

# Achieving Human-Agent Teamwork In eHealth Based Pervasive Intelligent Environments



Matthew Ball, Vic Callaghan, Michael Gardner  
School of Computer Science and Electronic Engineering  
University of Essex, Colchester, UK  
mhball@essex.ac.uk

Dirk Trossen  
BT Research  
Adastral Park, Ipswich, UK

**Abstract-** Pervasive intelligent environments, such as digital homes, offer huge potentials for eHealth services. However, there is a question as to how best to manage (i.e. configure and program) these systems. The majority of solutions to-date rely exclusively on either end users or autonomous agents to manage the environment, which may not be suitable for all users and in certain situations, especially in eHealth applications where users may have very specific requirements that can change considerably over time. In this work-in-progress paper we argue that human-agent teamwork is vital for the adoption of future pervasive computing systems into our everyday lives; we discuss these issues, exposing the problem of exclusive management and we explore a solution, presenting our novel architecture for an adjustable autonomy agent that enables human-agent based teamwork. This work contributes to the PAL project: a collaborative research effort between University of Essex, BT, University of Cambridge and Ericsson.

**Keywords-** *intelligent environments; human-agent teamwork; pervasive eHealth; adjustable autonomy*

## I. INTRODUCTION

A pervasive intelligent environment can be seen as an everyday environment, such as a home or work place, which is enriched by a number of embedded computer-based networked devices controlled by a group of intelligent agents with the aim of enhancing user experiences and improving quality of life. Pervasive intelligent environments offer huge potential in the area of healthcare services: not only could they be used for health monitoring, but could also provide automated medical and assistive services to those who need it in their own home, thus reducing the need for trained healthcare workers to be present and allowing the patient to remain more independent. Providing this type of automation in such a personal space as the home is not an easy task. People have their own individual needs and preferences, which can differ greatly from each other and may change significantly over time. This even more true when dealing with people with medical conditions, for example: people with physical disabilities, those with deteriorating medical conditions and the elderly. Hence, many assistive or personal medical services will need to be tailored specifically to their user. Obviously, it would not be feasible to have an expert (or indeed a team of experts) develop, tailor and continuously maintain a specific system for each of a large number of users. Instead, two mainstream approaches to this management problem have emerged from recent research that

argue pervasive intelligent environments should be programmed and managed over time after deployment by embedded-agents, either autonomously or in direct dialogue with the end-user. Autonomous-agent driven systems have the advantage that they remove the cognitive load from the user, whilst end-user driven systems have the advantage that, unlike autonomous agents, the system is not required to guess the intentions and needs of the user. We, however, believe that a more ideal system would provide both options for management and enable human-agent teamwork.

This work forms part of the PAL (Personal and Social Communication Services for Health and Lifestyle Monitoring) research project (<http://www.palproject.org.uk>). PAL is a collaborative project between the University of Essex, BT, the University of Cambridge and Ericsson and is funded by UK's TSB and EPSRC science and engineering funding agencies. The main focus of PAL is to look at how future healthcare services impact current and future communication infrastructures. In this work-in-progress paper we report our contributions to the early stages of this project; we explore the problems of exclusively end-user and autonomous-agent driven management of pervasive intelligent environments, we introduce the concepts of digital territories, adjustable autonomy and mixed-initiative interaction, and we describe our approach to overcoming these problems which involves building an agent with adjustable autonomy and enabling human-agent teamwork based management. PAL is a three year project divided into three linked but distinct phases, namely design (and conceptual specification), implementation and evaluation. We have completed the design phase, and this paper reports on that element of our work. As a consequence to having investigated the conceptual models, an important aim of this paper is to aid future research by raising awareness of issues of pervasive intelligent environment management from the point of view of both users of the system and the system itself.

## II. THE MANAGEMENT PROBLEM

In this section we describe the two different approaches of end-user and autonomous-agent driven management systems. By management, we mean configuring (forming topographical connections) and programming (customising the functionality). We illustrate the problems by explaining how we are implementing these ideas in a current project and describe how

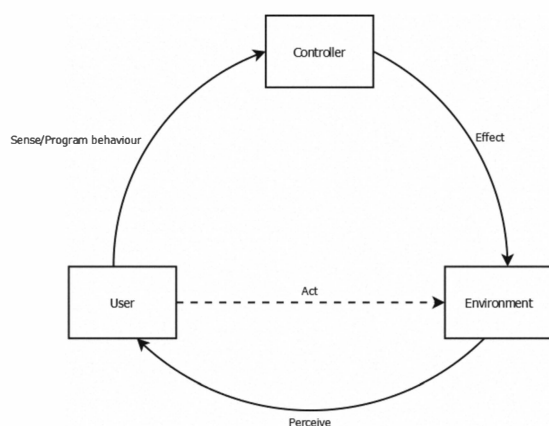


Figure 1. The management cycle

these ideas extend into the future vision of pervasive intelligent environments.

