

A Type-2 Fuzzy Embedded Agent For Ubiquitous Computing Environments

Faiyaz Doctor, Hani Hagra, Vic Callghan

Department of Computer Science, University of Essex

Wivenhoe Park, Colchester, CO4 3SQ, UK

Email: fdocto@essex.ac.uk, hani@essex.ac.uk, vic@essex.c.uk

Abstract--- In this paper, we describe a novel system for learning and adapting type-2 fuzzy controllers for intelligent agents that are embedded in Ubiquitous Computing Environments (UCE). Our type-2 agents learn online in a non intrusive manner and in a life long learning mode the user behaviours 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 ubiquitous computing environment test bed. We will show how our type-2 agents deal with the uncertainty and imprecision present in UCE to give a very good response that outperforms the type-1 fuzzy agents while using a smaller number of rules.

I. INTRODUCTION

Ubiquitous computing also referred to as pervasive computing, is a paradigm in which computing technology becomes virtually invisible by being embedded in our environments. These Ubiquitous Computing Environments (UCE) will 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 [2]. In this vision people are empowered through a digital environment that is "aware" of their presence and context, and is *sensitive, adaptive and responsive* to their needs [2]. One approach to achieve this 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.

Intelligent agents in UCE are facing a huge amount of uncertainties as the inputs to the agents are uncertain because the sensors measurements are noisy and imprecise and are affected by the environment conditions. In addition, the outputs of the agents are also uncertain due to the change of actuator 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. Moreover, the environmental factors such as the external light

level and temperature change and time of day (morning, evening...etc) can vary over a considerable long period of time due to seasonal variations. However, the main cause of uncertainty is humans occupying these environments as their behaviours and moods are dynamic and non-deterministic and change with time and season, in addition, different words means different things to different people.

The Fuzzy Logic Controller (FLC) is credited with being an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance in face of uncertainty and imprecision, however most of the FLC applications use the type-1 FLC. Type-1 FLCs have the common problem 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 [6]. Once the type-1 membership functions have been chosen, all the uncertainty disappears, because type-1 membership functions are totally precise [7].

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 $[0,1]$ [6]. 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 make it possible to directly model and handle uncertainties [6,7]. Therefore FLCs that use type-2 fuzzy sets to represent the inputs and outputs of the FLC can handle the uncertainties facing embedded agents in UCE 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 [6].

In this paper, we will present a novel system for learning and adapting type-2 fuzzy controllers for agents that can be embedded in UCE. 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. Our technique is a one pass method which does not require heavy computation so it is suitable for embedded computers which have limited computational abilities. We will present unique experiments in which the type-2 agent have learnt and adapted to the user behaviour during a total stay of five days in the intelligent Dormitory (iDorm) which is a real ubiquitous computing environment test bed.

In Section II, we will describe our test bed for UCE; the iDorm. In Section III, we will describe our learning and adaptation techniques for the type-2 agents. In Section IV, we will present the experiments and results. Finally conclusions and future work are presented in Section V.

II. THE iDorm

The iDorm shown in Fig. 1 is a real UCE test bed comprising of a large number of embedded sensors, actuators, processors and networks in a student bedroom environment. The iDorm is a multi-user space containing areas of different activities such as sleeping, working and entertaining. It contains the normal mix of furniture, found in a typical student study/bedroom environment, including a bed, work desk and a wardrobe. 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. Any networked computer that can run a standard Java process can access and control the iDorm directly, thus this PC 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 or a mobile phone. So it was possible to adjust the environment from anywhere inside and nearby outside the room, this 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 that incorporates an intelligent user friendly server with touch screen capability, which can also be used to control the devices in the room.



Fig. 1. The iDorm

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 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), this will lead to the kind of pervasive transparent infrastructure characteristic of an ambient intelligent system.

The iDorm is fitted with a liberal placement of sensors 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 [2]. 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.

III. THE LEARNING AND ADAPTATION TECHNIQUES FOR THE Type-2 AGENT

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 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 determining the exact membership functions for the fuzzy sets associated with the inputs and outputs of the FLC [5]. 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 [8]. 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. As the number of inputs to the FLC increase the potential rule reduction as a consequence of fewer labels becomes significantly greater [6]. In terms of the FLC, uncertainty can also fire rules which are not available in type-1 FLC [6]. Also, 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. Our technique uses an interval type-2 FLC (using interval type-2 fuzzy sets to represent the inputs and outputs) as interval type-2 FLC is computationally far less intensive than a general type-2 FLC, and is thus better suited for embedded computational artefacts.

The agents learn and adapt to the user behaviours in UCE 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 a type2 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. AOFIS comprises of five phases as follows (as illustrated in Fig. 2).

