

A Type-2 Fuzzy Embedded Agent To Realise Ambient Intelligence In Ubiquitous Computing Environments

Faiyaz Doctor, Hani Hagra, Victor Callghan

Department of Computer Science, University of Essex, Wivenhoe Park,
Colchester, CO4 3SQ, UK
Email: fdocto@essex.ac.uk.

Index Terms--- Type-2 Fuzzy Logic Systems, Type-2 Fuzzy Logic Controllers, Learning, Ambient Intelligence, Embedded Agents, Ubiquitous computing environments.

Abstract

In this paper, we present a novel approach for realising the vision of ambient intelligence in Ubiquitous Computing Environments (UCEs). This approach is based on embedding intelligent agents in UCEs. These agents are based on type-2 fuzzy systems which are able to handle the different sources uncertainty and imprecision in UCEs to give a good response. We have developed a novel system for learning and adapting the type-2 fuzzy agents so that they can realise the vision of ambient intelligence by providing a seamless, unobtrusive, adaptive and responsive intelligence in the environment that supports the activities of the user. The user's behaviours and preferences for controlling the UCE are learnt online in a non intrusive and life long learning mode so as to control the UCE on the user's behalf. We have performed unique experiments in which the type-2 intelligent agent has learnt and adapted online to the user's behaviour during a stay of five days in the intelligent Dormitory (iDorm) which is a real UCE test bed. We will show how our type-2 agents will realise the vision of ambient intelligence and deal with the uncertainty and imprecision present in UCEs to give a very good response that outperforms the type-1 fuzzy agents while generating a smaller number of rules.

1. Introduction

Ubiquitous computing, also referred to as pervasive computing, is a paradigm in which computing technology becomes virtually invisible as a result of being embedded into our everyday environment. Ubiquitous Computing Environments (UCEs) contain networked embedded computer artefacts that can interact with the users living or working within them. The challenge however is how to manage and configure the computer-based artefacts and systems present in these ubiquitous environments in a seamless and non-intrusive way; without the user being cognitively overloaded by having to manually configure these devices to achieve a desired functionality. The vision of *Ambient Intelligence* was introduced to address this challenge [5, 17]. In this vision people are

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, *Journal of Information Sciences*, Vol. 171, No. 4, pp. 309-334, May 2005

empowered through a digital environment that is “aware” of their presence and context, and is *sensitive, adaptive and responsive* to their needs [6].

Ambient intelligence improves the quality of life by creating the desired environmental conditions and functionality via intelligent, personalised interconnected systems and services. Ambient intelligence environments are characterised by their ubiquity, transparency and intelligence [6]. *Ubiquitous* because the user is surrounded by a multitude of inter-connected embedded systems; *transparent* because the computing equipment appears invisible to the user as it is seamlessly integrated into the background; *intelligent* because the system can recognise the people that live in these environments and is able to program itself to meet their needs by learning from their behaviour [6].

There have been several research projects concerned with designing systems for realising ambient intelligence. Context aware systems with ubiquitous sensing capabilities are the focus of the Aware Home work at Georgia Tech [1]. MIT’s Oxygen project [7] creates pervasive human-centred computing through creating intelligent spaces in which embedded devices provide large amounts of computation and interfaces to cameras and microphones; allowing users to communicate through speech, vision and gesture recognition. In addition, networks with dynamic changing configurations are used, supporting handheld devices providing mobile access points within the environment. The Intelligent Room [3] is a related project that has applied similar concepts found in Oxygen to a room environment making it responsive to the occupant by adding intelligent sensors to the user interfaces. Philips Research first initiative on ambient intelligence has taken the form of World Wide Information Communication and Entertainment (WWICE) [4] a prototype system supporting personalised functions in a networked home environment. Its main features include interoperability, speech access and the ability to pass personal information around the environment with the user. These projects represent a large body of current research effort; however they are mostly concerned with time independent context, sensing and user interactions rather than temporal history, learning and adaptation which are central to our requirements for agents supporting the vision of ambient intelligence [6].

One approach to achieve the vision of ambient intelligence is to embed intelligent agents in the user environments so that they can control them according to the needs and preferences of the user [9]. Embedded intelligence is the inclusion of some capacity for reasoning, planning and learning in an artefact. Embedded-computers that contain this kind of intelligent capacity are normally referred to as “embedded-agents” [2]. Each embedded agent is an autonomous entity, and it is common for such embedded-agents (as intrinsic parts of

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, Journal of Information Sciences, Vol. 171, No. 4, pp. 309-334, May 2005

intelligent artefacts”) to have network connections allowing them to communicate and cooperate with other embedded agents, as part of a multi embedded agent system [2].

The dynamics of a UCE means that any embedded agent control system will have to deal with the huge amount of uncertainties which exist in a UCE. The sources for these uncertainties can be as follows:

- Uncertainties in the agent’s controller inputs as the sensors measurements are noisy, imprecise and are affected by the environmental conditions such as variations in light level due to cloud cover or temperature change due to the effects of wind changing currents, ..etc.
- Uncertainties in the agent’s controller outputs due to the change of actuators characteristics with the changing environmental conditions. For example there would be a difference between low light level on a bright sunny afternoon in late summer and low light level on a dim overcast afternoon in mid winter.
- Uncertainties due to change of environmental factors (such as the external light level, temperature, time of day (morning, evening...etc)) over a considerable long period of time due to seasonal variations.
- The main cause of uncertainty is humans occupying these environments as their behaviours and moods are dynamic, unpredictable and non-deterministic and change with time and season. There is also the fact that different words mean different things to different people and a term such as ‘warm’ in reference to temperature could mean one range of values to one person though possibly a very different range of values to someone else and this can vary by the variation of season.

The Fuzzy Logic Controller (FLC) has been credited with being an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance when contending with the uncertainty and imprecision attributed to the real world. 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 also 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. FLCs exhibit robustness with regard to noise and variation of system parameters. This is due to their ability in dealing with vague and incomplete information, however most FLC applications use the traditional type-1 FLCs.

Type-1 FLCs have the common problem that they cannot handle or accommodate for the uncertainties as they use precise type-1 fuzzy sets. Type-1 fuzzy sets handles the uncertainties associated with the inputs and outputs by using *precise and crisp* membership functions that the user believes capture the uncertainties [14]. Once the type-1 membership functions have been chosen, all the uncertainty disappears, because type-1 membership functions are totally precise [15].

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, *Journal of Information Sciences*, Vol. 171, No. 4, pp. 309-334, May 2005

A type-2 fuzzy set is characterized by a fuzzy membership function, i.e. the membership value (or membership grade) for each element of this set is a fuzzy set in $[0,1]$, unlike a type-1 fuzzy set where the membership grade is a crisp number in $[0,1]$ [14]. The membership functions of type-2 fuzzy sets are three dimensional and include a footprint of uncertainty, it is the new third-dimension of type-2 fuzzy sets and the Footprint Of Uncertainty (FOU) that provide additional degrees of freedom that can make it possible to directly model and handle uncertainties [14, 15]. Therefore FLCs that use type-2 fuzzy sets to represent the inputs and outputs of the FLC can handle the uncertainties facing our embedded agents in UCEs to produce a good performance. Moreover, using type-2 fuzzy sets to represent the FLC inputs and outputs will result in the reduction of the FLC rule base when compared to using type-1 fuzzy sets. This is because type-2 fuzzy sets rely on uncertainty represented in the footprint of uncertainty to cover the same range as type-1 fuzzy sets with much smaller number of labels [14].

In this paper, we present a novel system for learning and adapting type-2 fuzzy controllers for agents that can be embedded in UCEs. The intelligent learning mechanism learns the particularised needs of the user and adjusts the agent controller based on a wide range of parameters in a non-intrusive and invisible way. It is also able to adapt online to changing conditions and user preferences in a life-long learning mode. We will demonstrate that our type-2 fuzzy controller is able to handle the effects of uncertainties that become inherent as environmental conditions and user behaviours change over a long period of time. Our technique is a one pass method which is not computationally intensive and is therefore suitable for embedded computers which have limited computing abilities. We will present unique experiments in which the type-2 agent has learnt and adapted to the user behaviour during a total stay of five days in the intelligent Dormitory (iDorm) which is a real UCE test bed.