Figure 1 shows an abstract view of how a general pervasive intelligent environment is managed. It shows a controller (which could be a program, intelligent agents, etc.), the environment and a user; these are connected in a cycle, which we will refer to as the management cycle. The management cycle starts with the user acting in the environment based on their perceptions and preferences. These preferences are then captured, either implicitly through autonomous sensing or explicitly through end-user programming, by the controller, which drives the effectors and produces actions in the environment based what it has recorded about the user's preferences/requirements. The cycle is completed by the user perceiving the controllers actions in the environment and responding in some way (which in turn, leads to cumulative loops around the management cycle).

#### A. Approaches in Previous Research

As introduced earlier, in recent research there are two mainstream approaches to IE management. Firstly, an end-user driven approach can be taken. In this approach it is the responsibility of the end-user to program the IE although the user of the system may not actually have any knowledge of computer programming nor any technical knowledge of the system. An end-user driven system generally adopts a simplified programming mechanism to allow the user to understand and program the system a lot easier; for example, in Humble's jigsaw puzzle approach the pervasive intelligent environment is broken down into device components (and their corresponding functions), which are each represented by a virtual jigsaw puzzle-like piece that can be 'snapped together' in a chain from left to right by the user to describe their desired functionality of the system [1]. Another example of end-user driven systems is Alfred, a management system developed as part of the Intelligent Room Project at MIT [2]. Alfred allows the user to record macros using natural language processing and voice recognition technology. Macros can be played back by the user speaking a specified command phrase or using a pre-defined hardware trigger. Pervasive interactive Programming (PiP), developed at the University of Essex, uses a programming-by-example technique to enable end-user programming [3]. PiP breaks down the environment, in a

similar way to Humble's jigsaw puzzle approach, and allows the user to pick and mix from different functionalities of devices, grouping them together to form virtual pseudo-appliances known as MAPs (Meta -Appliances -Applications). PiP then allows the user to demonstrate the desired behaviour for any given MAP by interacting with either the physical environment itself or graphical representations of devices in the environment. The system can then generate a collection of rules (sets of actuating conditions and resulting actions) for each MAP based on these interactions of the user, effectively programming the environment. The second approach is to make the system autonomous. The system then programs itself by learning from the user's behaviours and interactions with the environment in context with the current environmental state and conditions. One example is the iDorm developed at the University of Essex [4]. The iDorm generates rules for system behaviour in a life-long learning mode by continuously monitoring and interpreting the user's interaction with devices with respect to the current state of the environment. After an initial learning period, the system runs in its 'normal operating' mode until the system detects that the user is unhappy with the current system behaviour (i.e. the user is overriding the system rules by manually operating devices in the environment). Another example of an autonomous agent driven system is Mozer's Adaptive Home, which uses reinforcement learning and a predictive neural network to control systems such as lighting, heating and ventilation in an attempt to reduce operational costs whilst still maintaining an acceptable level of user comfort [5].

End-user driven and autonomous-agent driven approaches can be seen as being at two opposite ends of a scale. An end-user driven approach empowers the user, giving them complete control in managing the system while, an autonomous-agent driven system disempowers the user handing complete control over to a collection of agents. In most situations, producing a system that empowers the user might seem the logical choice; an end-user driven approach not only encourages the creativity of the user but potentially makes the user feel a lot more at ease with being completely immersed within such a complex computer system. Nevertheless, some users may lack the ability or confidence to program an IE system, even with a simplified programming interface. Furthermore, users with severe medical conditions or physical disabilities may find it very difficult or even impossible to interact with the computer devices. In these situations an autonomous system is clearly the superior choice; it greatly reduces the cognitive and sometimes the physical load placed on the user in programming and managing the system. The 3C model shown in Figure 2 is a socio-agent framework that illustrates this concept [6, 7]. Each quadrant represents one extreme type of usage that may be encountered as a system becomes exclusively autonomous or end-user driven, given the user has a phobia (fear) or philia (love) of the system. Ideally we wish to avoid misuse and sabotage of the system and maximise creative use and symbiosis between the user and system. The approaches to IE management described in this section have all been either entirely end-user driven or entirely autonomous-agent driven and hence maximise either creative use or symbiosis but cannot provide both (with respect to the 3C model). Maximising creative use and empowering the user with an end-user driven

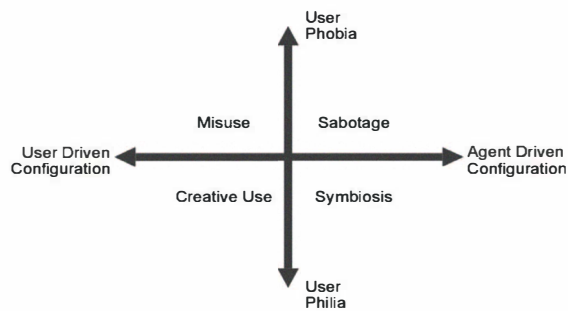


Figure 2. The Callaghan-Clarke-Chin (3C) model

system may bring about a user's phobia causing the user to misuse the system, albeit perhaps unintentionally. Conversely, trying to maximise symbiosis by providing an autonomous-agent driven system may also trigger a user's phobia and cause the user to sabotage the operations within the system, again albeit perhaps inadvertently.

