Published in FuzzIEEE 2002 held at Hilton Hawaiian Village Hotel, Honolulu, May 12-17, 2002

A Fuzzy Incremental Synchronous Learning Technique for Embedded-Agents Learning and Control in Intelligent Inhabited Environments

Hani Hagras, Martin Colley, Victor Callaghan, Graham Clarke, Hakan Duman, Arran Holmes

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

Abstract: In this paper we introduce a novel learning and adaptation system for embedded-agents embodied and situated in intelligent inhabited environments. The Fuzzy Incremental Synchronous Learning (ISL) techniques we describe seek to provide an online, life-long, non-intrusive method for learning personalised behaviour and anticipatory adaptive control for physical environments.

I. INTRODUCTION

In the next 10-20 years the world will be populated with goods that contain small processors networked and equipped with artificial intelligence. The evidence for this comes from research programmes such as the EU's "Disappearing Computer", the USA's "Invisible Computer", and from industrial efforts such as Lonworks, EIB and Batibus systems. It is anticipated that all forms of goods will be influenced by this development from items that are clearly electronic in nature today (e.g. mobile phones, home entertainment systems, kitchen appliances, etc.) to those that are currently not (e.g. clothing, desks, etc.) [4]. It is argued that the arrival of such network enabled, embedded intelligent computer-based devices heralds an era of opportunities for both consumers and business. The challenge will be enabling the creation of *intelligent inhabited environments (IIE);* spaces such as cars, shopping malls, homes and even our own bodies, that will allow these artefacts to respond "thoughtfully" to peoples needs [4]. Precursors to such intelligent environments can be found in current intelligent buildings. The work proposed by this paper is part of a larger research agenda that addresses this challenge. Our contribution focuses on the investigation and development of techniques required to allow such environments to respond to user needs and to produce efficient and effective rules based on the embedded agents monitoring of user behaviour.

We outline a scenario for an "Intelligent Domestic Environment" based on the Intelligent Dormitory (*iDorm*) located at the University of Essex that will allow experimentation on intelligent environments.

II. THE NATURE OF IIE EMBEDDED AGENTS

Embedded intelligence can be regarded as the inclusion of some of the reasoning, planning and learning processes, typical of a person, within an artefact. Embedded-computers that contain such an intelligent capability are normally referred to as "*embedded-agents*" [3]. It is now common for such "*embedded-agents*" to have an Internet connection thereby facilitating multi embedded-agent systems. In a fully distributed multi embedded-agent systems, each agent is an autonomous entity co-operating by means of associations with its neighbours.

Most automation systems (which involve a minimum of intelligence) utilise mechanisms that generalise actions (e.g. set temperature or volume that is the average of many people's needs). However, we contend that AI applied to intelligent environments needs to *particularise* itself to the individual. Thus, the value of an intelligent autonomous embedded agent lies in the agent's ability to learn and predict the person and system needs, automatically adjusting the environment to meet their needs based on a wide set of parameters. Thus there is a need to modify effectors for environmental variables such as heat etc on the basis of a *complex multi dimensional input vector* which cannot be specified in advance. For example, something happening to one system (e.g. reducing light level) may cause a person to change behaviour (e.g. sit down) which in turn may result in them effecting other systems (e.g. needing more heat). An agent that only looks at heat levels is unable to take these wider issues into account. An added control difficulty is that people are essentiality non-deterministic and highly individual, therefore there is a need for system that particularises for individual users rather than generalising for a group of users. When viewed in such integrated control terms, it is possible to see why simple PID or fuzzy controllers are unable to deal satisfactorily with the problem of online learning for embedded agents. There are a number of research groups working in IIEs [2, 10, 11, 13] however to the authors knowledge none of them had dealt with online learning and particularisation, as in our research [6,8]. This work is also applicable to the development of intelligent gadgets and gadget-worlds which are the subject of our eGadgets Disappearing Computer EU project (No IST200-25240)

III. The Intelligent Dormitory (iDorm)

A typical domestic environment provides a setting where there is wide scope for utilising computer-based products to enhance living conditions. For instance it is possible to automate buildings service (e.g. lighting, heating etc), make use of computer based entertainment's systems (e.g. DVDs, TV etc), install service machines (e.g. robot vacuum cleaners, washing machines, cookers etc), or enhance peoples safety (e.g. security and emergency measures, appliance monitors etc). Some of these artefacts will be part of the building infrastructure and static in nature (e.g. lighting, HVAC etc.), others will be carried on the person as wearables or mobiles, or temporarily installed by people as they decorate their personal space (e.g. mobile phones, TVs etc). Environments in which computers are used to control building services are generally referred to as "Intelligent Buildings" [3].

