computers in decision making, it aims to support older individuals to 'age in place' by helping them to maintain a good diet, take medication when required, notify family members of their well-being and support everyday tasks. The large number of sensors deployed in the Aware Home range from trip sensors (that detect when a door has been opened or closed) and motion detectors, to higher-fidelity sensors, e.g. embedded microphones and cameras. These sensors, all centrally connected to a computer, are mostly used to determine the current activities, facial expressions, locations and gestures of occupants. The sensors and actuators are fixed and dedicated to performing specific tasks only.

The Adaptive House [28] 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 [29].

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 [32]. 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 maximise the comfort and productivity of its inhabitants while minimising the operation cost.

Including reasoning, planning and learning in devices and artefacts and hence embedding intelligence was suggested by researchers at the University of Essex [3]. These

Fig. 1. The intelligent Dormitory (iDorm).

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 analysed for online learning and adaptation of the agents embodied in the iDorm such as the ISL [17,18] and AOFIS [7]. Both ISL and AOFIS employed fuzzy logic based embedded agents that seek to particularise (rather than generalise) 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 on going projects consist of the development and experimentation with new types of autonomous embedded agents such as 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 [8].

In this section, we have reviewed some of the IIEs that are most relevant to the work described in this paper. The overall intention of the aforementioned projects is to build smart environments populated with large numbers of embedded artefacts to assist us during our daily lives and boost productivity, efficiency and comfort. However, the vast number of artefacts and consequently embedded agents in IIEs would cause major network and processing problems as well as making the manual programming of the devices' associations almost impossible, therefore there is a need for research into automatic and adaptive association strategies in IIEs. From the above discussion and to the authors' knowledge, no work in IIEs has investigated creating a method to reduce the associations and interconnections between the various devices, artefacts and agents to those most needed while satisfying the user's personalised needs which will be the focus of this paper.

3. The Essex test beds for IIE

The intelligent Dormitory (iDorm), shown in Fig. 1, forms one of the main test beds for IIEs 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 [17,18,22]. 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 IIE 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 artefacts are seamlessly integrated into the environment. The embedded agents will provide the intelligent 'presence' as they are able to recognise the users and can autonomously program themselves to the users' needs and preferences by learning from their behaviour to control the environment on their behalf.

The agents are embedded within the various artefacts within the iDorm 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 the *internet Fridge* as well as the *mobile robots* that can navigate within the user environment to serve the user needs such as getting drinks and medicine [9,12].

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 [20]. UPnP provides an architecture for pervasive peer-to-peer network connectivity of intelligent embedded agents, wireless devices and artefacts 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 [18]. 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 [18]. 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

Fig. 2. The intelligent apartment (iDorm-2).

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 IIEs.

Another unique test bed, also located at the University of Essex, is the Essex intelligent apartment (iDorm-2) which is shown in Fig. 2. The iDorm-2 is a spacious two bedroom flat with a kitchen and a bathroom. The iDorm-2 provides a flexible test bed for research into IIEs and it offers the possibility for examining the deployment of embedded agents and sophisticated user interfaces within the intelligent environments of tomorrow. Although the iDorm-2 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 [3]. 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 iDorm-2 with embedded agents making them become dynamic and open environments where numerous new agents 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. Once new agents are included the user can configure and analyse them by using the *IIE Editor*.

The IIE Editor [10,11], also referred to as *IAS Editor*, is primarily concerned with assisting non-technical users during the design and use of their smart living spaces. The editor visualises not only the agent's properties and information, but it also allows the user to configure and equip the agents with behaviours in the form of human IF-Then fuzzy rules. Although it is not necessarily important to define rules beforehand, it is useful though if the user's expectations from the agent and IIE are known and clearly defined. Rules can be set to being non-changeable so that they will exist during the lifetime of the agent. This becomes extremely beneficial if one has to set up safety rules [11].

The IIE Editor can also be used as the control interface of the iDorm and iDorm-2 to control and manipulate any embedded agent located in the rooms. The IIE Editor can operate on any networked PCs (Fig. 3(a)), or PDA (Fig. 3(b)) capable of running standard Java programs. It is also possible to interact with the iDorm and iDorm-2 through mobile

(a) (b) (c) (d)

Fig. 3. The iDorm and iDorm-2 control interfaces using (a) PC, (b) PDA, (c) mobile phones, and (d) iFridge.

phones (Fig. 3(c)) using a WAP interface which is a simple extension of a web interface. Fig. 3(d) shows the IIE Editor operating on the internet Fridge (iFridge) provided by LG Consumer Electronics that incorporates a server with touch-screen capability which was mainly used for managing the associations for our experiments.

The IIE Editor graphical user interface is aimed to be user-friendly enabling any user to visually design and configure IIEs (using UPnP as the underlying communication protocol) in a *what you see is what you get fashion* through a large number of assistive features. The IIE Editor allows the user to configure available agents, create and delete associations and rules as well as controlling and testing them in real time with simple mouse clicks. In addition, the user can use predefined macros, e.g. *learning rules by monitoring the user* to explicitly teach the embedded agents their desired behaviours [11]. The behaviours can be obtained by either monitoring all available agents or limit them to a subset of user-specified associations. Through associations the user may also decide to assemble simple individual agents to more complex ones. The basic functions and operational steps of the IIE Editor are briefly described next.

- The *agent Discovery* forms the first function of the IIE Editor and allows the user to discover the embedded agents before any operation on them becomes feasible. Once discovered, the agents can be queried for their services and information.
- *Forming associations* requires the user to indicate and select the desired agents he/she wants to interconnect. During this step, the user is expected to declare his/her intentions of the working functionalities that need to be set up.
- Once all associations have been created, the IIE Editor can be used to *activate the agents and their configuration*. Hereafter, the agents can be controlled and tested as desired in real-time and visually examined at individual and/or associative level.

Fig. 4 illustrates snapshots and pseudo codes of the two procedures used within the IIE Editor. Fig. 4(a) shows how the user can generate associations between embedded agents manually and Fig. 4(b) describes the steps to automatically teach the agents the desired behaviour through monitoring the user. Although the *IIE Editor* is designed to be easy to use and handle countless number of agents, the cognitive load for the user would still become immense to go through all possible associations and configure them one by one, especially when the number of agents increase dramatically which is implied in IIEs. This necessitates research into intelligent association strategies which are able to operate in a non-intrusive way to support the user.

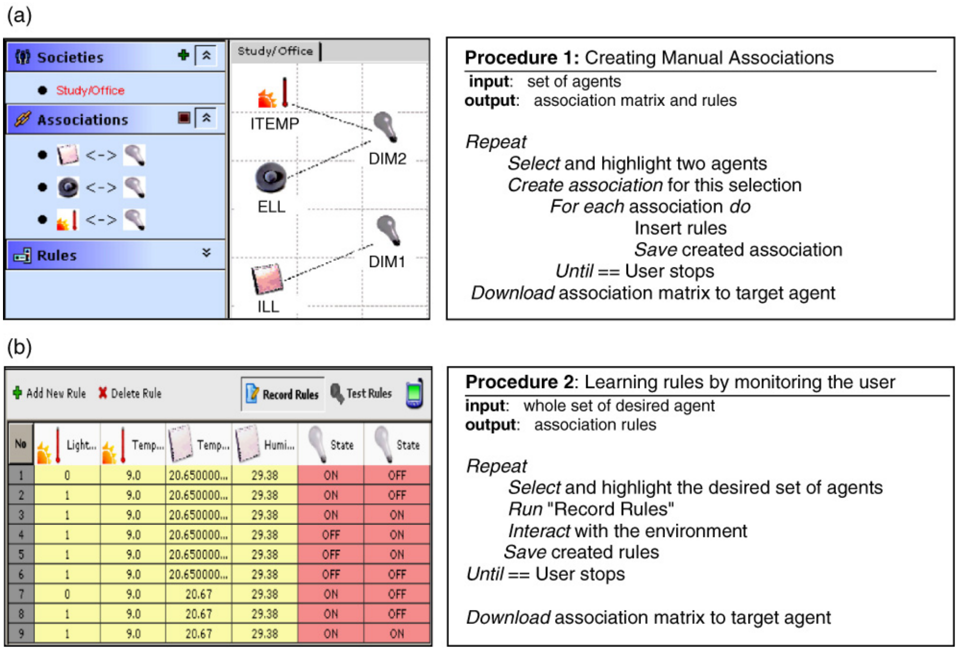


Fig. 4. Procedures to generate associations and rules using the IIE Editor. (a) Generating the associations manually, (b) learning rules by monitoring the user.

4. The nature of associations

Associations are specified by the user either via direct manipulation of the devices or via specially developed tools, like the IIE Editor described in the previous section. However in the following sections we also describe a new paradigm and introduce a new framework that can facilitate devices to learn their interconnections intelligently. In this section, first we briefly discuss the benefits of associations for the user and environment and then describe the nature of associations as they are used throughout this paper.

4.1. Benefits of associations for the user and environment

Large numbers of networked devices and artefacts are increasingly invading our everyday lives to form our 'new' living environment. In such spaces there is wide scope for utilizing networked computer-based devices to enhance living conditions. Some of these devices can form part of the building infrastructure and are static in nature (e.g. HVAC), others are mobile (e.g. phones), or nomadic (e.g. TVs) [3]. However, they all share a common characteristic: all of the devices can be freely interconnected to shape more complex systems depending on their purpose of use and functional properties. In other words, devices are no longer used in individual, isolated or fixed forms they are rather grouped into more powerful apparatus. From the simplest interconnection and association of devices, it is possible to see emergent use listed in the following simple applications:

Home comfort	User-specific control of light, heating, HVAC, blinds. E.g. the heater can be switched on when the temperature drops or the desk lamp switches on if the user sits on the chair, etc.
Energy efficiency	Energy-optimised HVAC control, light control, absence or vacation mode. E.g. the room lamps switch off if no one is in the room, the HVAC operates in the most efficient way and regulates the heat according to the current outdoor and indoor temperature, etc.
Safety	Access control, panic button, alarm (fire, water, electricity), health monitoring. E.g. depending on the heath condition provided by wearable sensors the doctor of the patient is contacted directly via mobile phone messages
Support	Extended comfort and safety functions (support for disabled), and servant robot. E.g. a servant robot delivers food and medicine to a bed-ridden person that directs the robot with a remote control

The listed examples are by no means complete and the emerging applications that can result are totally dependant on the functional objective preferred by the user. In case of a functional breakdown of a device, e.g. a light switch, the lamps associated to the broken switch can be bypassed to another switch available within the environment. Moreover, a switch, for instance, is no longer statically related to a lamp — a switch on the wall can also be used to switch on/off the TV or a washing machine. For many people this might not sound like a logical connection; however we only intend to show the possibilities on how beneficial associations become if they can be freely and randomly created.