In Section 2, we will describe the iDorm which is our test bed for UCE. In Section 3, type-2 fuzzy sets are introduced. Our learning and adaptation technique for the type-2 agent is described in section 4. In Section 5, we present our experiments and results. Finally conclusions and future work are presented in Section 6.

2. The iDorm

The intelligent Dormitory (iDorm) which is shown in Figure (1) is a real UCE test bed comprising of a large number of embedded sensors, actuators, processors and heterogeneous networks in a student bedroom environment. The iDorm is a multi-user space which contains areas of different activities such as sleeping, working and entertaining [9]. It includes the normal mix of furniture, found in a typical student study/bedroom environment, including a bed, work desk and a wardrobe. The iDorm is fitted with a liberal placement of sensors

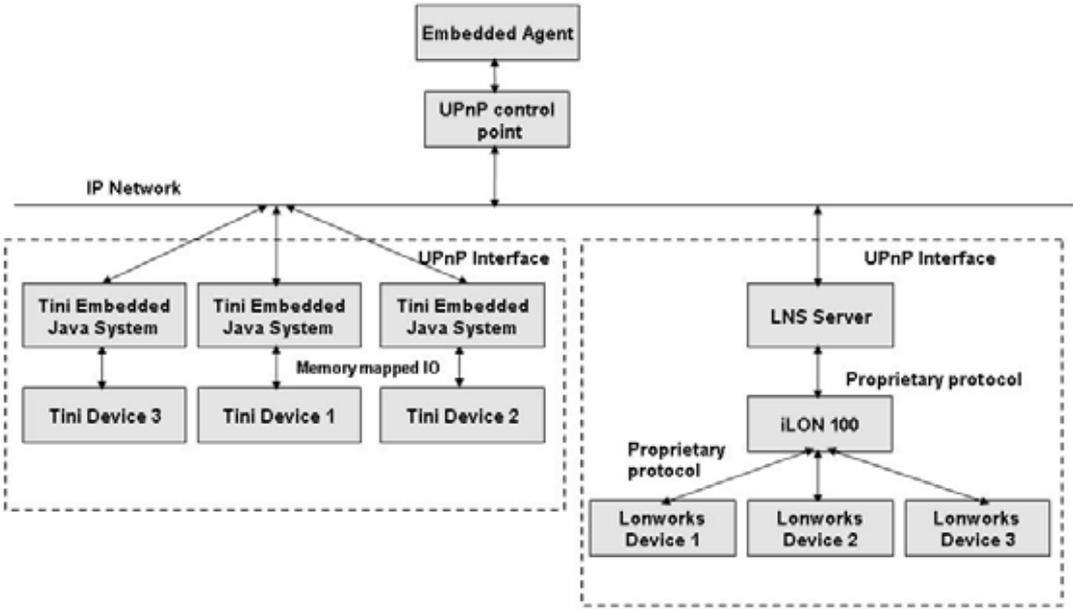
“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, Journal of Information Sciences, Vol. 171, No. 4, pp. 309-334, May 2005

and actuators. The sensors and actuators in the room are concealed (e.g. buried in walls) with the intention that the user should be completely unaware of the intelligent infrastructure of the room which is required by the ambient intelligence vision [6].



Figure (1): The iDorm

The iDorm is based around three networks, Lonworks, 1-wire (TINI) and IP which provide a diverse infrastructure allowing the development of network independent solutions [9]. A common interface to the iDorm and its devices is implemented through Universal Plug & Play (UPnP) which is an event-based communication middleware for allowing devices to be plug & play enabling automatic discovery and configuration. A gateway server is used to run the UPnP software devices that interface the hardware devices on their respective networks. The agent implementing our learning and adaptation mechanism was built on top of the low level UPnP control architecture enabling it to communicate with the UPnP devices in the iDorm and thus allowing it to monitor and control these devices. Figure (2) shows the logical network infrastructure of the iDorm.



“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, Journal of Information Sciences, Vol. 171, No. 4, pp. 309-334, May 2005

Figure (2) The iDorm logical network infrastructure.

There is 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 which is shown in Figure (3).



Figure (3) Video Projector and Multi-Media PC in the iDorm.

Any networked computer that can run a standard Java process can access and control the iDorm directly, thus this PC (Figure 4-a) can also act as an interface to control the devices in the room. Equally the interface to the devices could be operated from physically portable computational artefacts that can monitor and control the iDorm wirelessly such as a handheld PDA supporting Bluetooth wireless networking or a mobile phone shown in Figure (4-b) and (4-c) respectively. So it is possible to adjust the environment from anywhere inside and in the vicinity of the room which forms a type of “remote control” interface that would be particularly suitable to elderly and disabled users. There is also an internet Fridge in the iDorm shown in Figure (4-d) that incorporates an intelligent user friendly server with touch screen capability, which can also be used to control the devices in the room.



Figure (4): a) PC based interface. b) Portable iPAQ interface. c) Mobile phone interface. d) iFridge interface.

Our agent learning mechanism and interface currently operates from the standard multi-media PC in the iDorm. It is possible however for our agent to be embedded into any part of the environment. In terms of

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, Journal of Information Sciences, Vol. 171, No. 4, pp. 309-334, May 2005

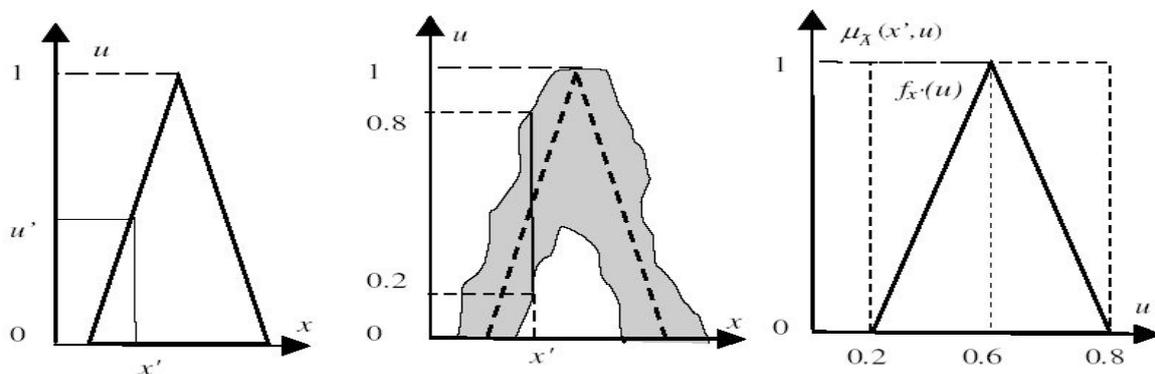
software the cross platform versatility of the Java programming language which the agent was written with, could allow it to be embedded onto internet devices. By embedding agents into such devices and integrating wireless communications (including wireless based interfaces, such as PDAs), will lead to the kind of pervasive transparent infrastructure that is characteristic of an ambient intelligent system.

3. Type-2 Fuzzy Sets

The task of designing an intelligent agent to effectively fulfil the needs of the user in UCE is akin to finding a solution to a highly challenging control problem. The environment within which the agent must operate can be viewed as a very complex control system, in which the user controlling it forms an essential part. The environment facing the human controller is so complicated that any mathematical model, if it exists, is strongly non-linear. In addition, the human controller, in their own right, is largely non-deterministic and a highly individual part of this system. The task here is to design an intelligent control system to realise the ambient intelligence vision [6] and control the environment on behalf of the human user [18].

Type-2 fuzzy sets are able to model the numerical and linguistic uncertainties faced by our agent controller to enable it to better model the user’s behaviours while handling the changing dynamics of the environment and the user activity, which is an essential requirement for an ambient intelligent system.

Type-2 fuzzy sets are able to model the numerical and linguistic uncertainties because the membership functions are themselves fuzzy [15]. One can imagine blurring the type-1 membership function depicted in Figure (5-a) by shifting the points on the triangle either to the left or to the right and not necessarily by equal amounts as in Figure (5-b). Therefore at a specific value of x , say x' , there is no longer a single value for the membership function (u'); instead, the membership function takes on values wherever the vertical line intersects the blurred area shared in grey. Those values need not all be weighted the same; hence, we can assign an amplitude distribution to all of those points. Doing this for all $x \in X$, we create a three-dimensional membership function which is a type-2 membership function that characterises a type-2 fuzzy set.