We have chosen the Essex Intelligent Dormitory (iDorm) shown in Figure (1) to be a demonstrator and test-bed for some of the techniques involved in the development of intelligent environments and artefacts in general. Being an intelligent dormitory it is a multi-use space (i.e. contains areas with differing activities such as sleeping, working, entertaining etc) and can be compared in function to a room for elderly or disabled people or an intelligent hotel room. Because this room is of an experimental nature we are fitting it with a liberal placement of sensors (e.g. temp. sensors, presence detectors, system monitors etc) and effectors (e.g. door actuators, equipment switches etc), which the occupant can configure and use. The room looks like any other but above the ceiling and behind the walls hides a multitude of networks and networked devices.

Figure (1): Photograph of the iDorm

The iDorm is based around three networks, Lontalk, Tini 1-wire and IP. This provides a diverse infrastructure and allows the development of network independent solutions. It also gives us an opportunity to evaluate the merits of each network.

The embedded agent used for the intelligent buildings shown in Figure (2) is based on 68000 Motorola processor with 4 Mbyte of RAM and an Ethernet network connection. It runs the VxWorks Real Time Operating System (RTOS).

To create a standard interface to the iDorm we have an iDorm gateway server. This exchanges XML formatted queries with all the principal computing components which overcomes many of the practical problems of mixing networks. The communications architecture is being extended to allow devices to be 'Plug N Play' (enabling automatic discovery and configuration). The iDorm logical infrastructure is shown in Figure (3).

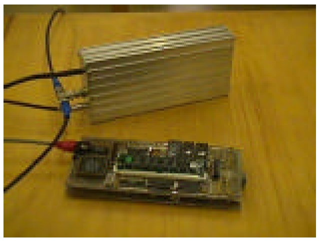

Figure (2): The Embedded Agent

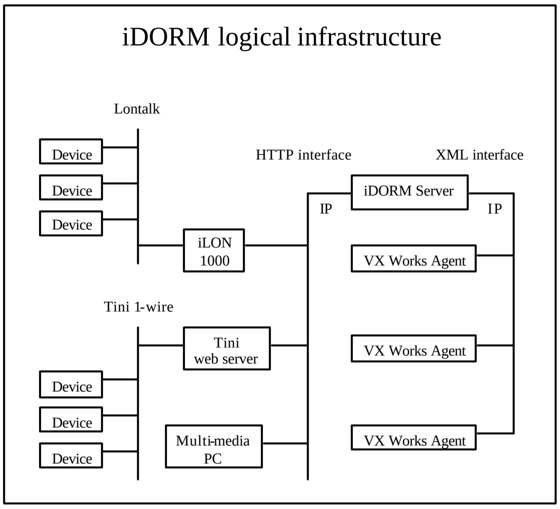

Figure (3): iDorm logical infrastructure

IV INCREMENTAL SYNCHRONOUS LEARNING

The Incremental Synchronous Learning (ISL) architecture is shown in Figure (4). The nature of the envisaged target environments means it needs to address to somewhat different learning needs, short "initialisation" and long "life-long" learning. In general a learning mechanism within a building would be non-intrusive.

The ISL forms the learning engine within the control architecture and is the subject of British patent 99-10539.7. The agent is an augmented behaviour based architecture which uses a set of parallel Fuzzy Logic Controllers (FLC), each forming a behaviour. The behaviours can be fixed, in the case of an Intelligent Building (IB) these could be - the Safety, Emergency and Economy behaviours or dynamic and adaptable such as, in an IB, Comfort behaviours (i.e. behaviours adapted according to the occupant's actual behaviour).

Each dynamic FLC has one parameter that can be modified which is the *Rule Base* (RB) for each behaviour. Also, at the high level the co-ordination parameters can be learnt [7,8]. Each behaviour is implemented as a FLC and uses a singleton fuzzifier, triangular membership functions, product inference, max-product composition and height defuzzification.