### B. A Vision of the Future

As pervasive technology develops further, the distance between the physical and virtual/digital worlds grows ever smaller and a world in which we are fully immersed in pervasive computing systems becomes ever more a possibility. The advent of such a world will undoubtedly bring a change in our how we live our lives and how our societies operate. A multitude of data and services will be available to us wherever we go. This may be extremely beneficial but will also create many issues with privacy, especially with personal data [7]. Hakala and Beslay introduce a vision for this information-rich new world: digital territories [8]. People, groups and societies can define their own digital territory or 'bubble' that acts as a barrier used to govern the flow of information. The owner of the bubble can then choose which data is accessible to people outside of their bubble, protecting personal or sensitive data, and which data is allowed to enter their bubble, blocking out unnecessary or unwanted data. The notion of bubbles can be quite easily applied to the physical world as it is naturally separated into buildings, people and other objects each of which could be seen to as distinct digital territories each surrounded by a bubble.

If we imagine a pervasive intelligent environment management system that operates in this imaginary future world, a management system that is mobile and can control multiple environments, the same notion of Hakala and Beslay's bubbles could be applied to multiple pervasive intelligent environments, and the objects and people within them. Also taking from the bubble concept, some form of information governance would be absolutely necessary. However, our management system is interested in something more than just governance and control of information flow: it is interested with governance of interaction. Our system's primary concern is to manipulate the environment to suit the user's needs and preferences as much as is plausible, which naturally may include the control of information flow where the user is concerned with privacy. For this to be achieved our management system will require something greater than a barrier/gateway for information; it will require something more of a 'digital persona'. A persona can be defined as a specific

representation of oneself. People, in general terms, create different personas for themselves to suit different contexts of different social situations. For example, a person may have a very professional and serious persona at work, while at home their persona is a lot more relaxed. A persona can be seen as defining how we act around and with others and how we expect or allow others to interact with us in the context of a specific situation. Hence, a digital persona would define how we or some other digital entity (e.g. people, intelligent buildings and devices) represent ourselves digitally and how we interact with and allow others to interact with us. In terms of a single pervasive intelligent environment, a user's digital persona would consist of some abstraction of behaviour rules that the system has learnt or that have been programmed into the system. This abstraction could be seen as a driving concern for the rule. For example, a user is concerned with privacy of sensed data in his home and so a rule is created that all collected data is stored on a secure server. This concern will then be added into the user's digital persona and hence when the user moves to another environment the same concern can be applied to create a rule to the same effect. The concept of user concerns and using them to create a digital persona is discussed further in section III.C.

### C. An Exemplar Scenario

For the purpose of this study we have created the following scenario. Simon lives in a world enriched by ambient intelligence; many homes, work places and many public buildings and areas encompass intelligent pervasive computing systems that provide convenient services to its end users in some contextually suitable way. Simon finds this very useful as he is visually impaired and finds it very difficult to see in low-lighting conditions. In his digital home he has programmed a number of meta-appliances (an assembly of coordinating network services) that control his lighting in different situations. In his home he is easily able to program and configure these meta-appliances at his leisure; however, in other environments this task becomes a lot more complicated. For example, Simon is a student and regular visits to the library are vital to his study. Simon's library has a number of personal study rooms that are equipped with very similar technology to Simon's digital home and allow their occupant to control the internal conditions to suit their needs. Simon can use his portable smart-phone to interact with these rooms. However, since this a public space, he cannot program meta-appliances as he does in his digital home; instead, he relies on his smart-phone to autonomously select available meta-appliances and use them to try to recreate the effects of Simon's previously programmed meta-appliances by considering his concerns. For instance, Simon requires a high level of light whilst reading books. In his home his meta-appliance makes sure that the current light level is at least 650lux. In the library study room, Simon's smart-phone can recognise that he has a concern relating to lighting conditions whilst reading books and uses the available lighting devices in the room to increase the level of lighting to at least 650lux, satisfying Simon's concern.

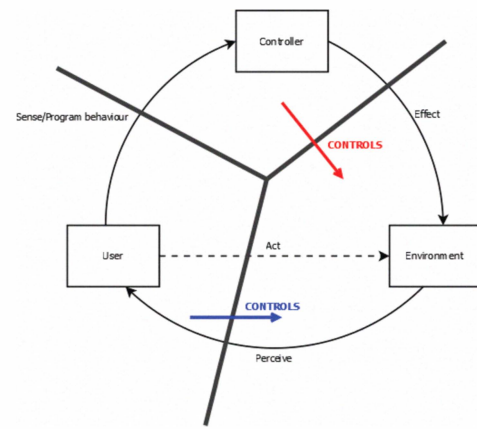
Thinking about how the environment is managed in this scenario, we see that whilst at home Simon can easily manage the lighting system himself using some form of end-user programming whereas, in a public space such as a library, he

has to rely on autonomous agents to manage the lighting system based on previously sensed information. Although this scenario may seem to describe a perhaps unrealistic and overly pedantic system for lighting, it does illustrate how in different pervasive intelligent environments management of the system may not only be changed because of user preference, but also it may be forced to change due to certain restrictions placed on the system. In the library scenario Simon is not allowed to control the lighting directly but must do it via agents. Another scenario might entail an autonomous system that, due to restrictions in the system (perhaps a lack of available sensors), might be forced to ask the user how it should act as it cannot grasp enough information to make a rational decision autonomously. moderately