(a) (b) (c)

Figure (5): a) Type-1 membership function. b) type-2 membership function. c) Triangular secondary membership function plotted in thick line; interval secondary membership function plotted in dashed line.

Formally A type-2 fuzzy set \tilde{A} is characterised by a type-2 membership function $\mu_{\tilde{A}}(x, u)$ [15] where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \quad (1)$$

in which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. \tilde{A} can also be expressed as follows [15]:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0,1] \quad (2)$$

Where $\int \int$ denotes union over all admissible x and u [15].

At each value of x say $x = x'$, the 2-D plane whose axes are u and $\mu_{\tilde{A}}(x', u)$ is called a vertical slice of $\mu_{\tilde{A}}(x, u)$ [15]. A *secondary membership function* is a vertical slice of $\mu_{\tilde{A}}(x, u)$. It is $\mu_{\tilde{A}}(x = x', u)$ for $x' \in X$ and $\forall u \in J_{x'} \subseteq [0,1]$ [15], i.e.

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) / (u) \quad J_{x'} \subseteq [0,1] \quad (3)$$

in which $0 \leq f_{x'}(u) \leq 1$. Due to $\forall x' \in X$, the prime notation on $\mu_{\tilde{A}}(x')$ is dropped and we refer to $\mu_{\tilde{A}}(x)$ as a secondary membership function [15]. According to Mendel [14] the name that we use to describe the entire type-2 membership function is associated with the name of the secondary membership functions; so, for example if the secondary membership function is triangular (as shown in Figure (5-c)) then we refer to $\mu_{\tilde{A}}(x, u)$ as a triangular type-2 membership function.

Based on the concept of secondary sets, type-2 fuzzy sets can be written as the union of all secondary sets as follows [15].

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x) / (x) = \int_{x \in X} \left[\int_{u \in J_x} f_x(u) / u \right] / x \quad J_x \subseteq [0,1] \quad (4)$$

The domain of secondary membership functions is called *primary membership* of x [15], and in Equation (4) J_x is the primary membership function of x , where $J_x \subseteq [0,1]$ for $\forall x \in X$ [15].

The inherent uncertainties in a type-2 membership function are encapsulated within the bounded regions termed Footprint of Uncertainty (FOU) [15], which is shown as a the grey region in Figure (5-b). Formally the uncertainty in the primary membership function of \tilde{A} consists of the bounded region defined as the FOU [15] which is the union of all primary memberships [15], i.e.,

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \quad (5)$$

The *FOU* therefore determines the extent of the uncertainties present in \tilde{A} [15].

Our learning and adaptation technique uses an interval type-2 FLC (using interval type-2 fuzzy sets to represent the inputs and outputs) as opposed to a general type-2 FLC. The fuzzy sets are represented using interval type-2 membership functions in which the secondary membership grades are equal to unity [12]. A type-2 interval membership function is represented by its left and right end-points, these two end points are associated with two type-1 membership functions referred to as upper and lower membership functions which are also the upper and lower bounds for the footprint of uncertainty FOU of the type-2 set [14]. Figure (5-c) illustrates the interval secondary membership function (plotted with the dashed line) at x' . The end points of the secondary membership function further reflect the upper and lower bounds of the FOU in the type-2 set shown in Figure (5-b). Formally the upper and lower membership function of a fuzzy set \tilde{A} associated with the upper and lower bounds is denoted by $\bar{\mu}_{\tilde{A}}(x), \forall x \in X$ and $\underline{\mu}_{\tilde{A}}(x), \forall x \in X$ respectively. According to Mendel [14] we can re-express Equation (4) as follows to represent the type-2 fuzzy set \tilde{A} in terms of upper and lower membership functions as follows:

$$\tilde{A} = \mu_{\tilde{A}}(x, u) = \int_{x \in X} \mu_{\tilde{A}}(x) / (x) = \int_{x \in X} [\int_{u \in J_x} f_x(u) / u] / x = \int_{x \in X} [\int_{\mu \in [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]} f_x(u) / u] / x \quad (6)$$

The secondary membership $\mu_{\tilde{A}}(x)$ can therefore be expressed in terms of upper and lower membership functions as shown below [14]:

$$\mu_{\tilde{A}}(x) = \int_{\mu \in [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]} f_x(u) / u \quad (7)$$

In the case of interval type-2 fuzzy sets when the secondary membership function are interval sets where $f_x(u) = 1$, the interval type-2 fuzzy set can be written as follows [14]:

$$\mu_{\tilde{A}}(x) = \int_{x \in X} [\int_{\mu_{\underline{\tilde{A}}}(x), \mu_{\overline{\tilde{A}}}(x)} 1/u] / x \quad (8)$$

An Interval type-2 FLC is computationally far less intensive than a general type-2 FLC, and is thus better suited for embedded computational artefacts.

From the above discussion, we see that type-2 FLCs using type-2 fuzzy sets have many advantages when compared to type-1 FLCs. For example, as type-2 fuzzy sets are able to handle the numerical and linguistic uncertainties faced by the agent operating in a UCE, then FLCs that are based on type-2 fuzzy sets will have the potential to produce a better performance than the type-1 FLCs. In addition, type-2 fuzzy sets enable us to handle the uncertainty associated with trying to determine the exact membership functions for the fuzzy sets associated with the inputs and outputs of the FLC [12]. The FOU handles the rich variety of choices that can be made for a type-1 membership function, i.e. by using type-2 fuzzy sets instead of type-1 fuzzy sets, the issue of which type-1 membership function to choose diminishes in importance [16]. Using type-2 fuzzy sets to represent the FLC inputs and outputs will also result in the reduction of the FLC rule base when compared to using type-1 fuzzy sets. This is because type-2 fuzzy sets rely on uncertainty represented in the footprint of uncertainty to cover the same range as type-1 fuzzy sets with a much smaller number of labels. As the number of inputs to the FLC increase the potential rule reduction as a consequence of fewer labels becomes significantly greater [14]. In terms of the FLC, uncertainty can also fire rules which are not available in type-1 FLC [14]. In type-2 FLC each input and output will be represented by a large number of type-1 fuzzy sets which are embedded within the FOU's of the type-2 fuzzy sets. The use of such a large number of type-1 fuzzy sets to describe the input and output variables allows for greater accuracy in capturing the subtle behaviours of the user in the environment.

4. The Learning and Adaptation Techniques for the Type-2 Agent

The agents learn and adapt to the user behaviours in UCEs using our type-2 Adaptive Online Fuzzy Inference System (AOFIS) technique which is an unsupervised data-driven one-pass approach for extracting fuzzy rules and membership functions from data, to learn an interval type-2 FLC that will model the user's behaviours. The data is collected by monitoring the user in the environment over a period of time. The learnt type-2 FLC provides an inference mechanism that will produce output control based on the current state of the inputs. Our adaptive type-2 FLC will therefore control the environment on behalf of the user and will also allow the rules to be adapted online as the user's behaviour drifts over time.

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, Journal of Information Sciences, Vol. 171, No. 4, pp. 309-334, May 2005

Our type-2 AOFIS technique aims to realise the vision of ambient intelligence by having the following characteristics:

- The agent is responsive to the particular needs and preferences of the user.
- The user is forever in control and can override the agent’s responses at any time.
- The agent learns the user behaviour and controls the environment on the user behalf in a non-intrusive way (although the user may be aware of the high-tech interface, he is unaware of the agent’s presence).
- The agent’s learnt behaviours can be adapted online as a result of changes in the occupant’s behaviour.
- Learning is life-long in that agent behaviours can be adapted and extended over a long period of time as a result of changes in environmental conditions and user activity.
- The agent uses a simple one pass learning mechanism for learning the user’s behaviours, and is therefore not computationally expensive.

These features satisfy many of the requirements for the ambient intelligence vision defined by the Information Society Technologies Advisory Group (ISTAG) to the European Commission [6].

AOFIS comprises of five phases as follows (as illustrated in Figure (6)).