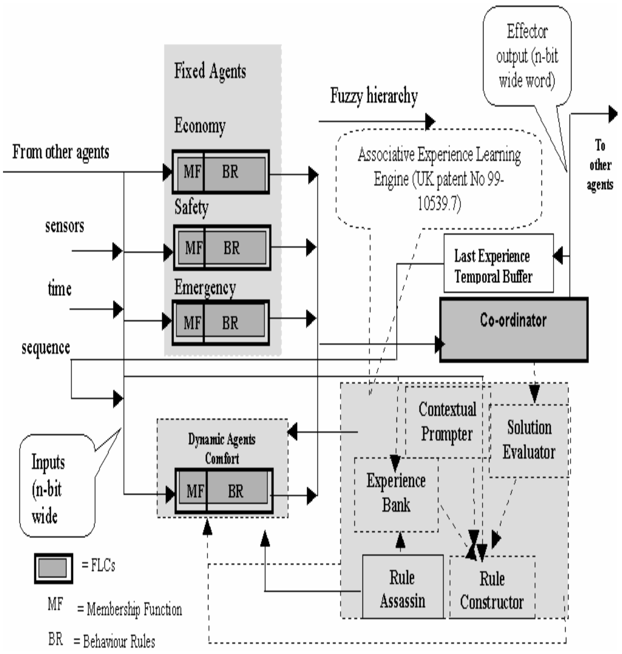

Figure (4): The ISL Embedded-Agent Architecture

The equation that maps the system input to output is given by:

$$Y_t = \frac{\sum_{p=1}^{M} y_p \prod_{i=1}^{G} \mu_{Aip}}{\sum_{p=1}^{M} \prod_{i=1}^{G} \mu_{Aip}} \qquad (1)$$

In this equation M is the total number of rules, y is the crisp output for each rule, $\prod \mu_{Ai}$ is the product of the membership functions of each rule input and G is the number of inputs. We use a higher level FLC to combine the preferences of different behaviour into a collective preference (giving a two-level behaviour hierarchy). In this model, command fusion is decomposed into two steps: preference combination and decision making and in the case of using fuzzy numbers for preferences, product-sum combination and height defuzzification. The final output equation is given by:

$$C = \frac{\sum_i (BW_i * C_i)}{\sum_i BW_i} \qquad (2)$$

Here i = economy, safety, comfort, .etc, C_i is the behaviour command output (e.g. room heat and temperature). BW_i is the behaviour weight. The behaviour weights are calculated dynamically taking into account the context of the agent. The agents receive their inputs from sensors. The output of each FLC is then fed to the actuators via the *Co-ordinator* that weights its effect. More information about the fuzzy hierarchical architecture can be found in [6,7,8].

The ISL system aims to provide life-long learning and adapts by adding modifying or deleting rules. It is also memory based in that it has a memory enabling the system to use its previous experiences (held as rules) to narrow down the search space and speed up learning.

We are at an early stage of this work and in our first experimental steps, reported below, we have constrained the environment to be one ISL agent and one occupant per room (multiple occupants may exist but not at the same time). The ISL works as follows:- when a new user enters the room he is identified and the ISL enters an initialisation Monitoring mode where it learns the users preferences during a non intrusive cycle. In the Experimental set-up we used a period of 30 minutes but in reality this is linked to how quickly and how complete we want the initial rule base. For example in a care setting we want this rule base to be as complete as possible with some fine tuning, in a hotel we want this initialisation period to be small to allow fast learning. The rules and preferences learnt during the initialisation Monitoring mode form the basis of the user rules which are retrieved whenever the user renters the room. During this time the system monitors the inputs and users action and tries to infer rules from the user monitored actions. The user will usually act when given an input vector the output vector is unsatisfactory to him. Learning is based on negative reinforcement, as the user will usually request a change to the environment when he is dissatisfied with it

After the Monitoring initialisation period the ISL enters a Control mode, in which it uses the rules learnt during the initialisation period to guide its control of the rooms effectors. Whenever the user behaviour changes, there is a need to modify, add or delete some of the rules in the rule base. In this event, the ISL suspends environmental control and goes back briefly to the non intrusive Monitoring cycle and infers the rule base changes necessary to determine the users new preferences (in relations to the specific components of the rule that has failed). This is a short cycle, transparent to the user, and distributed throughout the period of environment use, thus forming a life-long learning phase.