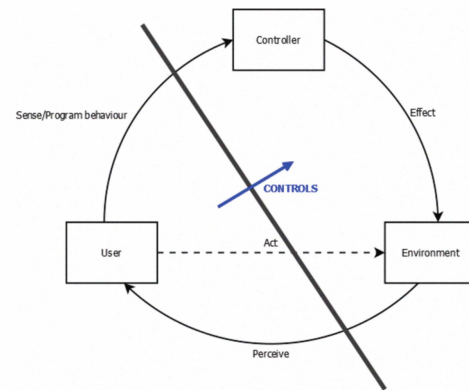
*D. Summarising the management problem*

Figures 3(a) and 3(b) depict the problems with fully autonomous-agent and fully end- user driven systems using the same abstract view of a pervasive intelligent environment as in Figure 1. Figure 3(a) illustrates how the management system becomes separated in a fully autonomous-agent driven system. The user and the controller are separated, they are operating separately although they are effectively working towards the same task – control the environment to suit the user’s needs and preferences. This working pattern could indeed result in the user and controller operating together as a team; however, it could also result in the user and controller working in opposition to each other. This is the cause for the problems we have previously described for autonomous systems. In a fully autonomous management system there is no direct communication between the user and controller; no matter how much intelligence is put into the controller, at some point it has to rely on guesswork to assume the user’s needs and preferences and if the controller guesses wrongly the user cannot directly tell the controller that it is wrong or exactly why this is the wrong action to take, which again leads to more guesswork. Furthermore, if the management system is operating in a restricted environment, it may not be able to get enough information to make a rational guess or decision. Any system that relies on guesswork will inevitably guess wrongly at some point, which could be highly annoying to the user or indeed completely unacceptable.

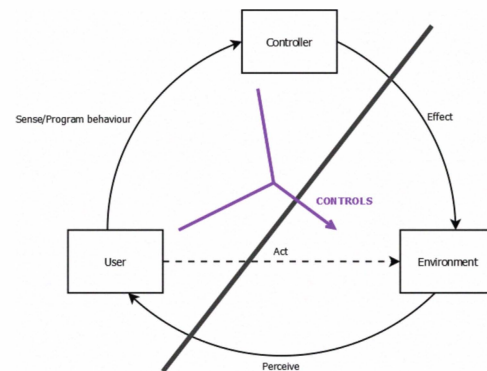
Figure 3(b) illustrates how the management system is separated in a fully end-user driven pervasive intelligent environment. Here the environment and the controller effectively become one entity; the controller becomes little more than an interface to the environment and hence the pervasive intelligent environment relies completely on the user to give it intelligence. Without a user to program the controller the environment would not know how to adapt to new situations and would remain static. This is the reason for the problems described previously for fully end-user driven system; the intelligence of the management system is dependent on the creativity, intelligence, willingness and ability of the user in programming the system. Such a complex system cannot rely on all users in all different situations to be creative, intelligent, willing and able enough to manage the system; it is simply not feasible in all situations for all users, even if the method of programming is greatly simplified.



(a). Separation in an autonomous system



(b). Separation in a end-user driven system



(b). Separation in a human-agent teamwork based system

Figure 3. Illustration of problems of exclusive management

Figure 3(c), again based on Figure 1, depicts our proposed solution to these problems. We seek to combine these two distinct approaches and create a system in which the user and controller (as an agent) work together as a team. If the user and controller can collaborate together, this reduces the chance of guesswork needing to be done by the controller and if either the user or controller cannot, for whatever reason, manage the system in the usual way, they can seek help from the other. This creates an overall more robust management system. Also the user is no longer forced to either manage and program the

environment themselves or be at the complete mercy of autonomous agents, instead they can manage the environment at level at which they feel comfortable doing so. Our proposed solution is discussed in more detail in the next section.

### III. TOWARDS A SOLUTION

In section II we have discussed the problems of the current approaches to pervasive intelligent environment management. Based on this discussion we have defined the following criteria that we believe a successful pervasive intelligent environment management system should meet:

1. A pervasive intelligent environment should allow for human-agent teamwork based management.
2. The user should be able to provide as much or as little initiative in the teamwork based management as they wish or are able to.
3. The management system should describe a user's digital persona, which should define how a user wishes to interact with other digital entities in differing digital territories.
4. The management system should be mobile and be able to control different pervasive environments to the same effect wherever possible based on the user's digital persona.

Two very important concepts we require in order to achieve these four criteria are adjustable autonomy and mixed-initiative interaction. In subsection A we describe these concepts and how they relate to pervasive intelligent environments. In subsection B we discuss our approach to using these concepts to create an adjustable autonomy agent and enable human-agent teamwork based management. In subsection C we describe the next stages of our project looking at the creation of a digital persona and the abstraction of behaviour rules into concerns. Finally, in subsection D, we describe the test-beds in which our research will be carried out: the iSpace and iCampus at University of Essex.