Clearly, many potential problems arise with this new association concept, e.g. the cognitive load for the user increases immensely since it will be the user's responsibility to manage and configure his/her environment populated with all of these networked appliances and devices. Thus, in the next sections of the paper we propose novel mechanisms to overcome these burdens however it cannot be overseen that networked devices and appliances and their associations will enhance our living conditions and allow us to be part of the design process of the infrastructure we live in.

4.2. Usage model of associations

A living environment such as the iDorm or iDorm-2 consists of a set of interconnected physical devices that are attached to the network. A device in a network can receive low-level network signals emitted by one of the devices connected to the same network. Two devices are also connected if a path via a set of connected devices of this network exists. The devices broadcast or multicast the messages or signals to the network and the targeted device gathers them according to the network address.

Different types of networks, like in the iDorm and iDorm-2, and various subnets based on proximity (rooms or buildings) etc. can be coupled together to form one large private area network (PAN) that hosts all the networked devices and appliances that are available to the environment's user. The network layer as shown in Fig. 5(a) combines these interconnected physical devices of different subnets and/or networks through routers and messages are broadcasted to all members of this network.

Since networks and devices can be of a different type (e.g. X10, LonWorks), they are all interfaced to a common middleware, UPnP, which enables the environment to

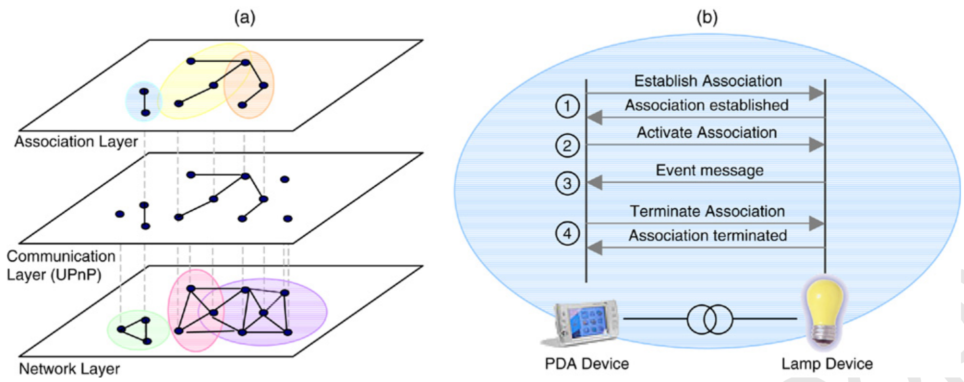


Fig. 5. (a) The usage model of associations. (b) Type of messages shared between associated devices.

become a heterogeneous device network, where devices can communicate with each other and exchange useful information. The UPnP communication layer also allows selective multicast, which means that only those devices requesting information from other devices receive them; the rest of the devices ignore the messages and don't process them.

An application running at the association layer of each device, as depicted in Fig. 5(a), can control and direct the associations to other devices e.g. by subscribing to services that it requires information from. Using this method, the devices receiving multicast messages but not associated with the sender do not process the message by ignoring them.

Hereafter, with association we intend to describe this higher level interconnection between devices and their services.

4.3. *Types of information exchanged in associations*

One of the main reasons for devices to associate with other devices is to exchange data; in particular to receive data and information from other devices to incorporate them into their control and decision making processes. The type of information is multifolded and categorized in Device Description, Service Description and Events.

A device can be regarded as a container of services and nested devices. For example a HI-FI device would consist of some services, plus some other sub-devices such as a CD player and a radio tuner. Each device maintains an XML description of its properties and a list of links to the description of its nested devices and services.

A service represents a function that the device can execute. It is modelled through actions (methods) and state variables. A history of states is stored within the state table of each device. Whenever a state changes the state table is automatically updated and publishes a state variable changed event.

A device associated to other devices may request to receive state variable information in the form of events. For this, the device establishes an association to the service and through this to the device it requires the event information from. This action activates the association as described previously.

4.4. The association process

The association process is based on four steps as depicted in Fig. 5(b) and is described as follows. (1) One of the devices sends a request message to establish an association to a target device. The message contains information concerning the local and remote device. When the remote device receives the message, it sends a positive response and an instance of an association is thereafter created. (2) Once an association is established, the local device sends an activation message to open a communication channel between the interconnected devices to enable secure data exchange. (3) The data exchanged is event-driven meaning that in the case of a state change the remote device contacts every single device that it has an association with through a time stamped 'Event Message'. (4) Finally, if one of the two devices leaves, breaks or intentionally requests to break the association, a 'Terminate Association' message is sent to the remote device. A successful termination request returns a positive response that causes the association to dissolve. In UPnP the association process is referred to as event subscription and event un-subscription and forms the underlying device communication middleware for the Intelligent Association System that is described in the next section.

4.5. The maximum number of associations

One important aspect of device association is to determine the number of possible associations each device can have. Since each networked device and appliance has different computational and communication capabilities, the number of associations this device can have varies. For example, a PC can handle more associations than a PDA because of larger processing power and networking facilities. In this paper, we assume that each device comes with technical constraint properties, provided by the manufacturer. This information is included in the Device Description (see previous section) and suggests the maximum number of simultaneous associations a device may have.

5. The intelligent association system framework

In this section, we will introduce the architectural framework of the *Intelligent Association System (IAS)*, which allows the intelligent selection of the most relevant embedded agent associations within the IIE. We start by describing the high level architecture of the IAS and then explain the notion of associations within the IAS. Finally, we show the effects of irrelevant associations within the system and explain the need for intelligent techniques to select associations.

5.1. Overview

Within an IIE different types of agents exist and can be explained by looking at their individual purpose and services. A granulated categorization of an agent that is used within an IIE is defined as follows [10,11]. (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

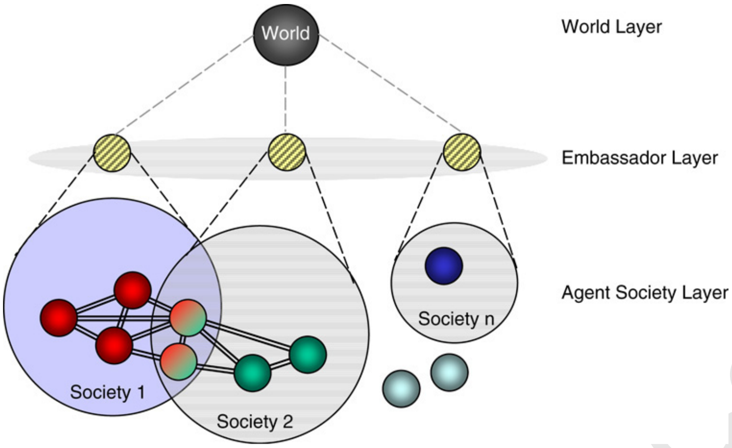


Fig. 6. The Intelligent Association System (IAS).

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 embedded agent is capable of acting on its own to execute tasks, the agents perform more useful and complex behaviours if they collaborate together towards a common objective, thus forming an *embedded agent society* [10,11]. Grouping embedded agents into societies aims to reduce the complexity of associating and managing the huge number of embedded agents within IIE, where the IAS will deal with a manageable number of agents within each society.

The embedded 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 Embedded Embassador agent, here after termed *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. 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 [10] and [11]. The *Intelligent Association System (IAS)* architecture integrates the above components into three layers as shown in Fig. 6.

5.2. The notion of associations within IAS

As mentioned above forming societies requires embedded agents to establish associations among each other. The IAS defines associations on top of the UPnP middleware and enables an agent to establish an association by subscribing to one or more services of another agent.

The notion of association within the IAS is as follows. An association is a physical or virtual communication channel and link between agents sending information which are expected to be useful and vital for the agent. The direction of this exchange is either uni- or bi-directional. Unidirectional associations are mostly set from a passive agent to a smart or intelligent agent. If both agents are at least of a smart type then the association may be set to bi-directional. The lifetime of an association is either permanent or temporary (e.g. mobile or portable agents).

The associations are selected and established as follows [11]:

- *Manual association* where agents are linked together by the user (e.g. using the IIE Editor)
- *Smart or automatic associations* where agents are linked together based on a search which can be ontology-based to find agents that are manufactured to be used together, e.g. a switch with a lamp; once the search is successful the agent automatically establishes an association to the target agent
- *Intelligent associations* where agents are capable of learning and managing their associations autonomously by using intelligent techniques to achieve self-configuration and fault tolerance.

In the next subsection, we will discuss the need of an intelligent association selection algorithm.

5.3. The need to find relevant associations

An IIE is expected to be equipped with large numbers of agents. Some of them are grouped into societies, some are waiting to become part of a society and others operate individually. There is a need to develop an intelligent adaptive system that will be able to learn and find the most relevant associations to the other agents for the following reasons:

- In an open and dynamic IIE, an agent may become less important over time or even leave or break down. On the contrary, new agents may join and publish new services which may be more relevant than the current ones. In addition, the user's desires may change over time necessitating the reconfiguration of the associations of the embedded devices within the IIE. An intelligent embedded agent should be capable of noticing these changes and adjusting to new conditions through self configuration and reorganization. It is obvious that a user-dictated manual association system considerably fails to deal within these highly dynamic environments populated with hundreds and thousands of embedded agents.
- By establishing fewer associations, the system can obtain only a rough interpretation of the relation between the agents with low resolution; this can be viewed as obtaining high level knowledge. On the contrary, by defining more associations, the knowledge representation becomes more granulated with more emphasis on detailing the relationship information. However, as the number of associations increases, the more complicated it gets to generate an accurate model as in high-dimensional spaces the sampling data become very sparse. Therefore, blindly associating to all agents is not

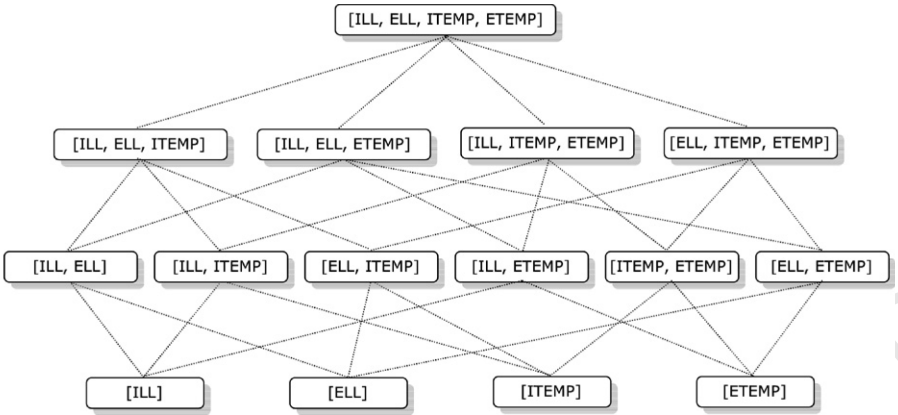


Fig. 7. The 15 possible associations of four embedded agents.

a good idea. Hence, only those agents that are most influential should be associated to any given agent.