As in the case of classifier systems, in order to preserve the system performance the learning mechanism is allowed to replace a subset of the classifiers (the rules in this case). The worst m classifiers are replaced by the m new classifiers [5]. In our case we will change all the consequences of the rules whose consequences were unsatisfactory to the user. We find these rules by finding all the rules firing at this situation where ? ?Ai > 0. We replace these rules consequents by the fuzzy set that has the highest membership of the output membership function. We have done this replacement to achieve the non-intrusive learning and to avoid direct interaction with the user. The learnt consequent fuzzy rule set is guided by the *Contextual prompter* which uses the sensory input to guide the learning.

The crisp output Y_t can be written as in (1). If the agent has N output variables, then we have Y_{tN}. The normalised contribution of each rule p output (Y_{pN}) to the total output Y_{tN} can be denoted by S_{rN} and is given by:

$$\mathbf{S_{rN}} = \frac{\dfrac{Y_{pN} \, ?_{i?1}^{G} \, ?_{Aip}}{?_{p?1}^{M} \, ?_{i?1}^{G} \, ?_{Aip}}}{Y_{t1}} \qquad (3)$$

During the non-intrusive Monitoring and life-long learning phases the agent is introduced to different situations, such as having different temperature and lighting levels inside and outside the room with the agent guided by the occupants desires as it attempts to discover the rules needed in each situation. The learning system consists of learning different episodes; in each situation only small number of rules will be fired. The model to be learnt is small and so is the search space. The accent on local models implies the possibility of learning by focusing at each step on a small part of the search space only, thus reducing interaction among partial solutions. The interaction among local models, due to the intersection of neighbouring fuzzy sets means local learning reflects on global performance [1]. So we can have global results coming from the combination of local models, and smooth transition between close models. It is necessary to point to a significant difference in our method of classifying or managing rules which is rather than seeking to extract generalised rules we are trying to define particularised rules.

After the initialisation phase the system then tries to match the user derived rules to similar rules stored in the *Experience Bank* that were learnt from other occupiers. The system by doing this is trying to predict the rules that were not fired in the initialisation Monitoring session thus minimising the learning time as the search is starting from the closest rule rather than starting from random. Also this action will be satisfactory for the user as the system starts from a similar rule-base and fine tune the rules.

After this the agent will be operating with the rules learnt during the Monitoring initialisation session plus rules that are dealing with uncovered situations during the monitoring process which are ported from the rule base of the most similar user, all these rules are constructed by the *Rule Constructor*. The system then operates with this rule-base until the occupant's behaviour indicates that his needs have altered which is flagged by the *Solution Evaluator* (i.e. the agent is event-driven). The system can then add, modify or delete rules to satisfy the occupant by re-entering, briefly, the Monitoring mode. In this case again the system finds the firing rules and changes their consequence to the desired actions by the users. We also employ a mechanism - *learning inertia* - that only admits rules to the rule base when their use has exceeded some minimal frequency (we have used 3). One of our axioms is that "the user is king" by which we mean that an agent always executes the users instruction. In the case where commands are inconsistent with learned experience, learning inertia acts as a filter that only allows the rule-based to be altered when the new command is demonstrated by its frequent use to be a consistent intention. It is in this way that the system implements a life long learning strategy. It is worth noting that the system can start with a more intensive initialisation for a long time to learn as much needed rules which is highly needed in care houses. The relative initialisation versus life-long weighting can be set to suit particular applications.

The emphasis on particularisation over generalisation can create problems when the particularised rules storage needs of different users exceed the physical memory limits. This implies that we must remove some stored information. To manage this we employ various mechanisms. One such mechanism is we attach a *difficulty counter* to count the time taken by the agent to learn a particular rule base. We also attach a *frequency counter* to count how often this rule base has been retrieved. The *degree of importance* of each rule base cluster is calculated by the *Rule assassin* and is given by the product of the *frequency counter* and the *difficulty counter*. This approach tries to keep the rules that have required a lot of effort to learn (due to the difficulty of the situation) and also the rules that are used frequently. When there is no more room in the *Experience Bank*, the rule base cluster that had the least *degree of importance* is selected for removal by the *Rule assassin*. If two rule base clusters share the same importance degree, tie-breaking is resolved by a least-recently-used strategy; derived from a "life invocation" flag, that is updated each time a rule is activated.