#### A. Adjustable Autonomy and Mixed-Initiative Interaction

Bradshaw describes adjustable autonomy as maintaining "the system being governed at a sweet spot between convenience (i.e. being able to delegate every bit of an actor's work to the system) and comfort (i.e. the desire to not delegate to the system what it can't be trusted to perform adequately)" [9]. By altering the level of autonomy in certain ways, we allow for mixed-initiative interaction. Bradshaw also presents us with an interesting vacuum cleaner analogy to explain mixed-initiative interaction [9]. The most manual (non-autonomous) is a 'plain old' vacuum cleaner. It is directly operated by a person's arm, apart from the ongoing sweeping and sucking action of the motor, every action is taken at the initiative and direction of the user. The opposite, a fully-autonomous vacuum cleaner, would turn itself on, vacuum until it decides it's finished and then retreat back to its storage place to recharge. With this vacuum cleaner no initiative or direction is required from the user; it relies only on the

initiative of agents. From this it is easy to imagine an example of a mixed-initiative vacuum cleaner; a vacuum cleaner that requires the user to switch it on, place it in a starting position and instruct it to start vacuuming. The vacuum cleaner would then autonomously move about the room ensuring every reachable spot gets cleaned and, when finished, return back to its starting position and wait to be turned off. Such a vacuum cleaner relies upon initiative and direction from both the user and agents; it relies on mixed-initiative interaction. Mixed-initiative interaction can be achieved at varying levels. At one level users can work together with agents in a very intricate way, as in Allen and Ferguson's human-machine collaborative planning system [10]. In this system the user and the agent collaborate through a conversational dialogue in order to produce an evacuation plan for a natural disaster on an island. The user-agent collaboration is somewhat similar to the way two humans might devise a plan through a back-and-forth natural language conversation. Another possible level of human-agent interaction is through task delegation. Research at NASA has explored how life support systems and scientific equipment can be made autonomous whilst still giving the user ultimate control when needed by allowing the user to adopt or be delegated certain tasks [11]. One example presented to us is a Mars rover. Given a rover that is traversing the planet's surface autonomously, a crewmember can take control of certain parts of the rover's system to undertake certain tasks if some unanticipated event occurs. For example, a crewmember may wish to take control of the rover's visual scanning systems if the rover stumbles across something interesting. In terms of pervasive intelligent environments, both of these levels of mixed-initiative interaction could be very useful in producing a hybrid autonomous-agent/end-user driven system. At the collaborative level, agents and users could collaborate together to program behaviour rules or, moreover, to resolve conflicts in the system. The task delegation/adoption level could be used to allow the user to adopt certain programming tasks themselves while delegating others to agents. It is clear that these concepts could be applied to a pervasive intelligent environment management system; the next subsection discusses our approach to doing so.

#### B. Creating an Adjustable Autonomy Agent

Figure 4 shows a high-level architecture that allows for adjustable autonomy and enables the user to collaborate in the creation of rules to manage a pervasive intelligent environment. It is inspired by the incremental synchronous learning (ISL) agent developed by Hagrais et al. [12]. The architecture takes the general form of a behaviour-based architecture as pioneered by Brooks at MIT [13]. In such architectures a number of agent behaviours run in parallel. A controller is employed to coordinate the behaviours or their given outputs into one single output to achieve the desired agent functionality. As with the ISL agent, our agent learns fuzzy rules to produce a set of fuzzy controllers used as behaviours. A higher-level fuzzy logic controller is then employed to coordinate the output of the behaviours. This enables our agent to control the environment based on the learnt rules. The ISL agent is fully autonomous; our architecture differs from this, however, as it enables adjustable autonomy and user collaboration in producing rules. It does this by having two sets of behaviour rules: one active