- An exhaustive search when trying to find the associations of the most relevant and important agents requires going through a huge set of association combinations which is computationally expensive where given N available embedded agents there are $2^N - 1$ possible associations between those agents. Fig. 7 shows all possible 15 associations for only four available embedded agents within an IIE. It can be seen that for an IIE with a large number of embedded agents this search would become incredibly time consuming for an agent with limited processing and communication capabilities.
- Due to the physical restrictions of embedded agents, the network bandwidth and communication overhead are forced to be kept to reasonable sizes. This means that the simultaneous associations to other agents have to be limited to relevant services from other agents. The following example aims to illustrate the impacts of redundant and irrelevant associations on the network and communication level. An intelligent embedded agent, such as a desk lamp, performs its predefined behaviours by monitoring the environment where the events triggered provide information about the environment through associated agents. Whenever the state of a linked agent changes, the communication protocol based on UPnP sends a message to the desk lamp to notify about the event occurrence. This is also the case if the desk lamp changes its own state and hence an event and feedback gets published to every agent and the desk lamp agent respectively. Moreover every event is timestamped and tagged with a unique ID. It is assumed that the IIE contains many redundant agents that randomly send irrelevant events in different time intervals where the delays arising from the low-level UPnP protocol are ignored in this example. Fig. 8 shows the results obtained from this example where we started experimenting with the desk lamp agent having one association and then we gradually increased this number of associations to other agents to twenty; the majority of these associations are assumed to be irrelevant that send useless information to the desk lamp. With the establishment of a new association, the desk lamp is programmed to broadcast a change of its stage in specific intervals

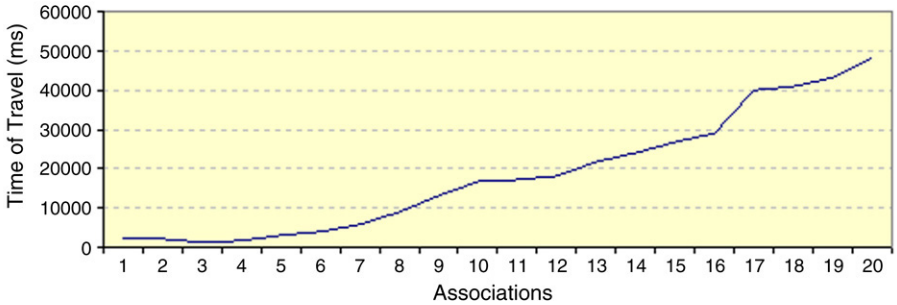


Fig. 8. Time of travel of events.

and timestamps this event when it happens. The difference of time when the desk lamp changes its state until the time that this change returns back as a feedback event is defined as *Time of Travel*. As depicted in Fig. 8, the *Time of Travel* increases almost exponentially in relation to the number of newly associated agents. In real-time control, the delayed feedback would decrease the performance of the overall system and would lead to instability and unreliability. To overcome these critical problems it is inevitable to include an intelligent approach to limit the associations to relevant agents only.

5.4. Potential harm caused by removing associations

In the previous subsection, we have listed the necessity of removing irrelevant associations; but how safe is it to remove an operating association and what harm would it cause to the overall performance and system functionality? The worst case scenario would be that the system stops operating in the correct way or even fails to operate totally.

An association is removed either temporarily or permanently. Temporary unavailability of an agent, for instance a mobile agent (PDA), would cause the associated agents to stop receiving messages from it thus causing the system to freeze or fail. However this depends entirely on the communication infrastructure. In event-based message systems like UPnP the associated agents only exchange data if a state changes, meaning that there is no polling of information that could cause the system to freeze. E.g. if a mobile agent leaves the environment, the agent that it has an association with will no longer have an update of its current state and will always assume that the state remains unchanged until the mobile agent returns. The same applies for an agent that breaks down. Permanent removal of agents would require the revision of the association of the agent to another agent. Using the IIE Editor, the user can check the availability and state of other agents that can be candidates for new associations or used to re-establish broken associations.

In addition, the agent's decision making process can suffer from the removed associations. E.g. every agent (as will be described in the next section) has a Rulebase where a record of associations and history of actions is kept. Once an association is removed the Rulebase becomes incomplete in the sense that an attribute of the rule becomes unknowable. The Rulebase becomes inconsistent and might result in incorrect actions for different conditions in the environment. A method to overcome this problem is to allow the

Rulebase to be dynamic rather than static: not only the actions and rules should be learnt but also the attributes for different conditions and different association structures. In the following sections of this paper, we present an example [13] on how to achieve a flexible and dynamic Rulebase.

In general, it can be said that removing associations do not cause any harm to the system or environment provided the embedded agents are programmed to be distributed, flexible and dynamic. In addition, the communication protocol or middleware is one of the key factors to achieve reliable and error-free messaging between agents.

In the next section, we will describe the Fuzzy-IAS agent (F-IAS) which is an intelligent adaptive embedded agent capable of learning from user interactions and adapting to online changes simply by monitoring the environment. At the same time, the F-IAS agent aims to select the associations that are most relevant and important for its behaviours.

6. The Fuzzy-IAS agent (F-IAS)

6.1. Fuzzy logic control systems

Fuzzy logic controllers (FLCs) have been credited with providing appropriate frameworks for generating human readable models for complex systems. A FLC is a model free approach which converts linguistic control information into mathematical control information and can represent a non-linear mapping of inputs to outputs. FLCs use linguistic IF-Then rules as well as fuzzy membership functions which quantify the raw crisp values of the sensors and actuators into linguistic labels such as *normal, cold or hot*. FLCs provide transparent and flexible representations which can be easily adapted due to the ability of fuzzy rules to approximate independent local models for mapping a set of inputs to a set of outputs. Moreover, 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 IIEs. As a result FLCs have been used in IIEs as in [7,17,18].

The FLC is shown in Fig. 9 and 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 [24]. 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 [24].

6.2. The F-IAS agent model

In IIEs, we define the F-IAS agents as those intelligent agents embedded within the actuation devices. The Fuzzy-IAS agent utilises an unsupervised data-driven one-pass approach for extracting fuzzy rules to learn and model the user's behaviours with the novel

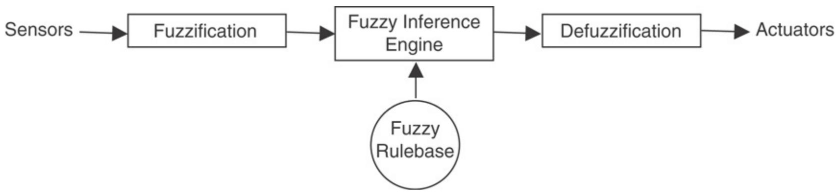


Fig. 9. Flow diagram of the FLC.

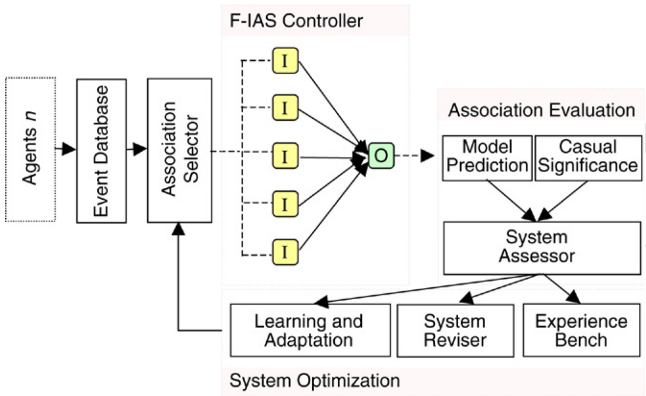


Fig. 10. The F-IAS agent model.

feature of selecting the most relevant associations to the other agents. The F-IAS agent has two learning modes which are an *offline learning* mode and an *online actuation and adaptation* mode. In the offline learning mode, the data is collected by monitoring the user in the IIE over a period of time (3 days in the case of the iDorm experiments). This data then is applied to the learning engine of the agent to produce the F-IAS agent controller that provides an inference mechanism that will produce output control responses based on the current state of the inputs. In the online actuation and adaptation mode, the F-IAS agent will control the environment on behalf of the user and will also allow the rules to be adapted and extended online in a lifelong learning mode. A novel feature included in both modes consists of a system optimisation module aiming to reduce the F-IAS agent's associations to the most efficient and important ones while keeping the agent's performance at the same level.

The F-IAS agent model shown in Fig. 10 involves the following phases and components: the *Event Database* which is formed from the data collection process that monitors and stores the user's interactions based on the *Association Selector* that associates given input agents (indicated as 'I') to the F-IAS agent output (indicated as 'O') either randomly, manually or even intelligently. Once enough data is collected the *Learning and Adaptation* module of the F-IAS Controller runs the *Association Evaluation and System Assessor* modules based on the recorded data to find the most influential and important input agents and then induces fuzzy rules and generates the model required for the control process thus enabling the F-IAS agent Controller to act on behalf of the user and to learn and adapt to changes in a non-intrusive way. During the online actuation and adaptation mode, the *Association Evaluation and System Assessor* continually evaluate the importance and relevance of the input agents based on model prediction and significance and ranks them according to their influence. Over time, an input agent or rule can lose its effectiveness and may be removed by the *System Reviser* to improve the system's performance. An input agent loses its importance when it no longer contributes to the F-IAS efficiency. However some input agents are portable, time or mood-dependent meaning that they should not be omitted but rather saved for certain times. The *Experience Bench* fulfils this function and keeps those agents and rules stored for later reference.

6.3. The agent controller

The F-IAS agent perceives the environment through the sensory information provided by the associated input passive agents and it affects the environment through its actuator based on its learnt fuzzy logic controller that approximate the particularised preferences of the user. We assume that each F-IAS embedded agent has a N:1 relationship meaning that N possible input passive embedded agents can be associated to 1 output. It should be noted that our approaches can easily be extended to a *multiple inputs associated to multiple outputs* relationship but for the sake of simplicity, we will consider that the F-IAS agent has only to control one output actuator. In addition, we should mention that an F-IAS agent output can also be used as an input for another F-IAS agent; however throughout this paper we will consider that the input agents will be mainly the passive agents that are embedded within the sensing devices.