Multi-Agent operation is supported by making compressed information available to the wider network. The compressed data takes the form of a status word describing which behaviours are active (and to what degree). As with any data, the processing agent decides for itself which information is relevant to any particular decision. Thus, multi-agent processing is implicit to this paradigm. We have found that receiving high level processed information from remote agents, such as "the room is occupied" is more useful than being given the low level sensor information from the remote agent that gave rise to the high-level characterisation. This is because the compressed form both relieves agent-processing overheads and reduces network loading.

V EXPERIMENTS AND RESULTS

We have conducted a number of experiments using the ISL to learn and adapt the behaviours of different users with different tastes and desires. For each experiment the users have spent up to 5 hours in the iDorm.

For the experiments the room had four environmental parameters to control;

? lighting level (I1) represented by 3 triangular fuzzy sets (Low Norm High)

? temperature (I2) represented by 3 triangular fuzzy sets (Low Norm High)

? outside level of lighting (I3) represented by three triangular fuzzy sets (Bright Dim Dark)

? user location represented by two fuzzy sets (I4) wither he is on the Desk or lying on the Bed.

There were seven outputs to control;

? two dimmable spot lights above the desk (O1) represented by five triangular fuzzy sets (VLow, Low, Norm, High, VHigh).

? two dimmable lights above the Bed (O2) again represented by five triangular fuzzy sets

? a Bedside Lamp (O3) represented by ON-OFF Fuzzy states;

? a Desk Lamp (O4) represented by ON-OFF Fuzzy states

? automatic blinds whose opening can be controlled (O5) represented by 5 triangular fuzzy sets, where the fuzzy sets VLow, Low, deal with blind opening to the left and VHigh, High deal with blind opening to the right and Norm is 50 % opening

? a Fan Heater (O6) represented by ON-OFF Fuzzy states and a Fan Cooler (O7) represented by ON-OFF Fuzzy states.

This forms a rule base of 3*3*3*2 =54 rules. However as the ISL learns only the rules required by the user we find that we don't need to learn the whole rule base, thus the rule base will be optimised.

Table (1) shows the learnt rule-base for "User-1" pictured in Figure (5) who occupied the room for more than two hours. In these experiments the users undertook many behaviours such as studying during the day (in which the lighting was bright) and studying in the evening (i.e. when external light was fading). This specific user preferred to use all the ceiling lights ON and the blind open to Norm. Another behaviour was lying in bed reading with the blind adjusted to his convenience and the bedside light being sometimes used. He would also close the blinds in the evening using combinations of the ceiling lamps and desk lamp. The data also contains "going to sleep behaviours" including reading before sleeping and getting up spontaneously at night to work (they are students!).

The ISL learnt 15 rules of which the first 7 rules were leamt during the Initialisation phase. The next 4 rules were ported from similar users and were satisfactory to the user. The last 4 rules resulted from fine tuning the ported rules. These rules dealt with darkness the first two rules dealt with the user wanting to sleep and he wanted all lights off while the similar user slept with the desk lamp ON because he doesn't like darkness. The last two rules dealt with user returning to the desk to read as he couldn't sleep he switches all lights to Norm and switch ON the desk and the bedside lamp and the blind to Norm to allow more light, the similar user had the same behaviour but he was closing the blind. It is obvious that the room user and the similar user actions are very similar and we needed only a fine tuning to satisfy the current user needs, this is an advantage of using the Experience Bank which reduces the life long learning time and satisfies the user. The XX in Table (1) indicate a No-Care situation, which resulted because the inside room lighting level was not important since when it was bright the user always took the same action. This shows also that ISL, besides optimising the rule base, can help to identify the input parameters required by the user and hence provide more optimisation to the system.