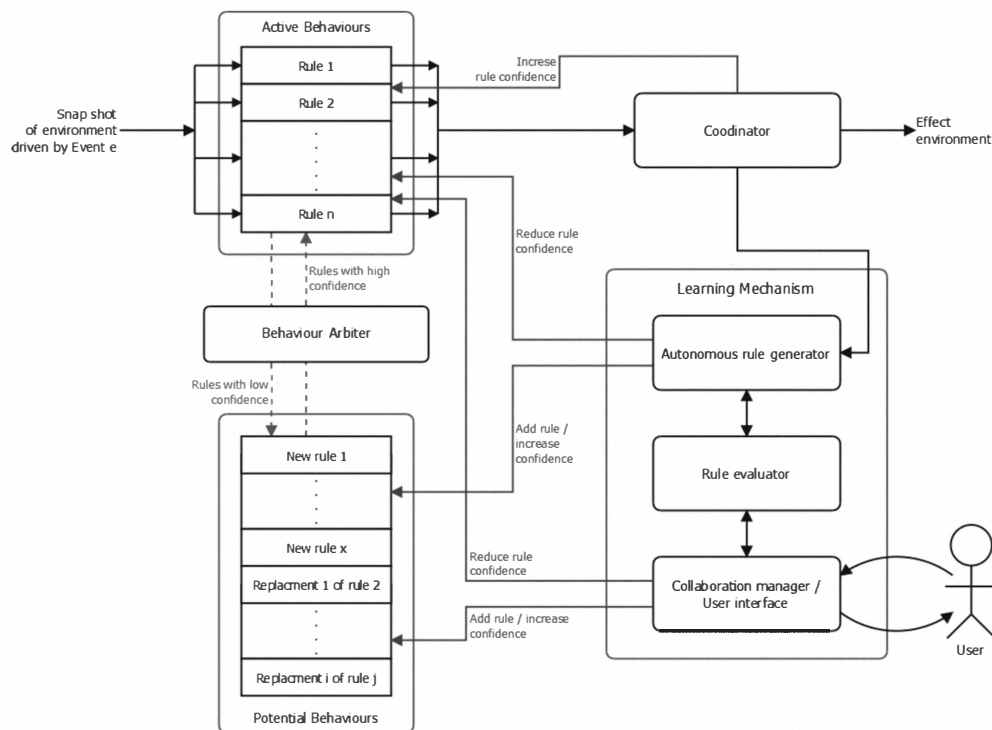


Figure 4. Adjustable autonomy management agent

set and one potential set. Each behaviour rule is assigned with a confidence level. A rule can only have an effect on the environment if it is active and can only be active if it has a high enough confidence level. Rules with a low confidence can only be potential behaviour rules and cannot effect the environment. In our architecture all behaviour rules are visible to all components of the agent. The behaviour arbitrator regulates the behaviour rules. Overtime the behaviour arbitrator will reduce the confidence of all behaviour rules. If an active rule's confidence level drops below a certain threshold, it will drop down into the potential set and if a potential rule's confidence level drops below a very low threshold (zero for example) then it is deleted. This confidence degradation reduces the chance that the agent's memory will become full. If, however, the agent's memory does become full, a number of rules with the lowest confidence can be removed as necessary. The coordinator regulates/merges the output of all the active behaviours into one single output so that each behaviour rule effects the external environment to an appropriate degree. Additionally, when an active behaviour rule effects the environment the coordinator increases the level of confidence of that rule, where the amount increased depends on the degree the rule is effecting the environment. Thus, the more a rule is used and the more it effects the environment, the less chance it will have of dropping into the potential set and ceasing to be active. If the user or agent senses the need for a new rule or an alteration to an existing rule, they can create a new potential rule (as an entirely new rule or a replacement for an existing active rule) or increase the confidence level of an existing potential rule. If a potential rule, that is intended to replace an existing active rule, is created or an existing one is given more confidence, then the confidence level of the active

rule it intended to replace is reduced. The amount of confidence the user or agent can give to a rule, new or existing, is restricted based on the selected level of autonomy. With a lower level of autonomy the user is required to give a potential behaviour rule confidence before it can become an active behaviour rule and the amount of confidence the agent can assign to the rule is limited to a low level. With a higher level of autonomy the agent is able to assign more confidence to a rule and less is required from the user. Of course, in a fully autonomous-agent or end-user driven system, either the agent or user can assign full confidence to a potential behaviour rule while the other is not expected or allowed to assign any. Through this confidence based mechanism we can force the agent to collaborate and hence alter its level of autonomy.

We previously discussed two ways which we could use mixed-initiative interaction to achieve human-agent teamwork: task delegation and direct collaboration. By changing the level of autonomy by explicitly restricting its confidence in its rules we can create a style of task delegation in which the user can choose to adopt or delegate a task to a certain degree. One can imagine a sliding-scale switch (similar to a volume control) that represents the level of autonomy of management: one end of the scale represents fully autonomous-agent driven management and the opposite end represents exclusively end-user driven. The user could use this theoretical sliding scale switch to explicitly say to what degree they wish to contribute to a given task, i.e. to what degree the management should be done autonomously. To further this analogy, one could imagine the pervasive intelligent environment being divided up into sub-systems, for example heating, lighting and security. Behaviour rules could then be categorised into these sub-systems; thus allowing us to create a theoretical mixing desk