For an IIE, 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 $\vec{x}^k \in \Re^N$, $y^k \in k = 1, 2, \ldots, K$. The fuzzy system rulebase comprises of a set of L IF-THEN fuzzy rules where the ith rule is having the following form:

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

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 uses singleton fuzzification, *max–min* inference method and the height defuzzification, so the crisp output of this controller can be written as follows [24]:

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

where $\overline{B}_i^*$ is the centre of the output fuzzy set of the ith 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

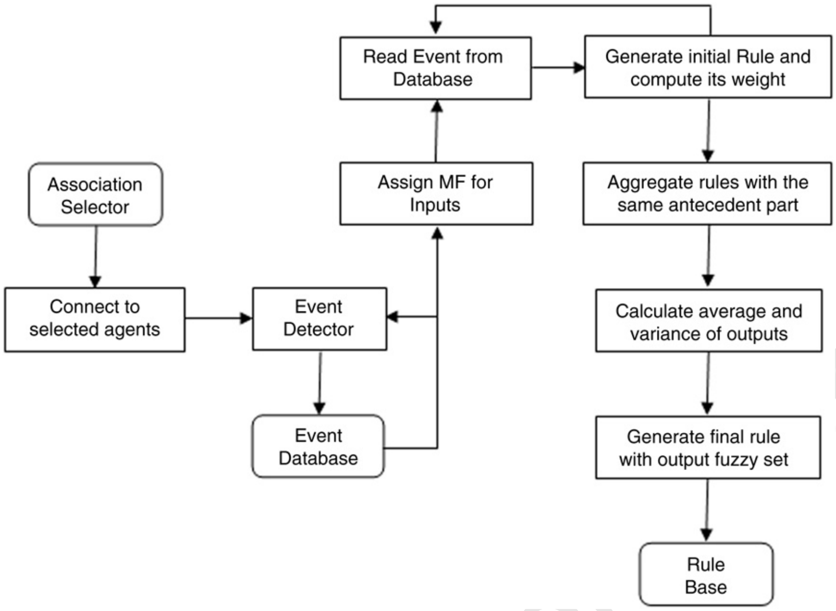


Fig. 11. Flowchart of F-IAS rule induction method.

as adapting, changing and removing the existing rules that are stored in the form of Eq. (1) in the rulebase. The rule induction method of the F-IAS which operates in an online and lifelong learning mode is described next.

6.4. Rule induction method of F-IAS

The rule induction method adopted by the F-IAS agents is based on an enhanced version of the Wang–Mendel (WM) method using a one-pass technique to extract fuzzy rules from a sampled data set [31]. Fig. 11 depicts a flowchart of the rule induction method used by the F-IAS agents. The procedure involves the following steps:

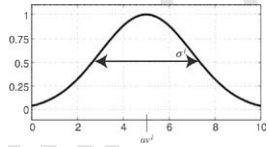
I. The *Association Selector* of the F-IAS agent establishes associations to selected input agents. These agents may have been selected by the user or intelligently (as will be discussed later).
II. The *Event Detector* monitors the user interaction with the associated embedded agents and, in the event of a change, the information is forwarded and saved in the *Event Database*.
III. Once enough data and events have been collected (the data collection in the case of the iDorm experiments lasted 3 consecutive days), assign for each input supplied by an input 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 [7].
IV. Expert rules are allowed and may be combined with the rules induced from the data.

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

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

VI. Repeat *Step V* for all k from $1, \ldots, K$ to obtain K data generated rules in the form of Eq. (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 Eq. (1) where B^{i^*} is a Gaussian fuzzy set. The antecedent and consequent of the obtained rule becomes the following form

$$\text{IF } x_1 \textit{ is } A_1^i \text{ AND } \ldots \text{ AND } x_N \textit{ is } A_N^i \text{ THEN}$$



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

$$av^l = \frac{\sum_{k=1}^{K_l} y_k^l \, w_k^l}{\sum_{k=1}^{K_l} w_k^l} \tag{4}$$

$$\sigma^l = \frac{\sum_{k=1}^{K_l} \left| y_k^l - av^l \right| w_k^l}{\sum_{k=1}^{K_l} w_k^l} \tag{5}$$

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

$$w_k^l = \prod_{j=1}^{N} \mu_{A_j^q}(x_j^k). \tag{6}$$

VII. Repeat this combination for all conflicting groups l to obtain the final rule set which contains L rules in the form of Eq. (1) and store it in F-IAS agent's *Rulebase*.

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) [31]. During the online actuation and adaptation mode, the above procedure allows the rulebase 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 recorded data. However, the objective of an F-IAS agent is also to reduce the large number of possible input agent's associations to a subset of the most important and effective associations without significantly reducing the agent's ability to model the user behaviour: therefore simultaneous procedures in the *Association Evaluation* module evaluates the efficiency and importance of the currently associated input agents. The next section introduces an efficient, reliable and non-intrusive approach for intelligent association selection.

7. Intelligent association selection algorithms for F-IAS agents

One of the greatest challenges to IIEs involves finding the most relevant embedded agent's associations that will satisfy the system and the user requirements [21]. As within an IIE, it is often found that for a given F-IAS agent, some of the inputs provided by the associated input agents $x_1, x_2, x_3, \ldots, x_N$ are redundant and only a subset of these input agents is sufficient and relevant to fulfil a given task or mission. With the increasing number of input agents, these redundant associations to the F-IAS agent will cause major network and processing overheads that the embedded agents will not be able to handle. Therefore, it is vital for the F-IAS agent to determine which of the available input agents are important, relevant and significant and should be associated to. This section introduces the methods used within the *Association Evaluation* and *System Assessor* components of the F-IAS agent to evaluate the relevance and significance of the potential associations based on model prediction and association causality.

7.1. Predictive versus causal significance

The two most common notions of association significance are the *predictive significance* and the *causal significance* [14–16]. Model predictive significance is based on monitoring the user model approximation error when an association to the F-IAS agent is established or removed. The causal significance (based on *single association evaluation*) investigates if an input embedded agent actually affects the associated F-IAS agent (cause–effect relationship) and thus contributes to influence the model prediction. Both the predictive significance and the causal significance mainly seek to answer the following question: "What happens with the F-IAS performance, if an association is established or removed?"

There are three conditions under which dissolving an association will not degrade the performance of the F-IAS agent, which are [4]: (1) the F-IAS agent output does not vary with the change of an input supplied by an associated input agent; (2) the F-IAS agent output change is mainly due to noise with respect to a given associated input agent; (3) the association of an input agent is redundant such that the F-IAS agent's behaviour can be obtained equally by being associated to other input agents. These three conditions occur when an association is redundant, not important or not relevant for the F-IAS agent [4]. The process to evaluate these three conditions is described in the following sections, where the important and relevant sets of associations from the F-IAS agent to the input agents are obtained based on predictive significance (using association subset evaluation) and causal significance.

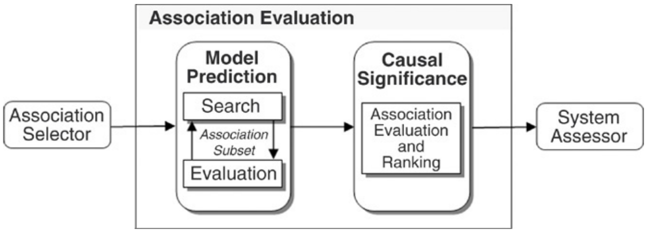


Fig. 12. Flow diagram for the intelligent association selection.

7.2. *The two stages of the intelligent association selection algorithm*

Fig. 12 shows the two stages of the Intelligent Association Selection and how it is used by the F-IAS agent. After the *Association Selector* establishes associations to input agents, the *Model Prediction* module first searches through all possible association subsets and filters the irrelevant associations based on predictive significance which is evaluated according to the user model prediction accuracy achieved with each association subset. The goals of this process are to reduce the set of possible associations available to the F-IAS agent while keeping the agent's performance stable and acceptable. The obtained subset of associations to the input agents are then evaluated and ranked using the *Causal Significance* component. Finally the *System Assessor* can perform a further reduction of the input agent associations based on the rankings performed by the *Causal Significance* component.

But how can the F-IAS agent know if it has successfully found the most relevant and important subset of associations and how can it know when to stop the search and how many associations to select? The answers to these questions are mostly related to the user's purpose on how he/she wants the F-IAS agent to operate and when he/she would consider a significant performance drop. For a given F-IAS agent, the removal of irrelevant input agents and their associations should only be performed if a *limited* decrease (with respect to a given threshold) of the user model prediction accuracy can be ascertained. In our experiments, we have used a threshold that does not allow the user model prediction accuracy to drop below 15%. For a given IIE, several association subsets can lead to achieving a user model prediction accuracy that will lie within the bounds of the given threshold. Hence, the number of rules learnt within the F-IAS agent would form a further selection factor where the association subset with the least number of associations and rules (given that the user model prediction accuracy falls within the given threshold) will be selected to form the most relevant and important set of associations from the F-IAS agent to the input agents.

In the next section, we will describe the methods and algorithms we used for the intelligent association selection within the F-IAS agents based on the above mentioned evaluation criteria.

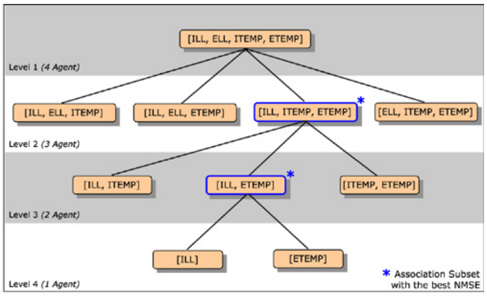
7.3. *Searching and evaluating associations subsets based on predictive significance*

The approach of finding the subset with the most relevant associations is based on combinational search and predictive significance analysis where the algorithm used within

```
Algorithm 1: Searching through association combinations
input:    set of input agents
output:   best subset of relevant associations
variable  n= no of input agents
          w = association subset level

Repeat
    l = n
    Repeat
        Select association combination m for l
            Create model and calculate error for m
    Until== (all possible m)
    Compare all m at l and select the subset with the least NMSE
    Assign selected subset as new l and reduce n by 1
Until (n == 0)
```

(a)



(b)

Fig. 13. (a) Pseudo code for the search through possible subsets of association to input agents. (b) An example showing the progress of the search given 4 input agents.

the F-IAS agents performs a search through the space of possible combinations of associations to input agents as follows:

1. *Starting point*: The initial subset of agents associations can affect the direction and duration of the search immensely especially if the number of available input agents is huge. One position is to begin with an empty set and successively connect to new embedded agents as needed, in this case the search is said to proceed *forward* through the search space. Conversely, the search can begin with all the available associations and successively remove agents as needed, in this case the search proceeds *backward* through the search space. The algorithm used within the F-IAS agent is essentially a *backward* search procedure that starts by associating all the input agents within the agent society (or within the whole IIE if the total number of agents is not so large) and the algorithm then reduces the number of associations at each stage [25].
2. *Search organization*: An exhaustive search of all possible association combinations is computationally expensive as with N initial input agents there are $2^N - 1$ possible association combinations. The F-IAS agents use a stepwise procedure to reduce the search to only $N(N+1)/2$ possible combinations, where the agent breaks the search into different levels where the first level involves the association combination of all the N agents and the following level will involve the association combination of $N-1$ agents and so on. The agents that exist within a given level w ($w = 2, \ldots, N$) are those agents whose association combination performed best in the previous level in terms of approximating and predicting the user model. Fig. 13(a) shows the pseudo code for the stepwise procedure to reduce the search as used in the F-IAS agent. Fig. 13(b) shows an example that depicts the steps of the procedure for four input agent, namely ILL, ELL, ITEMP and ETEMP. The figure illustrates the significance of the stepwise procedure which reduced the association combination search space from 15 association combinations to only 10 association combinations.
3. *Evaluation strategy*: Each association combination is evaluated using the *Association Evaluation* module, which uses the *Normalized Mean Square Error (NMSE)* to obtain the predictive significance for the user model.