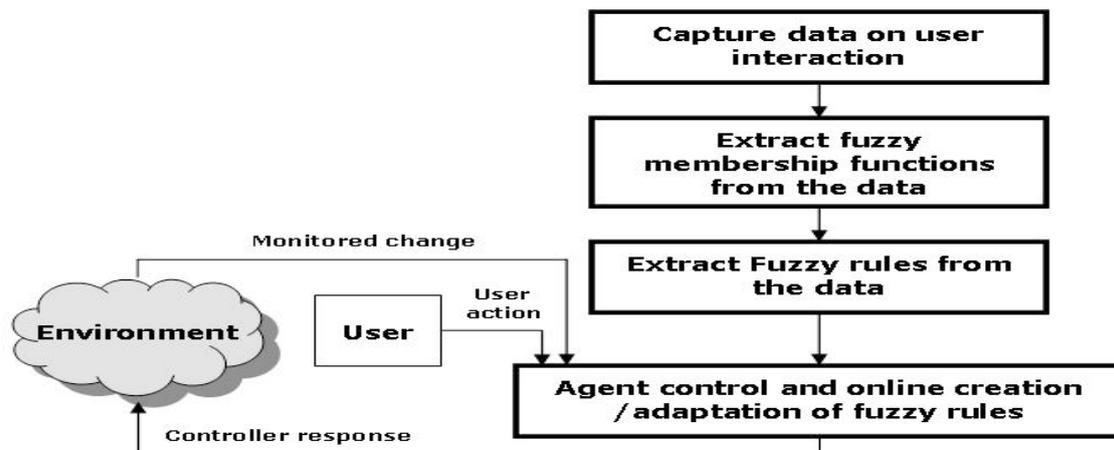


Figure (6): Flow Diagram Showing Five Phases of AOFIS

4.1 Capturing Input Output Data

The agent initially monitors the user’s actions in the environment. Whenever the user changes actuator settings, the agent records a ‘snapshot’ of the current inputs (sensor states) and the current outputs (actuator states with the new altered values of whichever actuators were adjusted by the user). These ‘snapshots’ are accumulated over a period of time (three days in case of our experiments) so that the agent observes as much of the user’s interactions within the environment as possible. AOFIS learns a descriptive model of the user’s behaviours from the data accumulated by the agent. Therefore given a set of multi-input multi-output data pairs:

$$(x^{(t)}; y^{(t)}), \quad t = 1, 2, \dots, N \quad (9)$$

where N is the number of data instances, $x^{(t)} \in R^n$ and $y^{(t)} \in R^k$. AOFIS extracts rules which describe how the k output variables $y = (y_1, \dots, y_k)$ are influenced by the n input variables $x = (x_1, \dots, x_n)^T \in R^n$ based on the sampled data. In our experiments in the iDorm we used 7 sensors for our inputs and 10 actuators for our outputs. The fuzzy rules which are extracted represent local models that map a set of inputs to the set of outputs without the need for formulating any mathematical model. Individual rules can therefore be adapted online influencing only specific parts of the descriptive model learnt by the agent.

4.2 Fuzzy Membership Function Extraction

It is necessary to be able to categorise the accumulated user input/output data into a set of fuzzy membership functions which quantify the raw crisp values of the sensors and actuators into linguistic labels such as normal, cold or hot. AOFIS is based on learning the particularised behaviours of the user and therefore requires these membership functions be defined from the user’s input/output data recorded by the agent. In our previous work, [5] we have developed a technique for generating type-1 membership functions from data that was based on using a Double Clustering approach combining Fuzzy-C-Means (FCM) and agglomerative hierarchical clustering [5]. We used this technique to generate type-1 membership functions and then added the FOU’s for the fuzzy sets to generate the interval type-2 membership functions as illustrated in figure (7-a) and (7-b).

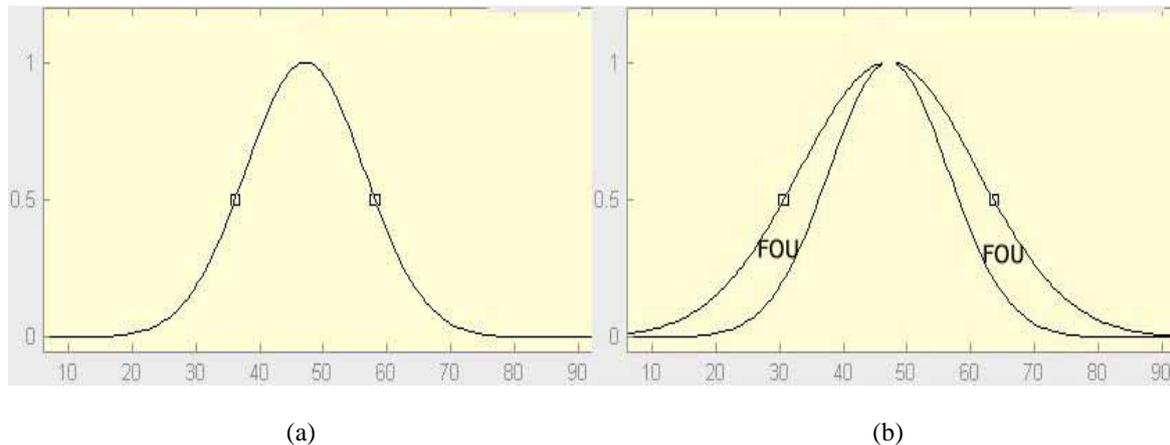


Figure (7): a) Type-1 fuzzy set with no uncertainties. b) type-1 set with added FOU to form type-2 fuzzy set.

Gaussian interval type-2 membership functions with uncertain standard deviations are used to describe the type-2 fuzzy sets \tilde{A}_z^j , (where $z = 1, 2, \dots, V$ and V represents the number of type-2 fuzzy sets for a variable j) the mathematical definition of which is

$$\mu_{\tilde{A}_z^j}(x) = \exp\left\{-\frac{1}{2}\left(\frac{x - m_z^j}{\sigma_z^j}\right)^2\right\} \quad \sigma_z^j \in [\sigma_{z_1}^j, \sigma_{z_2}^j] \quad (10)$$

where m_z^j is the value of the centre (average) and σ_z^j are the values of the spreads for each gaussian interval type-2 membership function z , for the j -th input/output variable.

We therefore obtain a set of V interval type-2 fuzzy membership functions defined for each input and output parameter of the user data that was sampled. These membership functions are distributed over the range of values of each parameter. The membership functions at the boundaries are modified such that they are extended indefinitely beyond their respective centres with an upper and lower membership value of 1. A semantic meaning can be associated with each of the resulting fuzzy sets. Specifically depending on the value of index z , a meaningful symbolic label can be given to \tilde{A}_z^j .

4.3 Fuzzy Rule Extraction

The defined set of interval type-2 membership functions are combined with the existing user input/output data to extract the rules defining the user's behaviours. The fuzzy rule extraction approach used by the type-2 AOFIS is based on an Enhanced version of the Mendel Wang (MW) method [5, 18] developed by L.X. Wang and by Mendel [14]. This is a one pass technique for extracting fuzzy rules from the sampled data. The fuzzy sets for the antecedents and consequents of the rules divides the input and output space into fuzzy regions.

The type-2 AOFIS extracts multi-input multi-output rules which describe the relationship between $y=(y_1, \dots, y_k)$ and $x=(x_1, \dots, x_n)^T$, and take the following form:

$$IF \ x_1 \text{ is } \tilde{A}_1^l \ \dots \ \text{and } x_n \text{ is } \tilde{A}_n^l, \text{ THEN } y_1 \text{ is } \tilde{B}_1^l \ \dots \ \text{and } y_k \text{ is } \tilde{B}_k^l \quad (11)$$

$l = 1, 2, \dots, M$, where M is the number of rules and l is the index of the rules. There are V interval type-2 fuzzy sets $\tilde{A}_s^q, q = 1, \dots, V$, defined for each input x_s where $(s = 1, \dots, n)$. There are V interval type-2 fuzzy sets $\tilde{B}_c^h, h = 1, \dots, V$, defined for each output y_c where $(c = 1, \dots, k)$. AOFIS now extracts rules in the form of Equation (11) from the data.