consisting of many sliding-scale switches to control the level of autonomy throughout the entire system. Direct collaboration could also be used in the system to ‘fine tune’ the level of autonomy further. Perhaps the simplest form of collaboration in our system would be for the agent to develop a rule and present it to the user for them to accept or reject it. Here we can say the system is highly autonomous as it requires a lot of initiative from the agent and little from the user. In order to reduce autonomy further, we must increase the level of initiative from the user. This can be done by giving the user the option of altering the rule before accepting it. The same choices could also be offered to the agent if a user has created a rule in a more end-user driven system or if a user suggests an alteration to a rule generated by the agent. Hence, we can create a kind of back-and-forth communication in a way that is quite similar to Allen and Ferguson’s collaborative planning system [10]. Each participant will be able to suggest new rules and accept, reject, or alter the others suggested rules. This collaborative mechanism will allow either participant to provide varying levels of initiative to the collaboration; the more a participant makes suggestions for new rules and alterations to the other’s suggested rules, the more initiative they provide. Changing the way that the collaborative mechanism is triggered will help to adjust the level of the agent’s autonomy. For example, in a more autonomous system, when an agent generated potential rule’s confidence level has reached a high enough level, say 75% (i.e. when the agent has seen the user repeat the same action a number of times in the environment), the remaining 25% confidence will have to be gotten from the user before the potential rule can become active and the agent will start a collaboration. Here, since the system requires an input from the user, we can say that the agent is no longer fully autonomous because of an explicit restriction of its confidence in its own generated rules. The agent’s level of autonomy is then adjusted further implicitly in the collaboration depending on the level of initiative provided by the user; if the user simply accepts the potential rule we can say the agent operated with a higher level of autonomy than if the user makes an alteration to the agent generated rule. In this way we provide an agent with adjustable autonomy, in which the level of autonomy is adjusted in two ways: firstly an initial level is explicitly set by the user (task delegation) and the level of autonomy can then be adjusted further implicitly through direct collaboration.

### C. Creating a Digital Persona

In the next stages of our project we intend to investigate how we can create a digital persona for the user, such that the user can carry it around with them, using a mobile device, and use it to manage and interact with multiple pervasive intelligent environments (digital territories). In terms of our project, a persona is equivalent to a group of rules (preferences) linked to a context (a digital territory). Achieving this goal will add a number of complications to how the system will be implemented. Most notably, since we are dealing with a mobile device, we have a very limited memory size and processing power. This is only exacerbated by the fact that we are now dealing with multiple environments, which could be very distinct from each other and each require many sets of behaviour rules of their own, if we were to use a management

system as described thus far. Another issue we will explore is how to sense and classify the differing contexts (different digital territories). In this, some rules will be generic to all contexts, others will be particularised to specific contexts. It is clear that a digital persona will not be able to contain all rules for all different contexts and still be mobile between environments; hence, it will require some form of abstraction. One form of abstraction we are investigating is user concerns.

A concern – A matter that engages a person’s attention, interest or care; or that affects a person’s welfare or happiness ([www.dictionary.com](http://www.dictionary.com)). In terms of a pervasive intelligent environment a concern is any preference, desire, or worry that a user has about some aspect of the system. It differs to a preference in that also incorporates behaviours that a user wants to prevent rather than just behaviours they wish to happen. The major role of a pervasive intelligent environment management system, we could say, is to satisfy desires and prevent worries or, in other words, to adapt the environment in order to resolve any concerns that the user might have. For example, in an intelligent apartment, the user may be concerned with the temperature being too low in the evenings or the user may be concerned with privacy due to the invasiveness of the multitude of sensors installed in the system. Realising this vision of a digital persona using concerns will require much further work. It is easy to picture how any action taken or event triggered by a person results from some concern, be it a simple preference or safety restriction; however, the question remains how it is possible for the system to capture these concerns? Moreover, we must consider how this can be done using end-user driven techniques, autonomously with agents as well as collaboratively in order to keep the level of autonomy adjustable throughout the management system.

### D. The Essex iSpace and iCampus

The test-beds we are using to evaluate these ideas are a fixed environment, the iSpace (an instrumented digital home) and a mobile environment, the iCampus (an instrumented university campus). The iSpace, shown in Figure 5, is a purpose built test-bed designed to aid in research of pervasive computing, ambient intelligence and human-computer interaction [7]. As well as everything one might expect to find in any other two-bed apartment, the iSpace is also equipped with a multitude of networked sensors and actuators, e.g. internal and external temperature and lighting sensors, real-time location tracking, automated heating and lighting, and electronically controlled doors. Built with cavity walls and ceilings containing power and network outlets, it allows researchers to deploy their experimental systems for testing on real human users in an unobtrusive way. All services and devices in the iSpace are networked and controlled electronically wherever possible using an underlying Universal Plug & Play (UPnP) based architecture, making them easily accessible to experimental systems. The iSpace mimics a home from our vision of the future, similar to Simon’s digital home in the previously described scenario (section II.C). We will deploy our adjustable autonomy management system in the iSpace to test our system initially within a single environment and then extend our work to produce a mobile system that can operate across multiple pervasive intelligent environments across the iCampus, a research network that



Figure 5. The Essex iSpace

spans over the university campus combining wireless, wired and fibre networking technologies. Using WiFi and WiMAX, the iCampus creates a wireless mobile environment in which mobile pervasive technologies can be tested. Within the iSpace a UPnP enabled real-time location system (RTLS) has been deployed, which can accurately locate mobile tags using ultra-wideband radio in different zones throughout the iSpace. Via the UPnP middleware, the RTLS can be conjoined with other applications and devices to implement services such as follow-me video and automated lighting. Services such as these give a simple operational example of digital territories as described in our future vision and scenario (sections II.B); as the user moves between different zones in the iSpace the service has to change its behaviour accordingly. This concept mirrors our mobile management system; when moving between different digital territories it has to take into account the different concerns in the different contexts and adopt new rules or adjust existing to change its behaviour accordingly. Moreover, this concept could be extended to the iCampus to simulate digital territories for experimentation. Defining and implementing the test-beds has contributed to the specification of the exemplary scenario and will be especially valuable in the evaluation phase, where we plan to conduct trails using members of the public staying in the iSpace.