As mentioned above, for a given IIE, several association subsets can lead to achieving a user model prediction accuracy that will lie within the bounds of the given threshold.

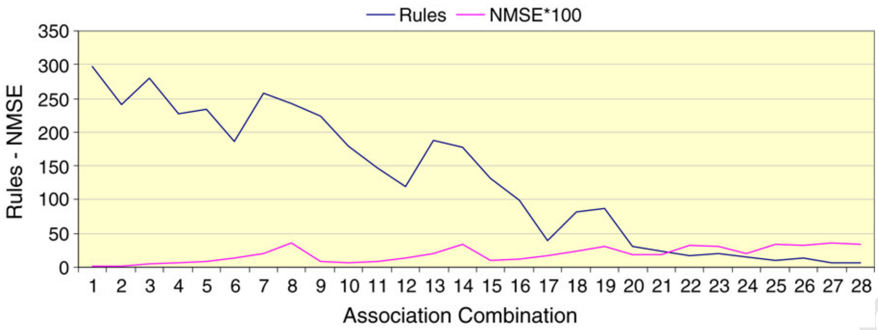


Fig. 14. Evaluating the predictive significance of 7 input agent association subsets.

Hence, the number of rules learnt within the F-IAS agent would form a further selection factor where the association subset with the least number of associations and rules (given that the user model prediction accuracy falls within the given threshold) will be selected to form the most relevant and important set of associations from the F-IAS agent to the input agents. It is clear that as the number of associations decreases, the number of rules learnt for the F-IAS agent will also decrease. However, reducing the number of associations drastically will lead to forming incomplete and inaccurate user models which will result in high values of the NMSE. Fig. 14 illustrates this issue and shows the results obtained from an example of an F-IAS agent (DIM2) with 7 input agent associations including ILL, ELL, ITEMP, ETEMP, BED, CHAIR, and HOUR. It is obvious that the best model accuracy (based on least NMSE) is obtained for the association combination 1 which involved associating to all the input agents; however the number of rules is maximum. On the other hand, the number of rules is minimum and the NMSE is maximum at the association combination number 28 which has only one agent. Thus we aim to find the best association combination subset that will use the least number of associations (thus providing fewer rules in the F-IAS agent rule base) while producing a reasonably accurate model that will lie within the given tolerance (thus having less NMSE). In Fig. 14, it is noticed that the association combinations subsets 1 to 5 satisfy the given 15% tolerance as their NMSE falls within this tolerance. However, it is seen that the association combination number 4 will produce less rules and hence less associations. Combination number 4 involved associating the DIM2 F-IAS agent to the following input agents: ILL, ELL, CHAIR, BED, HOUR after eliminating ITEMP and ETEMP input agents.

The selection of the best subset of associations based on the association combination search algorithm as described in this subsection and the pure evaluation of the F-IAS agent based on the generated model's prediction significance can be misleading and wrongly interpreted. Any association combination to input agents leading to an accurate model prediction and approximation doesn't necessarily mean that all of these selected agent associations are effective individually and efficiently contribute to the F-IAS agent. This is mostly the case if those agents are correlated among each other where the predictive significance would regard the significance of these correlated input agents as equally significance although some of them might not be at all relevant and important for certain behaviours. Thus, the second evaluation for searching the association subset within the

F-IAS is based on the causal significance which is based on the fuzzy casual association criteria which will be described in the next subsection.

7.4. Fuzzy causal association criteria (causal significance)

After using the predictive significance to generate an initial model of the F-IAS agent that incorporates a subset of associations, the causal significance of the individual input agents can be determined and the agents can be ranked so that we can achieve a further reduction of the input agent associations. The basic idea of this step is to analyse the *cause–effect relationship* of the input agents to the F-IAS agent. For this, the F-IAS agents uses the structural notions and descriptions of fuzzy cognitive maps (FCM) [24] allowing causal evaluation among associated input agents.

FCM is a combination of Fuzzy Logic and Neural Network; it combines the heuristic and commonsense rules of Fuzzy Logic with the learning heuristics of Neural Networks. They were introduced by Kosko [24], who used fuzzy reasoning to enhance the cognitive maps that had been previously used in the field of socio-economic and political sciences to analyse social decision making problems. Kosko considered fuzzy values in the variables of cognitive maps and utilized them in order to represent causal reasoning. FCMs have been applied for many applications in different scientific fields [24]. While the prediction capability of FCMs can be useful in answering the *what-if* questions in a decision support environment, little research has been done on the use of FCMs in goal oriented analysis. Such analysis starts with a desired goal, and aims to identify what initial state can trigger a chain of events leading to the state corresponding to that goal.

The two significant characteristics of FCM [24] complying with the F-IAS agent model includes:

(a) Causal relationships between the input agents and the F-IAS agent can take on values representing fuzzy set memberships.
(b) The system is dynamic.

The *Causal Significance* module within the *Association Evaluation* of the F-IAS agent simulates the *what-if* conditions to evaluate the possible cause–effect relationship of associations by computing and analysing the effectiveness of each input agent in respect to the F-IAS agent [12]. The evaluation is based on a simple idea: the most effective and relevant associations do the best job of predicting the output for the F-IAS agent.

The steps involved to analyse the causal significance are described by the following steps [25]:

1. For each F-IAS agent input provided an input agent x_j, compute the fuzzy membership function value of K data points (x_j^k, y^k), $j = 1, \ldots, N$, in each $x_j^k - y$ space $k = 1, \ldots, K$. For each input provided by an input agent, the fuzzy membership function is computed using the method reported in [7] and the membership function is in the form of a Gaussian fuzzy set as follows:

$$\mu^k_{A^i_j}(x^k_j) = \exp\left(-\left(\frac{x^k_j - c_{A^i_j}}{\sigma_{A^i_j}}\right)^2\right), \quad k = 1, 2, 3, \ldots, K \tag{7}$$

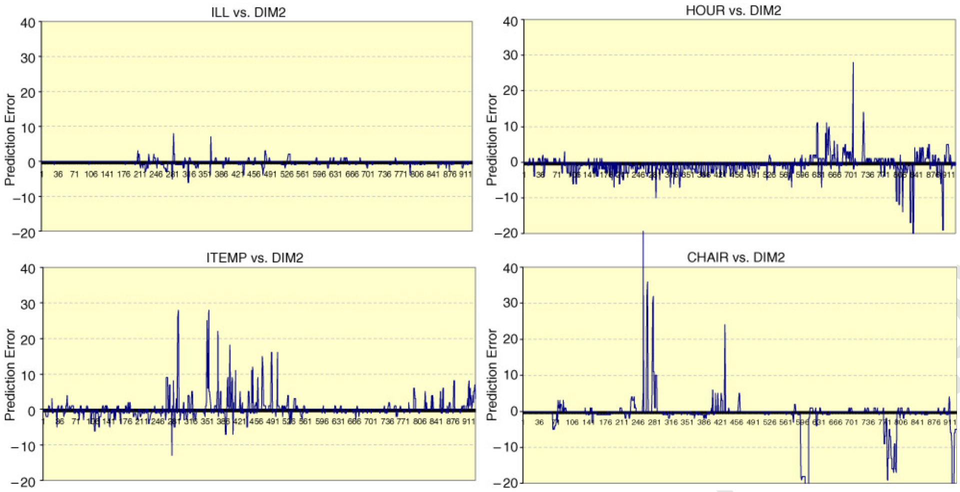


Fig. 15. Fuzzy Causal Associative Criteria for F-IAS (DIM2).

where A_j^i is the linguistic label of the fuzzy set, $c_{A_j^i}$ is the centre and $\sigma_{A_j^i}$ is the variance of the Gaussian membership function.

2. Defuzzify these K data points by applying the rulebase to the F-IAS agent to produce an output $\tilde{y}^k_{FCAC}$ for each input agent x_j [25] using

$$\tilde{y}^k_{FCAC}(x^k_j) = \frac{\sum_{i=1}^{L} \mu^k_{A^i_j}(x^k_j)\overline{B}^{i*}}{\sum_{i=1}^{L} \mu^k_{A^i_j}(x^k_j)}. \tag{8}$$

If an association to an input agent x_j is more important than an association to an input agent x_m, then the approximation $\tilde{y}^k_{FCAC}(x^k_j)$ will be closer to y^k than $\tilde{y}^k_{FCAC}(x^k_m)$. Fig. 15 shows the plotted $\tilde{y}^k_{FCAC}(x^k_j)$ for the input agents ILL, HOUR, ITEMP and CHAIR in respect to the F-IAS agent DIM2. It can be observed that the ILL results in less prediction errors in comparison to the other input agents. According to the plots, the significance of associations to given input agents can be ranked as follows: {ILL, HOUR, CHAIR, ITEMP}.

However, in order to enable the F-IAS agent to calculate the overall error as an indicative factor for causal significance the fuzzy causal association criteria are used and defined as follows [25]:

$$P\tilde{y}_{FCAC}(x_j) = \frac{1}{K \times v_y} \sum_{k=1}^{K} (\tilde{y}^k_{FCAC}(x^k_j) - y^k)^2 \tag{9}$$

where $v_y = (\sum_{i=1}^{K}(y_i - \bar{y})^2)/K$ is the variance of $y^1, y^2, \ldots, y^K$. The smaller $P\tilde{y}_{FCAC}(x_j)$ the more important the association to input agent x_j gets [25]. Hence, an ascending sorting of the performance criteria $P\tilde{y}_{FCAC}(x_j)$ gives a list of the input agent x_j in order of significance.