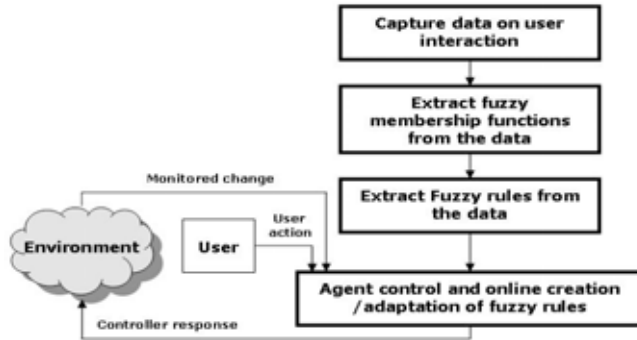


Fig. 2. Flow Diagram Showing Five Phases of AOFIS

A. 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 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:

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

where N is the number of data instances, $x^{(t)} \in \mathbb{R}^n$ and $y^{(t)} \in \mathbb{R}^k$. AOFIS extracts rules which describe how the k output variables $y = (y_1, \dots, y_k)$ are influenced by the n input variables $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ 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.

B. 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, [1] 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 [1]. We used this technique to generate type-1 membership functions and then we added to each fuzzy set its footprint of uncertainty to generate an interval type-2 membership function.

C. 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 [9, 1] developed by L.X. Wang and by Mendel [6]. 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:

$$\text{IF } x_1 \text{ is } \tilde{A}_1^{(l)} \dots \text{ and } x_n \text{ is } \tilde{A}_n^{(l)}, \text{ THEN } y_1 \text{ is } \tilde{B}_1^{(l)} \dots \text{ and } y_k \text{ is } \tilde{B}_k^{(l)} \quad (2)$$

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

$$\mu_{\tilde{A}_s^{q^*}}^{cg}(x_s^{(t)}) \geq \mu_{\tilde{A}_s^q}^{cg}(x_s^{(t)}) \quad (3)$$

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 $x_s^{(l)}$ as follows [6]:

$$\mu_{\tilde{A}_s^q}^{cg}(x_s^{(l)}) = f_{x_s^{(l)}}^{cg}(\tilde{A}_s^q) = \frac{1}{2} \left[\underline{\mu}_{\tilde{A}_s^q}(x_s^{(l)}) + \overline{\mu}_{\tilde{A}_s^q}(x_s^{(l)}) \right] \quad (4)$$

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

$$IF \ x_1^l \text{ is } \tilde{A}_1^q \ \dots \ \text{and } x_n^l \text{ is } \tilde{A}_n^q, \ \text{THEN } y \text{ is centred at } y^{(l)} \quad (5)$$

For each input variable x_s there are V 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 (3),(5).

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

$$wi^{(l)} = \prod_{s=1}^n \mu_{\tilde{A}_s^q}^{cg}(x_s(t)) \quad (6)$$

The weight of a rule $wi^{(l)}$ is a measure of the strength of the points $x^{(l)}$ 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 (5). 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 are therefore divided into groups, with rules in each group sharing the same IF part. If we assume that there is M such groups. Let group l have N_l rules in the following form:

$$IF \ x_1 \text{ is } \tilde{A}_1^{(q^l)} \ \dots \ \text{and } x_n \text{ is } \tilde{A}_n^{(q^l)}, \ \text{THEN } y \text{ is centred at } y^{(t_u^l)} \quad (7)$$

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

$$av^{(l)} = \frac{\sum_{u=1}^{N_l} y^{(t_u^l)} wi^{(t_u^l)}}{\sum_{u=1}^{N_l} wi^{(t_u^l)}} \quad (8)$$

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

$$IF \ x_1 \text{ is } \tilde{A}_1^{(l)} \ \text{and } \dots \ \text{and } x_n \text{ is } \tilde{A}_n^{(l)}, \ \text{THEN } y \text{ is } \tilde{B}^{(l)} \quad (9)$$

Where the output fuzzy set $\tilde{B}^{(l)}$ is chosen based on the

following. Among the W output fuzzy sets $\tilde{B}^1, \dots, \tilde{B}^W$ find the \tilde{B}^{h^*} such that

$$\mu_{\tilde{B}^{h^*}}^{cg}(av^{(l)}) \geq \mu_{\tilde{B}^h}^{cg}(av^{(l)}) \quad (10)$$

for $h = 1, 2, \dots, W$, $\tilde{B}^{(l)}$ is chosen as \tilde{B}^{h^*} , where $\mu_{\tilde{B}^h}^{cg}(av^{(l)})$ is the centre of gravity of \tilde{B}^h at $av^{(l)}$ as in Equation (4).

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 (8) and (10) are repeated for each output value.

D. 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. Fig. 3 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 [3].

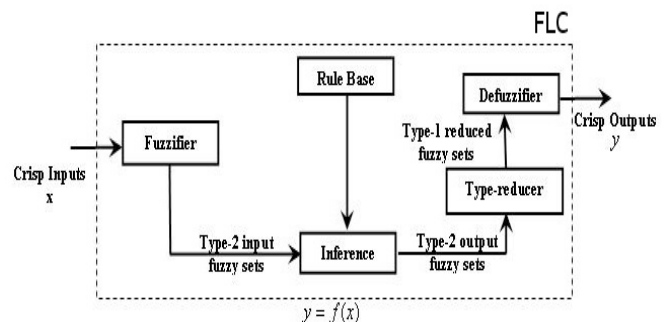


Fig. 3. Block diagram of a type-2 FLC

E. Online Adaptation and Life Long Learning

In the previous steps we have shown how our agent can learn an 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 [2] 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 product of the input membership function (degree of firing of the rule) in Equation (6) $wi^{(l)} > 0$, meaning that the rule fired, and would therefore have contributed to the overall control response generated by the agent's FLC. The consequent membership functions that give the highest membership values to the user defined actuator values are selected to replace the consequent sets of all fired rules in the rule base. The memberships are calculated as in equation (4) by calculating centre of gravity of the interval membership.

$$\mu_{\tilde{B}_c^{h^*}}(y_c) \geq \mu_{\tilde{B}_c^h}(y_c) \quad (11)$$

for $h = 1, 2, \dots, W$. 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.