#### IV. CONCLUSION

The vast majority of previously researched pervasive intelligent environment management systems have taken either an exclusively autonomous-agent or end-user driven approach. Although a lot of these research efforts have been fruitful, taking such an exclusive approach will undoubtedly cause problems for some users and certain situations as pervasive technology develops further. Hence, we argue that future management systems of eHealth based pervasive intelligent environments must allow for human-agent teamwork and adjustable autonomy. This will be especially important for users with very specific needs that can change unexpectedly overtime, such as those with physical disabilities and deteriorating medical conditions; a system with the extra dynamic of adjustable autonomy will enable a pervasive healthcare system to be highly customisable and allow the level of initiative and effort the system expects from the user to be changeable initially and overtime. This work forms part of the PAL project, a collaborative research project between the University of Essex, BT, the University of Cambridge and Ericsson and is funded by UK's TSB and EPSRC science and engineering funding agencies. The PAL project is scheduled

to take 3 years to complete and is divided into three distinct phases the first of these being an exploration and definition of the concepts. This work-in-progress paper has reported on this first phase. In particular we have discussed these issues and provided a clear description of the problems of allowing only exclusive management of pervasive computing systems by either end users or autonomous agents. We have also introduced the concepts of digital territories, adjustable autonomy and mixed-initiative interaction, and presented a novel architecture for an adjustable autonomy agent that enables human-agent teamwork based management. An important aim of this paper is to aid future research by raising awareness of issues of pervasive intelligent environment management from the point of view of both users of the system and the system itself. We believe that this not only contributes to research in pervasive computing but it also useful in other areas of artificial intelligence and human computer interaction where human-agent teamwork could be a factor. We look forward to presenting all significant progress of this ongoing work at subsequent pervasive healthcare conferences.

#### REFERENCES

- [1] Humble, J., Crabtree, A., Hemmings, T., Åkesson, K.P., Koleva, B., Rodden, T., Hansson, P., "Playing with the Bits", User-Configuration of Ubiquitous Domestic Environments", *Proceedings of UbiComp 2003*, Springer-Verlag (2003), 256-263.
- [2] Gajos, K., Fox, H., & Shrobe, H., "End user empowerment in human centered pervasive computing", in *Proceedings of Pervasive 2002*, (2002), 1-7.
- [3] Chin, J., Callaghan, V., Clarke, G., "An End User Tool for Customising Personal Spaces in Ubiquitous Computing Environments", *Lecture Notes in Computer Science: Ubiquitous Intelligence and Computing*, Springer-Verlag (2006), 1080-1089.
- [4] Hagrais, H., Callaghan, V., Colley, M., Clarke, G., Pounds-Cornish, A., Duman, H., "Creating an Ambient-Intelligence Environment Using Embedded Agents", *IEEE Intelligent Systems*, Vol.19, No.6, (2004), 12-20.
- [5] Mozer, M.C., "Lessons from an adaptive home", *Smart Environments: Technology, Protocols and Applications*, Wiley (2005), 273-298.
- [6] Chin, J., Callaghan, V., Clarke, G., "Soft-appliances: A vision for user created networked appliances in digital homes", *Journal of Ambient Intelligence and Smart Environments 1*, (2009), 69-75.
- [7] Callaghan, V., Clarke, G.S., and Chin, S.J.Y., "Some Socio-Technical Aspects Of Intelligent Buildings and Pervasive Computing Research", *Intelligent Buildings International Journal, Earthscan*, 1:1, (2008).
- [8] Beslay, L., Hakala, H.: Digital territory: Bubbles. In Wejchert, J., ed.: *The Vision Book*, Brussel (2005)
- [9] Bradshaw, J.M., Feltovich P.J., Jung, H., Kulkarni, S., Taysom, W., Uszok A., "Dimensions of adjustable autonomy and mixed-initiative interaction", in *Agents and Computational Autonomy: Potential, Risks, and Solutions*, (M. Nickles, M. Rovatos, and G. Weiss, Ed.), Springer-Verlag (2004), 17-39.
- [10] Allen, J., Ferguson, G., "Human-Machine Collaborative Planning", in *Proceedings of the NASA Planning and Scheduling Workshop*, (2002)
- [11] Dorais, G. A., Bonasso, R. P., Kortenkamp, D., Pell, B., and Schreckenghost, D., "Adjustable autonomy for human-centered autonomous systems on Mars", In *Proc. Of the First International Conference of the Mars Society*, (1998).
- [12] Hagrais, H., Colley, M., Callaghan, V., Clark, G., Duman, H. and Holmes, A., "A fuzzy incremental synchronous learning technique for embedded-agents learning and control in intelligent inhabited environments," in *Proc. IEEE Int. Conf. Fuzzy Syst., HI*, 2002, pp. 139-145.
- [13] Brooks, R., "Intelligence Without Representation", in *Artificial Intelligence*, (1991) 47:139-159.