The fuzzy causal association criteria can be used to automatically and quickly extract significant information about the causal significance of associations to input agents. According to this information the associations can be evaluated and ranked within the previously obtained subset that was evaluated and ranked based on predictive significance. The possibility to further discard irrelevant input agent associations of this subset may be re-initiated by the *System Assessor* and *System Reviser*. However, this can and will only be performed if the following condition is satisfied: the elimination of the least causally significant associations can only be realised if, and only if, this wouldn't cause a major decrease of the F-IAS agent's model prediction and performance beyond the given threshold.

One of the great benefits of the fuzzy causal association criteria method is that it operates in an online fashion and is very fast compared to other methods. The F-IAS agent integrates this causal association criteria algorithm (evaluating casual significance) together with the predictive significance within the *Association Evaluation* module which evaluates the association's significance based on prediction and causality at the same time to avoid repeated computation and regeneration of the fuzzy models during the association combination search.

In the next section, we report on the experiments that were carried out to evaluate the performance of the proposed F-IAS agent.

8. Experimental results and comparisons

We conducted several experiments within the iDorm. In this paper we will present a subset of these experiments where the user in Fig. 16 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 behaviour within the iDorm and reduces the associations to other embedded input agents that even if a small number of associations are omitted the network overheads will decrease significantly which will lead to the decrease of the average response time and Time of Travel of the events leading to much improved performance. In addition, we will also show 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. In the next subsection, we will introduce the experimental setup.

8.1. The embedded agents setup

All the experiments presented in this subsection have been carried out in the iDorm using the following embedded agents:

- *Input or Passive Embedded 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)

Fig. 16. The user occupying the iDorm.

Table 1
Fuzzy properties of the agents

	# Fuzzy Sets	Fuzzy values
ILL	7	vvlow, vlow, low, med, high, vhigh, vvhigh
ELL	7	vvlow, vlow, low, med, high, vhigh, vvhigh
ITEMP	7	vvlow, vlow, low, med, high, vhigh, vvhigh
ETEMP	7	vvlow, vlow, low, med, high, vhigh, vvhigh
CHAIR	2	On, Off
BED	2	On, Off
HOUR	7	vvlow, vlow, low, med, high, vhigh, vvhigh
Ceiling Lamp 1	7	vvlow, vlow, low, med, high, vhigh, vvhigh
Ceiling Lamp 2	7	vvlow, vlow, low, med, high, vhigh, vvhigh
Ceiling Lamp 3	7	vvlow, vlow, low, med, high, vhigh, vvhigh
Ceiling Lamp 4	7	vvlow, vlow, low, med, high, vhigh, vvhigh
Desk Lamp	2	On, Off
Bed Lamp	2	On, Off

- *F-IAS Intelligent Embedded agents* which were embedded in the following actuating devices: desk lamp (DESKLAMP), bed lamp (BEDLAMP), 4 independent ceiling lamps (DIM1, DIM2, DIM3, DIM4).

The input agents operate as UPnP devices running on limited computational platforms Every F-IAS agent has been implemented as an UPnP Control Point running autonomously on embedded devices.

The input and F-IAS agents are fuzzy-logic based, where the membership functions of the inputs and outputs of the various embedded agents were obtained using the method reported in [7]: the number of the fuzzy sets and the linguistic labels are listed in Table 1.

8.2. The initial associations setup

The user occupied the iDorm for 5 consecutive days during office hours to use the room as an office (Fig. 16). After a short introduction and teaching period, the user was asked to use the IIE Editor which was running on the iFridge to create desired and intended associations as well as controlling the devices that are located within the iDorm. The association that the user thought would be always applicable included the connection of the chair to the DESKLAMP and DIM1 by assigning the following association rules: (1) IF CHAIR is occupied THEN DESKLAMP is ON; (2) IF CHAIR is unoccupied THEN DESKLAMP is OFF and (3) If CHAIR is *occupied* THEN DIM1 is OFF and (4) IF CHAIR is *unoccupied* THEN DIM1 is OFF. Since it wasn't exactly obvious and predictable for the user how and when he will use all the agents for what purpose, the user decided to put all the agents into one society called 'study/office' indicating that the information of those agents should be regarded as potentially important for any F-IAS agents running in the same society. Since the number of agents of the society was not so high, the F-IAS agent established associations to all of the other agents within the same society. Once finished creating the society, the user activated the F-IAS agents. The user then was asked to interact with the iDorm by controlling the agents as preferred during the first 3 days. After the user F-IAS agent has learnt the user's behaviour and optimized the associations to the input agent, the F-IAS agent was then operated in the online actuation and adaptation mode for the remaining 2 days where the agent showed a very satisfactory performance to the user.

8.3. Learning associations for an F-IAS agent

In this section we first explain and analyse the obtained experimental results based on one F-IAS agent, the *ceiling lamp 1* (DIM1), with the aim to present a better understanding of the functionality of the system.

The tasks of the F-IAS agent involves monitoring the user interaction with the agents of the society for a certain time, keeping a record of the events that occurred and learn his particularised behaviour and thus extract and generate the appropriate controller as well as reducing the set of associations to the most relevant and important input agents. Any change in the society (e.g. whenever a new agent enters or/and an existing one leaves) activates the F-IAS to start new evaluations and analysis where the Intelligent Association Selection of the F-IAS works in a responsive and lifelong fashion. As a result, the agent continuously adjust to changes and evaluates associations to enable dissolving the ones that become less important over time and also allow new important associations to be included. A flowchart of the overall operational steps of the F-IAS within the iDorm is shown in Fig. 17.

The F-IAS agent monitored the user's interactions with the environment and the associated agents of the society and recorded the events in the *Event Database* of the F-IAS agent. After a minimum period of data collection (three consecutive days in the case of our experiments), the F-IAS agent activates the transition to the online actuation and adaptation mode (which lasted for 2 consecutive days in the case of our experiments).

The dataset obtained from the 'study/office' society during the data collection mode comprises of 800 instances. The nature of the agents within the society produced also many fluctuating and noisy data leading to a high number of events although the number of input agents may be regarded as small. Even with this small amount of agents we had a large increase of network delays and processing time for the F-IAS agent as listed in Fig. 18.

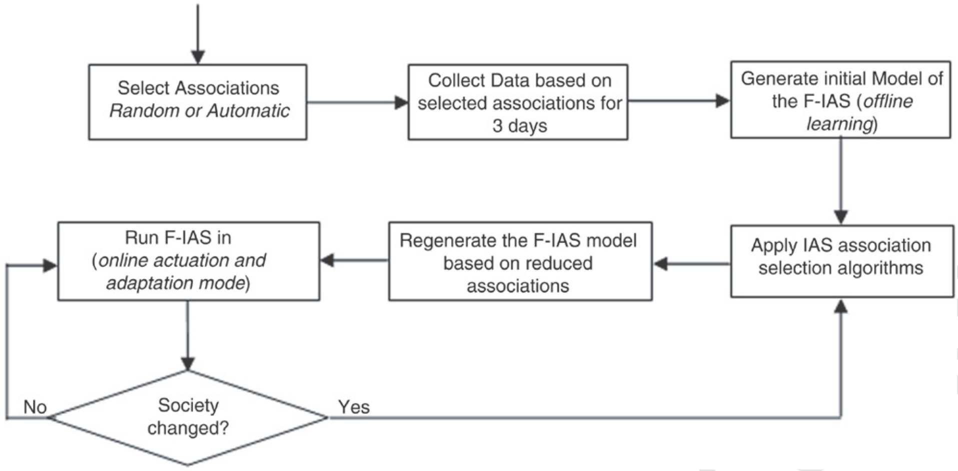


Fig. 17. Operational flowchart of an F-IAS agent in the iDorm.

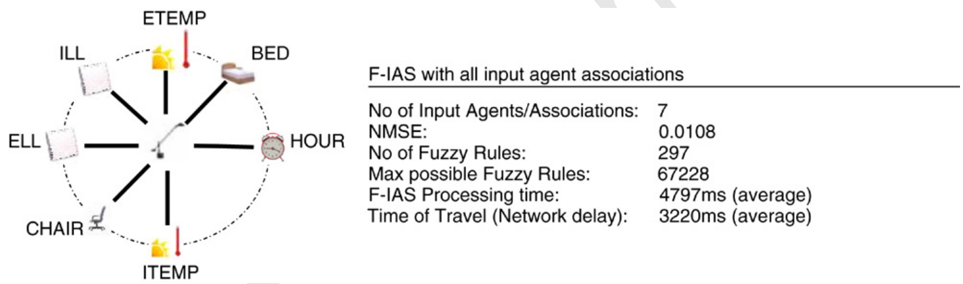


Fig. 18. F-IAS with all associations.

The collected data from the 'study/office' society of the iDorm were divided into two equal data sets: the *training* and *testing data sets*, each containing 400 instances. The training data set was used to generate an initial model by applying the rule induction method of the F-IAS as described in Section 6. Simultaneously and online the F-IAS agent analysed and evaluated the associations to every single input embedded agent based on these data. The performance measurement for each association is obtained based on the NSME obtained over the testing instances. During rule induction, the F-IAS generated a set of 297 fuzzy rules (out of 67 228 possible rules) from the 400 training instances in a very short time with a processing time of 4797 ms on average to go through a single control cycle as indicated in Fig. 18. The prediction NMSE of 0.0108 was calculated using the testing data sets which showed a good accuracy of the learnt model. This prediction accuracy of the F-IAS agent associating to all input agents within the society of 'study/office' forms the indicative comparison for the next step of the system which will use the IAS Association Selection based on association combination subset search, predictive significance as well as association causal significance.

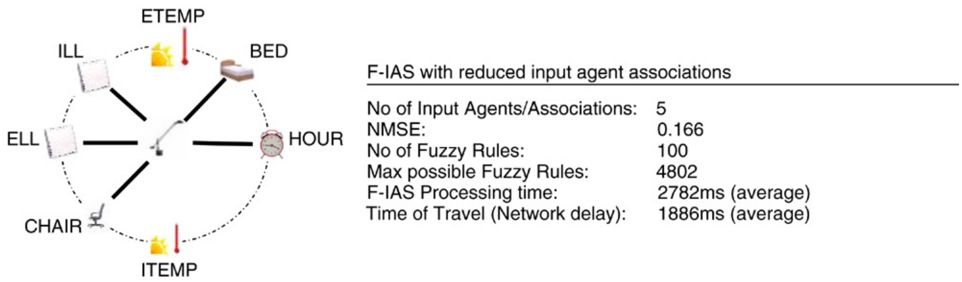


Fig. 19. F-IAS after applying intelligent association selection.