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 $\mu_{\tilde{A}_s^{cg}}(x_s^{(t)}) > 0$ 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 (11). 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.

IV. EXPERIMENTS AND RESULTS

We have performed unique experiments in which a user lived in the iDorm for a total period of five days. During the monitoring phase which lasted for three consecutive days in late summer early autumn (early September), the agent recorded the user interactions with the environment. 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.

A. Offline Experiments

The data from the iDorm that was captured during the monitoring phase was initially 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 [1]. 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 [1]. The type-1 AOFIS had outperformed the ANFIS and 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 suitable for embedded agents. The other approaches cannot easily be adapted online as this would require their internal structures to be re-learned 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 mode in a non intrusive manner.

We then proceeded to determine if our type-2 AOFIS would produce a better performance than the type-1 AOFIS using the same data samples. The training instances in each data sample were used as before to generate the type-2 agent parameters. 7 type-1 sets were used to represent the input and output parameters of the type-1 agents as this was shown to be the optimum number of sets from the previous results. Five interval type-2 sets were empirically derived from the 7 type-1 sets for each parameter to form an interval type-2 FLC. The interval type-2 fuzzy sets therefore covered the same ranges as the type-1 fuzzy sets such that the type-1 sets were approximately embedded within the type-2 sets. It was found that the type-2 agent produced an average scaled error of 0.1255 and a scaled standard deviation of 0.1138. While the type-1 agent produced a scaled average error of 0.1324 and a scaled standard deviation of 0.1257. So the type-2 agent had produced smaller error (i.e. captures 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.

B. 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 uncertainties between the original user dataset and the current conditions would be far higher. In this way we could determine if the type-2 agent adapted better to the new environmental conditions than a traditional type-1 agent. The dataset accumulated during the monitoring 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 the user could override and adapt the agent's learnt control responses, if it was necessary to modify and tune them further. The agent could also autonomously add new rules to its rule base. The online performance of the agents could be measured by monitoring how well it adjusted the iDorm environment to the user's preferences such that the user intervention was reduced over time.

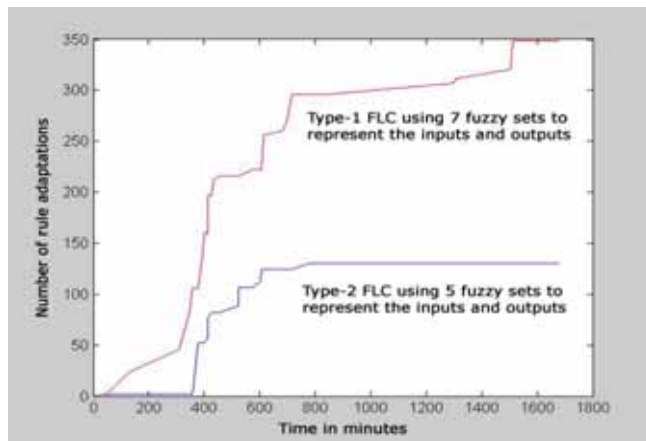


Fig. 4. Number of Online Rule Adaptations

Fig. 4 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 Fig. 4 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 UCE.

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.

III. CONCLUSION

In this paper we presented a novel system for learning and adapting type-2 fuzzy controllers for agents that can be embedded in UCE. Our agent learnt a 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 carried out unique experiments in which a user stayed in the iDorm for a total stay of five days. The offline and online performance of the type-2 agent showed that the 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 changed. The type-2 FLC had also used less number of rules than the type-1 FLC. The type-2 FLC was 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 data.

REFERENCES

- [1] F. Doctor, H. Hagrass, 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.
- [2] K. Ducatel, M. Bogdanowicz, F. Scapolo, J-C. Burgelman, "Scenarios For Ambient Intelligence in 2010," IPTS, Seville, Feb, 2001.
- [3] H. Hagrass, V. Callaghan, 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
- [4] H. Hagrass, M. Colley, V. Callaghan, G. Clark, 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.
- [5] Q. Liang, N. Karnik, 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.
- [6] J. M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice Hall PTR, Prentice Hall Inc, 2001.
- [7] J. Mendel, R. John, "Type-2 fuzzy sets made simple," IEEE on Transactions on Fuzzy Systems, Vol. 10, pp. 117-127, April 2002.
- [8] J. Mendel, 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.
- [9] L. X. Wang, 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.