To simplify the following notation, the method for rules with a single output is shown, as the approach is quite easily expanded to rules with multiple outputs. In the following steps we will show the different steps involved in rule extraction:

Step 1: For a fixed input-output pair $(x^{(t)}; y^{(t)})$ in the dataset, $(t=1, \dots, N)$, compute the upper and lower membership values $\bar{\mu}_{\tilde{A}_s^q}(x_s^{(t)})$ and $\underline{\mu}_{\tilde{A}_s^q}(x_s^{(t)})$ for each fuzzy set $q=1, \dots, V$, and for each input variable s ($s = 1, \dots, n$). Find $q^* \in \{1, \dots, V\}$ such that

$$\mu_{\tilde{A}_s^{q^*}}^{cg}(x_s^{(t)}) \geq \mu_{\tilde{A}_s^q}^{cg}(x_s^{(t)}) \quad (12)$$

for all $q=1, \dots, V$, where $\mu_{\tilde{A}_s^q}^{cg}(x_s^{(t)})$ is the centre of gravity of the interval membership of \tilde{A}_s^q at $x_s^{(t)}$ as follows [14]:

$$\mu_{\tilde{A}_s^q}^{cg}(x_s^{(t)}) = f_{x_s^{(t)}}^{cg}(\tilde{A}_s^q) = \frac{1}{2} \left[\bar{\mu}_{\tilde{A}_s^q}(x_s^{(t)}) + \underline{\mu}_{\tilde{A}_s^q}(x_s^{(t)}) \right] \quad (13)$$

Let the following rule be called the rule generated by $(x^{(t)}, y^{(t)})$:

$$IF \ x_1 \text{ is } \tilde{A}_1^{q^*(t)} \ \dots \ \text{and } x_n \text{ is } \tilde{A}_n^{q^*(t)} \ \text{THEN } y \text{ is centred at } y^{(t)} \quad (14)$$

For each input variable x_s there are V type-2 fuzzy sets \tilde{A}_s^q , $q=1, \dots, V$ to characterise it; so that the maximum number of possible rules that can be generated is V^n . However given the dataset only those rules among the V^n possibilities whose dominant region contains at least one data point will be generated. In step 1 one rule is generated for each input-output data pair, where for each input the fuzzy set that achieves the maximum membership value at the data point is selected as the one in the IF part of the rule, as explained in Equations (12), (13) and (14).

This however is not the final rule which will be calculated in the next step. The weight of the rule is computed as

$$wi^{(t)} = \prod_{s=1}^n \mu_{\tilde{A}_s^{q^*}}^{cg}(x_s^{(t)}) \quad (15)$$

The weight of a rule $wi^{(t)}$ is a measure of the strength of the points $x^{(t)}$ belonging to the fuzzy region covered by the rule.

Step 2: Step 1 is repeated for all the t data points from 1 to N to obtain N data generated rules in the form of Equation (14). Due to the fact that the number of data points is quite large, many rules are generated in step 1, that all share the same IF part and are conflicting, i.e. rules with the same antecedent membership functions and different consequent values. In this step rules with the same IF part are combined into a single rule. The N rules

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, Journal of Information Sciences, Vol. 171, No. 4, pp. 309-334, May 2005

are therefore divided into groups, with rules in each group sharing the same IF part. If we assume that there are M such groups. Let group l have N_l rules in the following form:

$$IF\ x_1\ is\ \tilde{A}_1^l\ \dots\ and\ x_n\ is\ \tilde{A}_n^l\ THEN\ y\ is\ centred\ at\ y^{(t_u^l)} \quad (16)$$

Where $u = 1, \dots, N_l$ and t_u^l is the index for the data points in group l . The weighted average of all the rules in the conflict group is then computed as

$$av^{(l)} = \frac{\sum_{u=1}^{N_l} y^{(t_u^l)} wi^{(t_u^l)}}{\sum_{u=1}^{N_l} wi^{(t_u^l)}} \quad (17)$$

We now combine these N_l rules into a single rule of the following form:

$$IF\ x_1\ is\ \tilde{A}_1^l\ and\ \dots\ and\ x_n\ is\ \tilde{A}_n^l,\ THEN\ y\ is\ \tilde{B}^l \quad (18)$$

Where the output fuzzy set \tilde{B}^l is chosen based on the following. Among the V output interval type-2 fuzzy sets $\tilde{B}^1, \dots, \tilde{B}^V$ find the B^{h*} such that

$$\mu_{\tilde{B}^{h*}}^{cg}(av^{(l)}) \geq \mu_{\tilde{B}^h}^{cg}(av^{(l)}) \quad (19)$$

for $h = 1, 2, \dots, V$, \tilde{B}^l is chosen as \tilde{B}^{h*} , where $\mu_{\tilde{B}^h}^{cg}(av^{(l)})$ is the centre of gravity of the interval membership of \tilde{B}^h at $av^{(l)}$ as in Equation (13).

As mentioned above AOFIS deals with input-output data pairs with multiple outputs. Step 1 is independent of the number of outputs for each rule. Step 2 is simply expanded to allow rules to have multiple outputs where the calculations in Equations (17), (18) and (19) are repeated for each output value.

4.4 Agent Controller

Once the agent has extracted the membership functions and the set of rules from the user input/output data, it has then learnt the type-2 FLC that captures the human behaviour. The agent FLC can start controlling the environment on behalf of the human according to his desires. The agent starts to monitor the state of the environment and affect actuators based on its learnt type-2 FLC that approximate the particularised preferences of the user. This operation is performed in a non-intrusive way to realise the ambient intelligence vision [6]. Figure (8) shows a block diagram of the interval type-2 FLC which consists of a fuzzifier, rule base, fuzzy inference engine, centre of sets type-reducer and defuzzifier, more information about this real time type-2 FLC can be found in [8].

The type-2 FLC works in the following way, the crisp inputs comprising the sensory state of the environment are first fuzzified into the input interval type-2 fuzzy sets (we will use singleton fuzzification) which then activates the inference engine and the rule base to produce output type-2 fuzzy sets. The type-2 fuzzy outputs of the inference engine are then processed by the type-reducer which combines the output sets and then performs a centroid calculation which leads to type-1 fuzzy sets called the type-reduced sets [14]. The defuzzifier can then defuzzify the type-reduced type-1 fuzzy outputs to produce crisp outputs to be fed to the actuators.

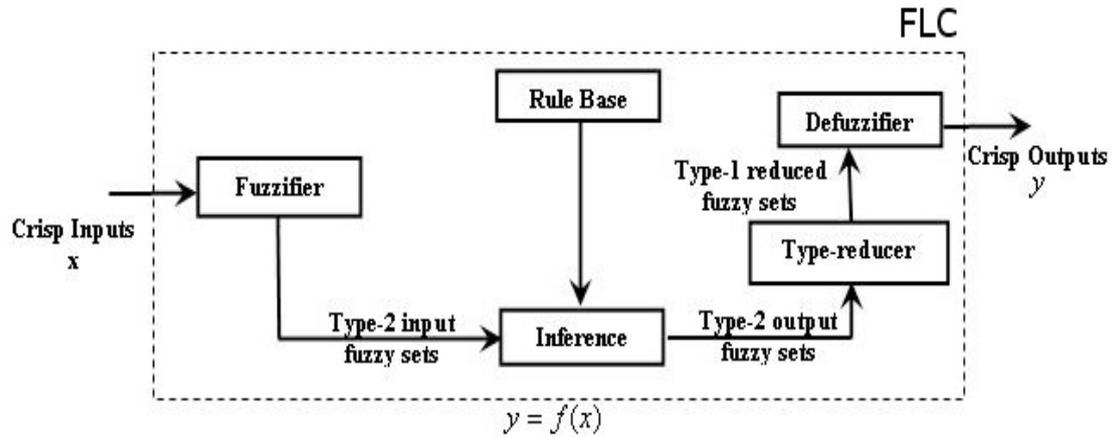


Figure (8): Block diagram of a type-2 FLC

In the inference engine the firing interval for each rule based on the input and antecedent operations is calculated as follows [14]:

$$F^l(x') = [\underline{f}^l(x'), \overline{f}^l(x')] \equiv [\underline{f}^l, \overline{f}^l] \quad (20)$$

where

$$\underline{f}^l(x') = \prod_{i=1}^n \underline{\mu}_{\tilde{A}_i}(x_i) \quad (21)$$

and

$$\overline{f}^l(x') = \prod_{i=1}^n \overline{\mu}_{\tilde{A}_i}(x_i) \quad (22)$$

where $l = 1, \dots, M$ refers to the l^{th} rule in the rule base and n is the number of inputs.