I1	I2	I3	I4	O1	O2	O3	O4	O5	O6	O7
XX	Norm	Bright	Desk	VHigh	VHigh	OFF	OFF	Norm	OFF	OFF
XX	High	Bright	Desk	VHigh	VHigh	OFF	OFF	Norm	OFF	OFF
High	Low	Dim	Desk	VHigh	VHigh	OFF	OFF	Norm	OFF	OFF
Norm	Norm	Bright	Bed	VHigh	VLow	ON	OFF	VLow	OFF	OFF
Norm	High	Bright	Bed	VHigh	VLow	ON	OFF	VLow	OFF	OFF
High	Norm	Bright	Bed	VHigh	VLow	ON	OFF	VLow	OFF	OFF
High	High	Bright	Bed	VHigh	VLow	ON	OFF	VLow	OFF	OFF
Norm	Norm	Dark	Desk	Norm	Norm	ON	ON	Norm	OFF	OFF
Norm	High	Dark	Desk	Norm	Norm	ON	ON	Norm	OFF	OFF
High	Norm	Dark	Desk	VHigh	VHigh	ON	OFF	VHigh	ON	OFF
High	High	Dark	Desk	VHigh	VHigh	ON	OFF	VHigh	ON	OFF
Low	Norm	Dark	Bed	VLow	VLow	OFF	OFF	VLow	OFF	OFF
Low	High	Dark	Bed	VLow	VLow	OFF	OFF	VLow	OFF	OFF
Low	Norm	Dark	Desk	Norm	Norm	ON	ON	Norm	OFF	OFF
Low	High	Dark	Desk	Norm	Norm	ON	ON	Norm	OFF	OFF

Table (1): The learnt Rule Base for "User-1"

Figure(5): “User 1 in the iDorm”.

Table (2) shows the learnt rule base for another user with different desires and taste. His Experiments were conducted at night giving rise to 12 learnt rules. 8 of which were generated during the Initialisation phase, 2 rules being ported from a similar user and were satisaforty to the user. The last 2 rules were tuned during the life long learning period as when sitting on a Desk and reading he wanted the desk light to be ON while for the similar user it was OFF so again it is a small fine tuning to satisfy the user which verifies the use of the Experience Bank.

Of course the learnt rules in both tables are different as our agent particularises rather than generalises. To adequately show all the characteristics of this agent we would need to run experiments over much longer periods (e.g. 12 months to get a full environment variation).

The memory management functions or wider performance issues can only properly be demonstrated when the memory bound was exceeded, requiring numerous rules generated over much longer periods of time. We intend to do such experimentation for a follow–on paper.

I1	I2	I3	I4	O1	O2	O3	O4	O5	O6	O7
Norm	Norm	Dark	Desk	VHigh	VHigh	ON	OFF	VHigh	OFF	OFF
Norm	High	Dark	Desk	VHigh	VHigh	ON	OFF	VHigh	OFF	OFF
High	Norm	Dark	Desk	VHigh	VHigh	ON	OFF	VHigh	OFF	OFF
High	High	Dark	Desk	VHigh	VHigh	ON	OFF	VHigh	OFF	OFF
Norm	Norm	Dark	Bed	VHigh	VHigh	ON	OFF	VHigh	OFF	OFF
Norm	High	Dark	Bed	VHigh	VHigh	ON	OFF	VHigh	OFF	OFF
High	Norm	Dark	Bed	VHigh	VHigh	OFF	ON	VHigh	OFF	OFF
High	High	Dark	Bed	VHigh	VHigh	ON	OFF	VHigh	OFF	OFF
Low	Norm	Dark	Bed	VHigh	VHigh	OFF	ON	VHigh	OFF	OFF
Low	High	Dark	Bed	VHigh	VHigh	OFF	ON	VHigh	OFF	OFF
Low	Norm	Dark	Desk	VHigh	VHigh	OFF	ON	VHigh	OFF	OFF
Low	High	Dark	Desk	VHigh	VHigh	OFF	ON	VHigh	OFF	OFF

Table (2): The learnt Rule Base of “User-2”

We have compared this method, which is an online proactive method, with other offline supervised learning systems such as the Mendel-Wang ANFIS approach [9]. We found, from the experimental results, that our system gives comparable results to that of such offline approaches. However our system has the added advantage of being on-line, adaptable to new users and able to particularise rather than generalise. Offline methods have to repeat the learning cycle from the beginning and require that the initial training set, plus any newly acquired data, be used. Our system works by cause-effect actions in the form of fuzzy rules, based on the occupant’s actions. The advantage of this is that the system responds and adapts to the users needs in an immediate manner by adding, deleting and modifying rules which cannot be achieved by off-line methods. More information about comparison with other techniques are listed in [6,8].