Table 2
Input agent ranking based on FCAC

	Ranking	FCAC
ILL	1	0.001619
ELL	3	0.032655
CHAIR	4	0.061655
BED	5	0.116068
HOUR	2	0.019865

The F-IAS agent starts a routine to search through possible input agent association combinations as described in Section 7.3, where the stepwise search efficiently reduces the number of possible association subset evaluations from 127 to 28. After a twelve seconds search through possible combinations, the best subset of association combinations based on predictive significance includes {ILL, ELL, CHAIR, BED, HOUR}. The F-IAS agent generates a new model based on the selected input agents and starts a new evaluation based on the causal significance of each input agent association. After applying the Fuzzy Causal Association Criteria method, we obtain the ranking table shown in Table 2.

Since further elimination of associations results in a major decrease (outside the given threshold of 15%) of the F-IAS agent's model prediction and performance, the selected subset of associations remains as selected before. With this reduced set of association, the number of rules generated was 100 with a prediction error of 0.166. Although the prediction accuracy dropped, the error is still within an accepted range and the performance drop will be hardly noticeable for the user. The system will be able to relearn and adjust the rules that it has 'forgotten' during the *online learning and adaptation mode* in a very short time interval. Fig. 19 shows the F-IAS after omitting the ITEMP and ETEMP. With 5 remaining associations, the F-IAS reduced its processing time by almost 42% to 2.782 s on average. In addition, the Time of Travel of network events multicast by the agents dropped from 3220 ms to 1886 ms on average as shown in Fig. 19 which is a reduction of about 41%.

8.4. Results of all F-IAS agents

The results of the intelligent association selection methods of simultaneously running F-IAS agents within the iDorm are listed in Table 3. The first column *NMSE (all)* of the

Table 3
Association selection results for all F-IAS agents

	NMSE (all)	ILL	ELL	ITEMP	ETEMP	CHAIR	BED	HOUR	*# Rules*	*NMSE*	*Time of travel* (ms)	*Processing* (ms)
DM1	0.0108	X	X	–	–	X	X	X	100	0.0108	1886	2782
DM2	0.10430273	X	X	X	–	–	X	X	186	0.12475	1562	3734
DM3	0.05415031	X	X	–	–	–	X	X	61	0.1357	983	2098
DM4	0.082062826	X	X	X	–	X	X	X	168	0.18129	1983	3217
DESKLAMP	0.051863287	X	X	X	–	–	–	–	31	0.04894	729	1203
BEDLAMP	0.014872335	X	X	X	–	–	X	–	39	0.04698	758	1327

table indicates the overall model prediction error for the fully associated F-IAS agents to the available input agents. The listed F-IAS agents include (DIM1, DIM2, DIM3, DIM4, DESKLAMP and BEDLAMP). 'X' describes the availability of an association set from the F-IAS to the input agent mentioned in the header of the column while '–' is an indication of no association. The initial fuzzy model of every individual F-IAS agent contained 297 rules with an average processing time of 4797 ms and average time of travel of 3220 ms as mentioned above.

After the association selection and evaluation period, it can be seen that the rulebase of all F-IAS agents reduced drastically, e.g. 90% in the case of DESKLAMP, after omitting learnt irrelevant associations while the overall system remains as accurate as the initial model with all associations. In addition, the processing time and network overhead reduces considerably with the removal of every single association leading to a smaller number of fuzzy rules for the F-IAS agent to process. A notable example for this efficiency can be seen in the case of the DESKLAMP where from 297 rules initially generated by the F-IAS agent, the system successfully reduced this number to only 31 rules by dissolving 4 associations that it noticed to be irrelevant and redundant. Without doubt, through the reduction of the rules, some good rules might be deleted; however these rules can be re-learnt in a very short time again during the online actuation and adaptation mode.

8.5. Comparison of the F-IAS with the fuzzy set variations method

As the F-IAS agent's model accuracy is also dependent on the design of the fuzzy controller which is consequently dependent on the definition of fuzzy membership functions, so we have also evaluated a method for association ranking based on the input agent's fuzzy membership function manipulation as described in [31]. The main idea of this approach comprises of the following observation: if the F-IAS agent's output is sensitive to an input agent, then by defining more fuzzy sets to cover this association, a more accurate fuzzy prediction model would be obtained [31]. In contrast, if the F-IAS agent's output is not sensitive to the input agents, then the prediction accuracy will not change much when more fuzzy sets are defined. Note that this approach is similar to the F-IAS causal importance in respect to cause–effect relationship evaluation. The main difference, however, is that the F-IAS uses fuzzy sets obtained by the techniques provided by [7] and they are not changed during the lifetime of the system, whereas in [31] fuzzy sets are manually manipulated where the number of fuzzy sets is either increased or decreased. The compared algorithm includes the following four steps:

- Define for each associated input agents of the F-IAS agent the initial seven fuzzy sets.
- First decrease the number of fuzzy sets defined for *one* input agent to 3; then increase to 5 while keeping the remaining input agents' membership functions unchanged.
- Compute the NMSE of over the whole training data.
- Repeat these steps for all of the input agents and the associations will be ranked according to the NMSE as shown in Table 4.

Once applied, in general we obtain a different ranking compared to the F-IAS for DIM1. A membership function composed of 5 fuzzy sets suggests that {HOUR} is the most influential input agent followed by {ILL}. An evaluation of association and input

Table 4
Comparison of association rankings based on the fuzzy set manipulation method suggested in [31]

No fuzzy sets	ILL	ELL	ITEMP	ETEMP	HOUR	
7	No Rules: 297		NMSE: 0.0108			
3	252	268	201	254	255	*No Rules*
	0.044	0.09	0.088	0.054	0.058	*NMSE*
	5	1	2	4	3	*Ranking*
5	287	282	251	272	282	*No Rules*
	0.016	0.015	0.015	0.013	0.017	*NMSE*
	2	3	4	5	1	*Ranking*
F-IAS (All asssociations)	No Rules: 297		NMSE: 0.0108			
	2	1	5	4	3	

agent significance only based on this method can be misleading and insufficient since the different ranking of the input agents doesn't specifically describe the overall significance of an input agent. This can be easily observed from Table 4 where the ranking of the input agents depending on the assigned number of fuzzy sets changes and makes it very difficult to decide which of them really imply. Furthermore, a main drawback of this method is that it can be only applied to continuous inputs in the form of fuzzy sets but not to binary discrete values (e.g. like the bed pressure sensor we use in the iDorm) [31].

8.6. Comparison with other soft computing techniques

We compared our association selection method with two other soft computing based techniques: the Neural Networks (NN) and Genetic Algorithms (GA). The selected techniques are well-established methods in machine learning and modelling with the characteristics of supporting input variable selection [13,30]. For the experiments conducted with the NN and GA the aim was to find the best possible model with the least NMSE. The training data of 400 instances that are provided to the NN and GA is the same that is used for the DIM1 F-IAS rule induction.

The NN used consisted of a *Linear Network* having direct connections between input layer and output layer. The network was implemented using the Easy-NN tool [13] and was trained using back-propagation with a learning rate set to 0.2 [13]. The obtained NMSE is 0.101533 for the DIM1 as shown in Table 5. The importance of associations between the input and output agents is calculated by the sum of the connection weights of each input. The relative importance shows how much an associated input agent affects the NMSE compared to others. As a result, the sensitivity analysis ranks the importance of the input agents as follows: {ELL, HOUR, ETEMP, CHAIR, BED, ITEMP, ILL}.

The GA used a population of 100 individuals evolving over 100 generations with a mutation rate of 0.1 and crossover rate of 1. We used Trajan [30] which is a tool with an integrated feature of selecting input variables based on GA. After an exhaustive search through populations, the GA suggests limiting the input agents, in no specific order, to {ILL, ELL, ETEMP, BED, CHAIR}. The computed NMSE of the model generated from the testing data sets is specified as 0.0371332, which, in comparison to the NN,

Table 5
Comparison of NMSE and association rankings for NN, GA and F-IAS

		ILL	ELL	ITEMP	ETEMP	CHAIR	BED	HOUR	NMSE
NN	Weights	0.1045	−0.7241	0.0362	0.0428	−0.0416	−0.1129	0.0835	0.101533
	Sensitivity analysis	1.00036	2.17845	1.00133	1.02286	1.01828	1.01695	1.03608	
	Ranking	7	1	6	3	4	5	2	
GA		ILL	ELL	ITEMP	ETEMP	CHAIR	BED	HOUR	NMSE
	Sensitivity analysis	X	X	–	X	X	–	X	0.0371332
F-IAS		ILL	ELL	ITEMP	ETEMP	CHAIR	BED	HOUR	NMSE
	*FCAC	0.753	0.893	0.2120	0.2633	0.675	0.41	0.283	0.0108
	Ranking	2	1	7	6	3	4	5	

*Fuzzy Causal Association Criteria.

promises a better accuracy and performance. The NN, on the other hand, would regard the {ILL, ITEMP} as the least important agent associations. Consequently, a retraining of the network without these two input agents decreases the NMSE to 0.92763. Both the NN and GA, however, when compared to F-IAS perform worse. The iterative nature of the NN and GA makes them also highly computationally intensive. F-IAS requires far less computation due to the one-pass rule induction procedure and thus it is more suitable for embedded agents. In addition, the GA and NN cannot be easily applied online as this would require them to repeat their time-consuming relearning cycles if either a new association was to be added or existing ones removed.

Other methods, like statistical approaches in the form of correlations analysis (e.g. Pearson correlation), principal component analysis (PCA), and analysis of variance (ANOVA) are well known tools for correlation measurements but mostly based on linear regression modelling. However, in the iDorm, more generally in IIEs, the agent models are almost always nonlinear, where these kinds of tools would fail to discover the significant associated input agents and thus are not used in the comparison.

9. Conclusions

One of the Grand Challenges for Computing Research within the next century as specified by the UK Engineering and Physical Sciences Research Council (EPSRC) is to develop intelligent adaptive configuration tools and systems that are needed to support scalable smart environments. The multitude of interconnected devices and artefacts in such environments results in major network and processing delays as well as resulting in complexities in programming and configuring smart environments to personalise themselves to suit the user's individual needs. This work aims to be a step towards addressing these challenges.

In this paper we presented a novel intelligent embedded agent technique for reducing the number of associations and interconnections between the various agents operating within the IIE in order to minimise the network and processing overheads whilst reducing the cognitive load of programming these associations to personalise themselves to the user needs within IIEs. We started by defining 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 intentions. Our intelligent embedded agents and techniques were evaluated in the Essex intelligent Dormitory (iDorm) which forms the experimental test bed for IIE research at the University of Essex. The main goals of the proposed fuzzy based F-IAS agents operating within the iDorm included learning and adapting to the user behaviour in a lifelong non-intrusive mode to control the environment on his behalf. In addition, the F-IAS agent aimed at reducing the agent associations and interconnections to the most relevant set in order to reduce the network and processing overheads and thus improving the system overall efficiency.