Type-reduction was proposed by Karnik and Mendel [10, 11, 13], it is called type-reduction because this operation takes us from the type-2 output sets of the inference engine to a type-1 set that is termed the type-reduced set [13]. These type-reduced sets are then defuzzified to obtain crisp outputs of the actuator values.

As we are dealing with interval sets, the type-reduced set for the c^{th} output Y_{TR_c} will also be an interval set [14] and has the following structure:

$$Y_{TR_c} = [y_{lc}, y_{rc}] \quad (23)$$

As in [13] we use the centre of sets type reduction, as it has reasonable computational complexity that lies between computationally expensive centroid type-reduction and the simple height and modified height type-reduction which have a problem when only one rule fires [14]. The computation of centre of sets type-reduction will allow for real time operation if the rule base is not large. The type reduced set using the centre of sets type-reduction can be expressed as follows:

$$Y_{\cos}(x)_c = [y_{lc}, y_{rc}] = \int_{y_c^l \in [y_{lc}^l, y_{rc}^l]} \dots \int_{y^M \in [y_{lc}^M, y_{rc}^M]} \int_{f^l \in [\underline{f}^l, \bar{f}^l]} \dots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} 1 / \frac{\sum_{l=1}^M f^l y_c^l}{\sum_{l=1}^M f^l} \quad (24)$$

Where $Y_{\cos}(x)_c$ for the c^{th} output is an interval set determined by its left most point y_{lc} and its right most point y_{rc} , M is the number of rules. y_c^l corresponds to the centroid of the type-2 interval consequent set \tilde{B}_c^l of the l^{th} rule for the c^{th} output; y_c^l is a type-1 interval set determined by its left most point y_{lc}^l and its right most point y_{rc}^l [13]. f^l denotes the firing strength (degree of firing) of the l^{th} rule which is an interval type-1 set determined by its left most \underline{f}^l and right most point \bar{f}^l [13] where \underline{f}^l is calculated using Equation (21) and \bar{f}^l is calculated using Equation (22).

The calculation of the type-reduced sets is divided into two stages. In the first stage the centroids of type-2 interval consequent sets of the l^{th} rule are calculated using the iterative procedure developed by [14]. This is conducted ahead of time and before starting the control cycle of the agent's FLC. The second stage consists of calculating the type-reduced sets using the iterative procedure developed by [13, 14]. The type-reduced sets are then defuzzified to produce the crisp output for the actuators; this occurs at each control cycle. The iterative procedure for type-reduction is proven [13, 19] to converge in no more than M iterations to find y_{rc} and M iterations to find y_{lc} where M is the number of rules. As mentioned earlier the potential for the rule base to become uncontrollably large is reduced by that fact that given the dataset only those rules among the V^n possibilities whose dominant region contains at least one data point will be generated. In our system we have also set a memory limit that saves only the most used rules to avoid an increase in the number of rules beyond a certain limit that will hinder the real time performance of our system. As we require less type-2 fuzzy sets for

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, Journal of Information Sciences, Vol. 171, No. 4, pp. 309-334, May 2005

accurately representing the input parameters, this also means that the potential number of rules is considerably reduced.

From the type-reduced stage we produce for each output a type-reduced set $Y_{\cos}(x)_c$. Each type-reduced set is an interval type-1 set determined by its left most point y_l and right most point y_r . We defuzzify the interval set by using the average of y_{lc} and y_{rc} hence the defuzzified crisp output for each output c is [19].

$$Y_c(x) = \frac{y_{lc} + y_{rc}}{2} \quad (25)$$

4.5 Online Adaptation and Life Long Learning

In the previous steps we have shown how our agent can learn a type-2 FLC that approximates the user's behaviour. However, the user may need to make adjustments to tune the system or their behaviour might change as the user requirements change over time. So our agent needs to adapt to the user's behavioural changes in a non intrusive manner and in a short time interval.

In realising the non-intrusive aspect of ambient intelligence [6] whenever the user is not happy with the agent's actions, he can always override the agent's control responses by simply altering the manual control of the system. When this occurs the agent will adapt its rules online or add new rules based on the new user preferences.

Whenever the user overrides the agent's control responses and actuates any of the controlled output devices, a snapshot of the state of the environment is recorded and passed to the rule adaptation routine. Each input parameter in the input vector \mathbf{x} is compared to each of the antecedent sets $\tilde{A}_s^{(l)}$ of a given rule in the rule base to determine its upper and lower membership values. The weight of the rule is then calculated to determine if the degree of firing of the rule in Equation (15) $w_i^{(l)} > 0$, meaning that the rule fired, and would therefore have contributed to the overall control response generated by the agent's FLC. The consequent fuzzy sets that give the highest membership to the user defined actuator values are selected to replace the consequent sets of all fired rules in the rule base. The consequent fuzzy sets are found as in Equation (4) by calculating centre of gravity of the interval membership.

$$\mu_{\tilde{B}_c^{h^*}}(y_c) \geq \mu_{\tilde{B}_c^h}(y_c) \quad (26)$$

for $h = 1, 2, \dots, V$. The \tilde{B}_c is chosen as $\tilde{B}_c^{h^*}$. Where $c=1, 2, \dots, k$. The fired rules are therefore adapted to better reflect the user's updated actuator preferences given the current state of the environment.

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, *Journal of Information Sciences*, Vol. 171, No. 4, pp. 309-334, May 2005

If none of the existing rules fired, new rules are added based on forming rules from the input fuzzy sets. For each input parameter x_s the fuzzy sets that give a membership value where $\mu_{A_s^g}^{cg}(x_s^{(t)}) > 0$ are identified.

This leads to a grid of identified fuzzy set(s) for each input parameter. From this grid new rules are constructed based on each unique combination of consecutive input fuzzy sets. The consequent fuzzy sets for each of the new rules are determined using Equation (26). This allows new rules to be gradually added to the rule base. The agent will also add new rules when the currently monitored environmental state is undefined by the existing rules in the rule base; i.e. none of the existing rules fired. In this case the agent will create new rules where the antecedent sets reflect the current input states of the environment and the consequent fuzzy sets are based on the current state of the actuators.

The agent adopts life long learning where it adapts its rules as the state of the environment and the preferences of the user change over a significantly long period of time.

5. Experimental Results

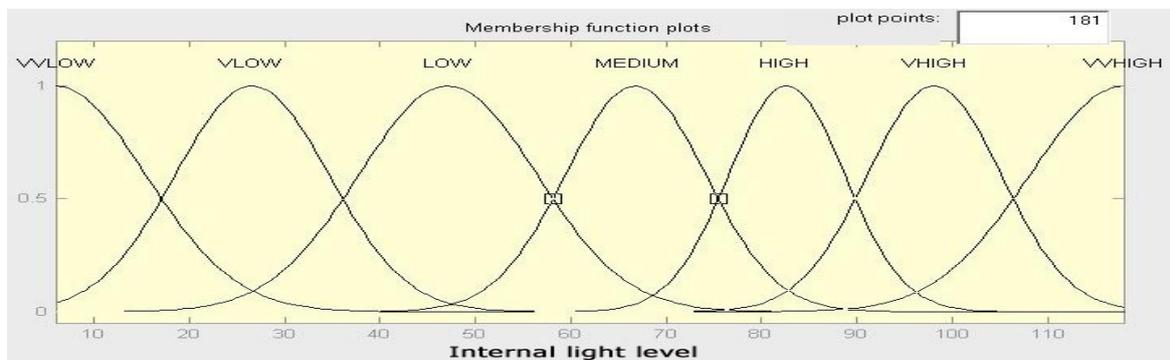
We have performed unique experiments in which a user lived in the iDorm (shown in figure (9)) for a total period of five days. During the initial monitoring phase which lasted for three consecutive days in late summer early autumn (early September), the agent recorded the user interactions with the environment. The user performed the normal variety of behaviours and activities one would associate with a study bedroom environment; and the agent recorded the user interactions in an unobtrusive and non-intrusive way to realise the vision of ambient intelligence [6]. Seven input sensors were monitored which are: internal light level, external light level, internal temperature, external temperature, chair pressure, bed pressure and time measured as a continuous input on an hourly scale. Ten output actuators were controlled consisting of the four variable intensity spot lights, the desk and bed side lamps, window blinds, the heater and the two PC based applications comprising of a word processing program and a media playing program. The outputs thus covered the spectrum of physical devices and computer based applications found in a typical study bedroom environment. As we mentioned previously the user was able to interface with the devices in the room via the multi-media PC on which our intelligent agent was embedded. In a similar way the agent could have been embedded on any other wireless or networked computational artefact in the room allowing a remote intelligent embedded interface to the iDorm environment.