VI CONCLUSIONS

In this paper we have presented our Incremental Synchronous Learning (ISL) techniques for online learning and adaptation of embedded agents in intelligent inhabited environments. The techniques were evaluated in the Essex iDorm and, whilst at an early stage, have demonstrated the capability of the method to provide online learning in both short setup and life long learning cycles and particularising to the user desired action rather than generalising for a group of users. For our current and future work we have plans to conduct more and longer experiments with the iDorm (up to a year to get a full climate cycle), significantly expand the sensor-effector set and explore more fine-grained and course grained distributed embedded-agents (e.g. with agents in gadgets, or communicating rooms). We are also investigating the integration of mobile agents such as robots and wearable agents (e.g. cellphones, watches etc).

VII Acknowledgements

We are pleased to acknowledge the funding support from the EU IST Disappearing Computer program (eGadgets) and the joint UK-Korean Scientific fund.(cAgents) which has enabled much of this research to be undertaken.

VIII REFRENCES

[1] A. Bonarini, "Comparing Reinforcement Learning Algorithms Applied to Crisp and Fuzzy Learning Classifier systems", Proceedings of the Genetic and Evolutionary Computation Conference, pp. 52-60, Orlando, Florida, July, 1999.

[2] R. Brooks, “Intelligent Room Project”, Proc 2nd Int’l Cognitive Technology Conference (CT’97), Japan 1997.

[3] V.Callaghan,, G.Clarke, M.Colley , H.Hagras, “A Soft -Computing based DAI Architecture for Intelligent Buildings” in the book series entitled Studies in Fuzziness and Soft Computing , Springer Verlag, August 2000.

[4] V. Callaghan, M. Colley, G. Clarke, H. Hagras, “Embedding Intelligence: Research Issues for Ubiquitous Computing”, Proceedings of the Ubiquitous Computing in Domestic Environments conference, Nottingham, September 13 - 14, 2001

[5] M. Dorigo, M. Colombetti, "Robot Shaping: Developing Autonomous agents through learning", Artificial Intelligence Journal, Vol.71, pp. 321-370, 1995.

[6] H.Hagras, V.Callaghan, M.Colley, G.Clarke, “A Hierarchical Fuzzy Genetic Multi-Agent Architecture for Intelligent Buildings Sensing and Control ”, Proceedings of the International Conference on Recent Advances in Soft Computing 2000, pp. 199-206, 2000.

[7] H. Hagras, V. Callaghan, M. Colley, "Prototyping Design and Learning in Outdoor Mobile Robots operating in unstructured outdoor environments" IEEE International Robotics and Automation Magazine, Vol. 8, No.3, pp.53-69, Septemper 2001.

[8] H. Hagras, V. Callaghan, M.Colley, G. Clarke, "A Hierarchical Fuzzy Genetic Multi-Agent Architecture for Intelligent Buildings Learning, Adaptation and Control", to appear in the International Journal of Information Sciences, February 2002.

[9] J. Mendel, L.Wang, "Generating Fuzzy Rules by Learning Through Examples, IEEE Transactions on Systems, Man and Cybernetics", vol .2 pp. 1414-1427, 1992.

[10] N. Minar, M. Gray, O. Roup, R. Krikorian and P Maes. "HIVE: Distributed Agents for Networking Things". MIT Media Lab, Appeared in ASA/MA, 1999.

[11] M. Mozer "The Neural Network House: An Environment That Adapts To Its Inhabitants". In Proc of American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, pp. 110-114, AAAI Press, 1998.

[12] A. Saffiotti, "Fuzzy Logic in Autonomous Robotics: Behaviour Co-ordination", Proceedings of the 6th IEEE International Conference on Fuzzy Systems, Vol.1, pp. 573-578, Barcelona, Spain, 1997.

[13] A Sherwin. *Internet Home Offers a Life of Virtual Luxury.* The Times, pp. 10, 3rd Nov 1999.