We carried out unique experiments in which a user stayed in the iDorm for five consecutive days. The intelligent lifelong learning and adaptation of the F-IAS agent occurred in a non-intrusive manner while the user carried out his normal activities in the environment. After the data collection phase, the F-IAS agent had learnt the particularised user behaviours using a simple one-pass method which is computationally non-demanding. We have also demonstrated a novel feature of the F-IAS agent in that it selects the most relevant and important associations to other agents within the iDorm. This selection was obtained based on the following: (1) predictive significance which searches through association subsets and evaluates the accuracy of the model prediction and (2) causal significance which evaluates and ranks the associations based on an individual cause–effect analysis. During our experiments, we have shown that the F-IAS agents were able to recognise redundant and irrelevant associations and dissolve them while keeping an acceptable system performance. As a result, during the experiments the F-IAS agent managed to decrease its processing time by up to 75% as well as reducing the number of stored rules by up to 90% and reducing the Time of Travel and thus the network delay by up to 63%.

The proposed F-IAS agent with its novel intelligent association selection method was compared also with other soft-computing based approaches; namely GA, NN and Fuzzy Sets Variations. The results showed that the optimum performance of the F-IAS agent produced on average a lower NMSE than both GA and NN and was computationally less intensive and better suited for online learning than the other compared approaches. In addition, the results obtained by the compared methods were misleading and unreliable.

For our current and future work, we intend to perform more and longer experiments with multiple users and larger numbers of agents.

Acknowledgements

We are pleased to acknowledge the funding support from the EU IST Disappearing Computer program and the joint UK–Korean Scientific Fund. We would also like to thank Anthony Pounds-Cornish, Faiyaz Doctor, Graham Clarke, Martin Colley, and Jeanette Chin for their indirect contributions arising from many stimulating discussions on intelligent embedded agents and IIE issues. We also would like to thank the anonymous reviewers for their constructive comments which helped to improve the paper.

References

[1] M. Bielikova, T. Krajcovic, Ambient intelligence within a home environment, ERCIM News 47 (1991) 12–13.
[2] B. Brumitt, B. Meyers, J. Krumm, A. Kern, S. Shafer, EasyLiving: Technologies for intelligent environments, in: Proceedings of the HUC, 2000, pp. 12–29.
[3] V. Callaghan, G. Clarke, M. Colley, H. Hagras, J. Chin, F. Doctor, Inhabited intelligent environments, BT Technology Journal 22 (3) (2004) 233–247.
[4] S. Chiu, Selecting input variables for fuzzy models, Journal of Intelligent and Fuzzy Systems 4 (4) (1996) 243–256.
[5] CISCO — The Internet Home, http://www.cisco.com/uk/ihome/.
[6] M. Coen, Design principles for intelligent environments, Proceedings of the AAAI/IAAI (1998) 547–554.
[7] F. Doctor, H. Hagras, V. Callaghan, 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) (2005) 55–65.
[8] F. Doctor, H. Hagras, V. Callaghan, A type-2 fuzzy embedded agent to realise ambient intelligence in ubiquitous computing environments, International Journal of Information Science 171 (4) (2005) 209–334.
[9] H. Duman, H. Hu, United we stand, divided we fall — team formation for multiple mobile robot applications, International Journal of Robotics and Automation 16 (2001).
[10] H. Duman, V. Callaghan, H. Hagras, A soft-computing based approach to intelligent association in agent-based ambient-intelligence environments, in: Proceedings of the 4th International Conference on Recent Advances in Soft Computing 2002, RASC ’02, Nottingham, UK, April 2002, pp. 387–393.
[11] H. Duman, H. Hagras, V. Callaghan, G. Clarke, M. Colley, Intelligent association in agents based ubiquitous computing environments, in: Proceedings of the International Conference on Control, Automation, and Systems, Muju, Korea, October 2002.
[12] H. Duman, Z. Bien, H. Hagras, V. Callaghan, G. Clarke, M. Colley, 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.
[13] EasyNN-Plus, Neural Network Planner Software, http://www.easynn.com, 2002.
[14] A. Gaweda, J. Zurada, R. Setiono, Input selection in data-driven fuzzy modeling, in: Proceedings of the 10 International Conference on Fuzzy Systems, FUZZ-IEEE ’01, Melbourne, Australia, December 2001.
[15] S. Guillaume, Designing fuzzy inference systems from data: An interpretability-oriented review, IEEE Transactions on Fuzzy Systems 9 (3) (2001) 426–443.
[16] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[17] H. Hagras, V. Callaghan, M. Colley, G. Clarke, H. Duman, Online learning and adaptation for intelligent embedded agents operating in domestic environments, in: Fusion of Soft Computing and Hard Computing for Autonomous Robotic Systems, in: Studies in Fuzziness and Soft Computing Series, Physica-Verlag, 2002.
[18] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, H. Duman, Creating an ambient-intelligence environment using embedded agents, IEEE Intelligent Systems Magazine 19 (6) (2004) 12–19.
[19] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, E. Jansen, The gator tech smart house: A programmable pervasive space, IEEE Computer Magazine (2005) 50–60.
[20] M. Hellenschmidt, T. Kirste, Self-organization for multi-component multi-media environments, in: Proceedings of UbiComp ’04, Nottingham, UK, September 2004.
[21] T. Hoare, R. Millner, Grand Challenges in Computing Research, British Computer Society, http://www.ukcrc.org.uk/grand_challenges/index.cfm, 2004.
[22] A. Holmes, H. Duman, A. Pounds-Cornish, The iDorm: Gateway to Heterogeneous Networking Environments, in: Proceedings of the International ITEA Workshop on Virtual Home Environments, Paderborn, Germany, February 2002.
[23] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, The aware home: A living laboratory for ubiquitous computing research, in: Proceedings of the 2nd International Workshop on Cooperative Buildings, CoBuild ’99, 1999.
[24] B. Kosko, Neural Network and Fuzzy Systems, Prentice-Hall, ISBN: 0-13-611435-0, 1992.

[25] Y. Lin, G. Cunningham, S. Coggeshall, R. Jones, Nonlinear system input structure identification: Two stage fuzzy curves and surfaces, IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans 28 (5) (1998).
[26] N. Minar, M. Gray, O. Roup, R. Krikorian, P. Maes, HIVE: Distributed agents for Networking Things, ASA/MA, 1999.
[27] MIT Oxygen Project: An Overview, http://oxygen.lc s.mit.edu.
[28] M. Mozer, The neural network house: An environment that adapts to its inhabitants, Proceedings of the AAAI (1999).
[29] M. Mozer, An intelligent environment must be adaptive, IEEE Intelligent Systems Magazine (1999).
[30] Trajan Neural Network Software, http://www.trajan-software.demon.co.uk/.
[31] L.-X. Wang, The WM method completed: A flexible fuzzy system approach to data mining, IEEE Transactions on Fuzzy Systems 11 (6) (2003) 768–782.
[32] M. Youngblood, L. Holder, D. Cook, Managing adaptive versatile environments, in: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, PERCOM'05, 2005, pp. 351–360.
[33] R. Want, T. Perin, G. Borriello, K. Farkas, Disappearing hardware, IEEE Pervasive Computing 1 (1) (2002) 36–47.
[34] M. Weiser, Some computer science problems in ubiquitous computing, Communications of the ACM (1993).

Hakan Duman is a teaching assistant of the Department of Computer Science at the University of Essex. He received his Diploma (M.Sc) in Computer Science from the University of Applied Sciences in Regensburg and is currently submitting his Ph.D. thesis entitled 'On Investigating Intelligent Associations in Intelligent Inhabited Environments and Appliances'. His research interests include ambient intelligence, embedded agent-based and serviced-oriented computing in ubiquitous computing environments. He is also strongly interested in the areas of computational intelligence, intelligent control, consumer electronics, and robotics. Hakan has served as a program committee member for several conference and workshops, and has been a reviewer for several international journals and magazines. He is a member of the IEEE Computer and Computational Intelligence Society, International Institute of Informatics and Systemics (IIIS), and the Institute of Engineering and Technology (IET).

Hani Hagras is a Senior Lecturer in Computer Science in the University of Essex. His major research interests are in the areas of Computational Intelligence, namely Fuzzy Logic, Neural Networks, Genetic Algorithms and Evolutionary Computation. His research interests also include Ambient Intelligence, Ubiquitous Computing, Disappearing Computing, Invisible Computing and Intelligent-Buildings. He is also strongly interested in the areas of Embedded-agents, Robotics and intelligent control of industrial processes. He has authored more than 90 papers in international journals, conferences and books. He is a Senior Member IEEE and he is also the co-chair of the IEEE International Task Force on Extensions to Type-1 Fuzzy Sets. He is a member of the IEEE Computational Intelligence Society Fuzzy Systems Technical Committee, and is also a member of the IEEE IndustrialElectronicsSociety Technical Committee of the Building Automation, Control and Management. In addition, he is a member of the Executive Committee of the IEE Robotics and Mechatronics Professional Network. He is associate editor of the International Journal of Robotics and Automation; he is also on the editorial board of the International Journal of Computational Intelligence. He is the General co-chair for the 2007 IEEE International Conference on Fuzzy systems, London, UK; he is also the Program Chair for the 2006 IEE International Conference on Intelligent Environments, Athens, Greece, June 2006 as well as the Co-Chair of the organizing committee for the 2nd International Symposium on Evolutionary Fuzzy Systems, Lake District, UK, September 2006. He served as the Program Co-chair for the 10th IASTED International Conference on Robotics and Applications, Hawaii, USA, August 2004. He was also the General Chair for the IEE International Workshop on Intelligent Environments, Colchester, UK, June 2005

Vic Callaghan is professor of computer science, director of the Digital Lifestyles Centre and leader of the Inhabited Intelligent Environments group at Essex University. He is a graduate of the University of Sheffield where he gained both a B.Eng and Ph.D. in electronics and computing, respectively. His main expertise concerns embedded computing, especially applied to the creation of digital homes, intelligent buildings and other pervasive computing environments. He has authored approximately 100 journal papers, conferences and books on these topics and holds 2 key patents on embedded agents and end-user programming techniques. He is on the organizational team of major pervasive computing conferences (e.g. Pervasive Computing & Applications, Ubiquitous Intelligence & Computing, and Intelligent Environments) and journals (e.g. Pervasive Computing & Communications, Ubiquitous Computing & Intelligence).