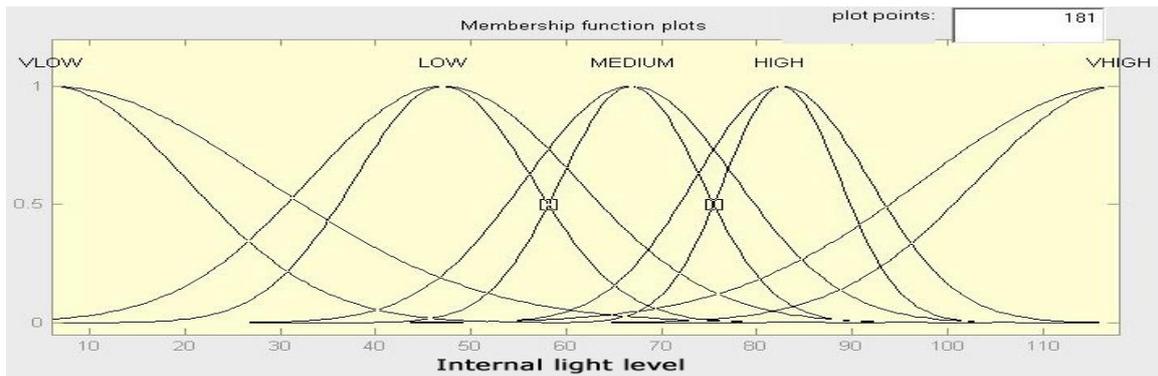
Figure (9): User in the iDorm

5.1 Offline Experiments

The data from the iDorm that was captured during the monitoring phase was used to compare the offline performance of the type-1 AOFIS with three other soft-computing based techniques which are Genetic Programming (GP), the Adaptive-Neuro Fuzzy Inference System (ANFIS) and the Multi-Layer Perceptron Neural Network [5]. The dataset comprised of 408 instances and was randomised into six samples. Each sample was then split into a training and test set consisting of 272 and 136 instances respectively. The offline performance error for each technique was obtained on the test instances as the Root Mean Squared Error which was also scaled to account for the different ranges of the output parameters. From our previous work it was found that for the type-1 AOFIS, the optimum number of type-1 fuzzy sets for AOFIS is 7 [5]. The type-1 AOFIS had outperformed the ANFIS and the MLP and gave a comparable result to the GP. The iterative nature of the compared approaches made them more computationally intensive than the one pass type-1 AOFIS technique which makes it better suited for embedded agents with limited computational resources. The other approaches cannot easily be adapted online as this would necessitate their internal structures to be re-learnt every time either new rules were added or existing rules were adapted. So our method is unique in that it can learn a good model of the user's behaviour which can then be adapted online in a life long learning mode in a non intrusive manner.



(a)



(b)

Figure (10): a) Type-1 Fuzzy Sets. b) Derived type-2 Fuzzy Sets, for Internal Light Level.

We then proceeded to determine if our type-2 AOFIS would produce an improved performance over the type-1 AOFIS using the same data samples. The training instances in each data sample were used to generate the type-2 agent parameters. 7 type-1 sets were used to represent the input and output parameters of the type-1 agent as this was shown to be the optimum number of sets from our previous experiments. Five interval type-2 sets were derived from the 7 type-1 sets for each parameter to form an interval type-2 FLC. The interval type-2 fuzzy sets covered the same ranges as the type-1 fuzzy sets such that the type-1 sets were approximately embedded within the type-2 sets. Figure (10-a) and (10-b) shows for the input parameter *internal light level* the 7 type-1 fuzzy sets used for the type-1 agent FLC and the 5 type-2 fuzzy sets used for the type-2 agent FLC.

The results obtained showed that the type-2 agent produced an average scaled error of 0.1255 and a scaled standard deviation of 0.1138. In comparison the type-1 agent produced an average scaled error of 0.1324 and a scaled standard deviation of 0.1257. So the type-2 agent had produced a smaller error (i.e. captured better the human behaviour) than the type-1 agent. The type-2 agent generated 121 rules from the 272 training instances compared with the type-1 FLC that produced 153 rules.

5.2 Online Experiments

The online performance of the agent was evaluated on how well the type-2 AOFIS could model the user's behaviour from their observed activity that had been recorded over the initial three days of monitoring in early September. The performance of the learnt type-2 FLC could then be gauged online in its ability to control the environment and satisfy the preferences of the user when the environmental conditions were significantly different such that differences between the original user dataset and the current conditions would be considerably higher. In this way we could determine if the type-2 agent adapted better (handled better the uncertainties) to the new environmental conditions than a traditional type-1 agent. The dataset accumulated during the monitoring

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, *Journal of Information Sciences*, Vol. 171, No. 4, pp. 309-334, May 2005

phase was used to learn the type-1 and type-2 FLC's. Both agents were then each separately run online for two days in mid winter (mid December) during which they monitored the environment and user's activities, and produced the appropriate control responses based on their learnt FLC's. During this time each agent's FLC controlled the environment in a non-intrusive and invisible way while the user continued to carry out the assortment of behaviours and activities they were performing during the initial monitoring phase. The user could now however override and adapt the agent's learnt control responses, if it was necessary to modify and tune them further.

One of the characteristics of our agent is that the user is always in control and he can override the agent at any time and his instructions are executed immediately, to achieve the responsive property implied in the ambient intelligence vision [6] unless safety is compromised. Thus whenever changes to controls were made by the user, the agent received the request, generated new rules or adjusted previously learnt rules and allowed the action through. The agent would autonomously continue to monitor the environment and generate new rules when the state of the environment was not captured by its existing rule base.

The online performance of the agents could be measured by monitoring how well they adjusted the iDorm environment to the user's preferences such that the user intervention was reduced over time.

Figure (11) plots the number of online rule adaptations against time measured in minutes that occurred over the course of the two days for both the type-1 and type-2 FLC's. From Figure (11) we can see that the type-2 agent required significantly less user interaction than the type-1 agent. This is because the type-2 agent had modelled better the user behaviour as it can handle the linguistic and numerical uncertainties facing embedded agents in UCEs. Both plots show the user intervention was initially high but then stabilised by the end of the first day. The type-2 agent initially learnt 121 rules from the user dataset. Over the subsequent two days 92 new rules were created by the agent. In comparison the type-1 FLC initially learnt 153 rules and the agent created 341 new rules over the two days. Both agents were therefore able to learn and adapt in a non intrusive way to the user's preferences over the duration of the two days. The type-2 FLC however was able to adapt better to the new environmental conditions with less user interaction and a fewer number generated of rules. Figure (12) shows an example of the type of rules that our agents produced.

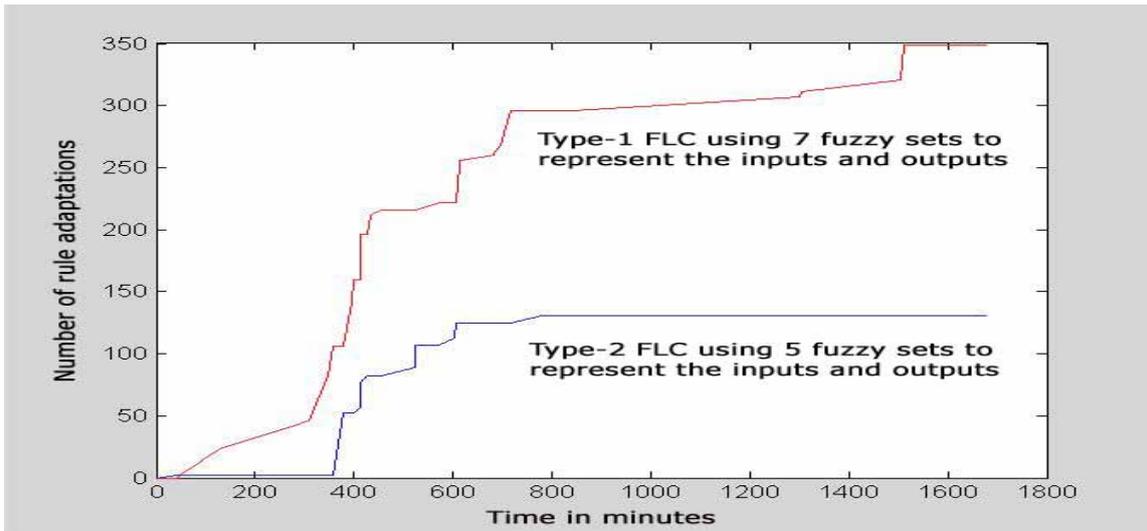


Figure (11): Number of Online Rule Adaptations

```
IF InternalLightLevel is VVLOW AND ExternalLightLevel is VVLOW AND
InternalTemperature is VVHIGH AND ExternalTemperature is MEDIUM AND
ChairPressure is OFF AND BedPressure is ON AND Hour is Evening THEN
ACTION_Light1_value is VHIGH AND ACTION_Light2_value is HIGH AND
ACTION_Light3_value is LOW AND ACTION_Light4_value is VVLOW AND
ACTION_Blind_state is CLOSED AND ACTION_BedLight_state is ON AND
ACTION_DeskLight_state is OFF AND ACTION_Heater_state is OFF AND
ACTION_MSWord_state is STOPPED AND ACTION_MSMediaPlayer_state is
RUNNING
```

Figure (12): Typical Rule Produced by AOFIS

From the experiments we can deduce that the agent has tried to realise the vision of ambient intelligence as it was intelligent and it learnt the user particularised behaviour and adapted it online to any changes in a life long learning mode in a non intrusive way. The agent was also responsive to the user commands. In addition, the intelligent environment in the iDorm was transparent and ubiquitous in that the pervasive interconnected embedded systems were seamlessly integrated into it. The user was therefore unaware of the invisible intelligently responsive infrastructure of the environment.

6. Conclusion

In this paper we presented a novel system for learning and adapting type-2 fuzzy controllers for agents that can be embedded in UCEs. This we hope will be a step towards the realisation of the vision of ambient intelligence.

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, *Journal of Information Sciences*, Vol. 171, No. 4, pp. 309-334, May 2005

These agents are based on type-2 fuzzy systems which are able to handle the different sources uncertainty and imprecision in UCEs to give a good response.

Our agent learnt a type-2 FLC that modelled the user’s particularised behaviour and it was adaptive as it allowed the learnt behaviours to be modified and extended online and in a life-long learning mode as the user’s activity and environmental conditions changed over time. We have demonstrated that our type-2 fuzzy controller is able to reduce the effects of uncertainties that arise as environmental conditions and user behaviours change over a long period of time.

The intelligent learning and adaptation occurred in a non intrusive manner while the user carried out his normal activities in the environment and the agent was always responsive to the user’s commands. The iDorm environment was transparent and ubiquitous and the pervasive infrastructure of the interconnected embedded systems was seamlessly integrated into it. The user was therefore surrounded by an invisible though intelligently responsive intelligent ambience. Our technique was a simple one-pass method and thus it is not computationally expensive and could be incorporated in many embedded devices within pervasive environments.

We carried out unique experiments in which a user stayed in the iDorm for a period totalling five days. The offline and online performance of the type-2 agent using our type-2 AOFIS showed that its type-2 FLC outperformed a type-1 FLC at both learning the behaviours of a user and adapting and tuning its rules online to meet the user’s preferences when the environmental conditions had significantly altered. The type-2 FLC also used less number of rules than the type-1 FLC. We were therefore able model and minimise the effects of uncertainties to produce a better over all performance of the system.

In our future work we propose to design an automated process for generating type-2 fuzzy sets directly from user data.

References

- [1] G. Abowd, M. Eibling, G. Hunt, and H. Lei. “Context Aware Computing,” *IEEE Pervasive Computing*, Volume 1, No 3, pp. 22-23, 2002.
- [2] V. Callaghan, M. Colley, G. Clarke and H. Hagraas, “Embedded Intelligence: Research Issues for Ubiquitous Computing,” *Proceedings of the Ubiquitous Computing in Domestic Environments conference, Nottingham*, Sep, 2001.
- [3] M. Coen. “Design Principles for Intelligent Environments,” *In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI’98)*, 1998.

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, *Journal of Information Sciences*, Vol. 171, No. 4, pp. 309-334, May 2005

- [4] E. M. A. Diederiks, B. M. van de Sluis, and R. van de Ven, “User Interface for WWICE2 Interaction Concepts for the Connected Home”, Submitted in: User Interface 2001 (Internal Philips Conference).
- [5] F. Doctor, H. Hagra, and V. Callaghan, “An Intelligent Fuzzy Agent Approach for Realising Ambient Intelligence in Intelligent Inhabited Environments”, Submitted to the *IEEE transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*.
- [6] K. Ducatel, M. Bogdanowicz, F. Scapolo, and J-C. Burgelman, “Scenarios For Ambient Intelligence in 2010,” IPTS, Seville, Feb, 2001.
- [7] B. Eisenberg, People-centric Computing: MIT's Project Oxygen, *Pacific Connection Web Site*, http://www.gihyo.co.jp/magazine/SD/pacific/SD_0210.html, October 2002.
- [8] H. Hagra, V. Callaghan, and M. Colley, “A Behaviour Based Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots Navigating in Unstructured Environments”, submitted to the *IEEE transactions on Fuzzy Systems*
- [9] H. Hagra, M. Colley, V. Callaghan, G. Clark, and H. Duman, A. Holmes, “A Fuzzy Incremental Synchronous Learning Technique for Embedded Agents Learning and Control in Intelligent Inhabited Environments,” *Proceedings of 2002 IEEE International Conference on Fuzzy Systems*, pp 139-145, Hawaii, USA, 2002.
- [10] N. Karnik and J. Mendel, “An Introduction to type-2 fuzzy logic systems,” USC report, <http://sipi.usc.edu/~mendel/report>, October 1998.
- [11] N. Karnik, J. Mendel and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE Transactions on Fuzzy Systems*, Vol. 7, pp. 643-658, December 1999.
- [12] Q. Liang, N. Karnik, and J. Mendel, "Connection admission control in ATM networks using survey- Based type-2 fuzzy logic systems," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, Vol. 30, pp. 329-339, August 2000.
- [13] Q. Liang, and J Mendel, “Interval type-2 fuzzy logic systems: theory and design,” *IEEE Transactions on Fuzzy Systems*, Vol. 8, pp.535-550, October 2000.
- [14] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice Hall PTR, Prentice Hall Inc, 2001.
- [15] J. Mendel, and R. John, "Type-2 fuzzy sets made simple," *IEEE on Transactions on Fuzzy Systems*, Vol. 10, pp. 117-127, April 2002.

“A Type-2 Fuzzy Embedded Agent to Realise Ambient Intelligence in Ubiquitous Computing Environments”, *Journal of Information Sciences*, Vol. 171, No. 4, pp. 309-334, May 2005

[16] J. Mendel, and H. Wu, "Uncertainty versus choice in rule-based fuzzy logic systems," *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1336-1342, Honolulu, USA, 2002.

[17] G. Riva, P. Loreti, M. Lunghi, F. Vatalaro, and F. Davide, “Presence 2010: The Emergence of Ambient Intelligence,” *Ios Press Amsterdam, The Netherlands*, 2003.

[18] L. X. Wang, and J. M. Mendel, “Generating Fuzzy Rules By Learning From Examples,” *IEEE Trans. On Systems Man and Cybernetics*, vol. 22, pp. 1414-1427, Nov/Dec, 1992.

[19] H. Wu, and J Mendel, “Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems”, *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 5, pp.622-639, October 2002.