University of Essex

Department of Computer

INHABITED INTELLIGENT ENVIRONMENTS GROUP

TECHNICAL REPORT

CSM 513

Victor Callaghan,
Graham Clarke,
Martin Colley,
Hani Hagras,
Anthony Pounds-Cornish,
Arran Holmes,
Hakan Duman,
Jeannette Chin.

APRIL 2002

Towards the Realisation of Ambient Intelligence: An Embedded-Agent Approach to Ubiquitous Computing

Victor Callaghan, Graham Clarke, Martin Colley, Hani Hagras,
Anthony Pounds-Cornish, Arran Holmes,, Hakan Duman, Jeannette Chin.

Intelligent Inhabited Environments Group
Essex University
vic@essex.ac.uk

Abstract. In this paper we outline a vision, methodology and architectural design for creating an ambient environmental intelligence composed of a ubiquitous computing environment made up of scaleable, networked agent-based artefacts, which has arisen from our work in the EU's Disappearing Computer Initiative (DCI) eGadgets project. We discuss the research challenges involved, particularly those relating to embedding intelligent agents into artefacts, and include a prototype design of one such artefact based agent. We describe a practical ubiquitous computing test-bed environment (the iDorm) which combines a number of artefact based agents, several different networks and provides a common protocol to act as a gateway between the different sensors and effectors. In addition we introduce a web/wap based iDorm emulation and visualisation system that acts as an interface to this ubiquitous computing environment, allowing users to control and monitor the agent-based artefacts.

1. Introduction

1.1. The Vision

Perhaps nobody has articulated the vision for ubiquitous computing better than Mark Weiser who in 1994 said "For thirty years most interface design, and most computer design, has headed down the path of the 'dramatic' machine. Its highest ideal is to make a computer so exciting, so wonderful, so interesting that we never want to be without it. A less travelled road I call the "invisible"; its highest ideal is to make a computer so embedded, so fitting, so natural, that we use it without even thinking about it.". Only eight years since this incisive statement was made the variety of computer-based artefacts, and their capabilities, are growing at an unprecedented rate fuelled by advances in microelectronics and Internet technology. Recent figures presented by Robert Metcalfe (Ethernet inventor and 3Com Corp founder) at last years ACM1 conference suggested that some 8 billion microprocessors were produced in 2001, with only 2% of them going into PCs. He suggested that most ended up as part of the all pervasive fabric of computing that he described as being woven around and through our lives via a wide range of devices, some of which we don't even recognize as computers. Clearly people's domestic spaces are becoming increasingly "decorated" by electronic or computer based artefacts varying from, mobile telephones through CD players to transport systems and beyond. Cheap and compact microelectronics means most everyday artefacts (e.g. clothing, desks) are now potential targets of embedded-computers, while ever-pervasive networks will allow such artefacts to be associated together in both familiar and novel arrangements to make highly personalized systems. Thanks to pervasive networking (e.g. the Internet) such machines and artefacts can communicate and collaborate together so as to support our lives. Already the existence of the of electronic/computer artefacts, such as video recorders, are very evident in our lives. However, a powerful vision being advanced by many researchers, such as those engaged in the EU's Disappearing Computer Initiative (DCI), is that computers will physically disappear into the fabric of our life (become micro or even nano-scale) and cognitively disappear (become capable of relieving us of many everyday tasks without us being aware of their existence). A key to cognitive disappearance is embedding useful amounts of intelligence (i.e. reasoning, planning, learning) into machines thereby relieving the users of such systems of this mental load.

Some of this vision is already becoming a reality as more of the goods we buy are based on tiny embedded computers (e.g. TV, security systems, mobile-phones, washing machines etc) that can be networked together by specialist home networks. Useful mobile robots are starting to appear such as autonomous vacuum cleaners, lawn mowers, service-trolleys, smart-tractors etc. Presently such technology is mostly applied to construct buildings that are safer, more energy efficient, more comfortable, more enjoyable and easier to control. However, this is only the beginning and numerous new possibilities for people are expected to arise from this technology;. The ultimate vision is perhaps of planetary and deep space habitats (c.f. Star Trek) where proponents envision such technology playing a critical role in realizing the aspiration for people to inhabit space.

However, in order to realise this vision, technologies must be developed that will support ad-hoc and highly dynamic (re)structuring of such artefacts whilst, wherever possible, shielding non-technical users from the need to understand or work directly with the technology that will be “hidden” inside such artefacts or systems of artefacts. The authors are engaged in one thread of research that aim to contribute to these ends; the eGadgets project [http://www.extrovert-gadgets.net] funded by the EU DCI programme which, in part, aims to develop compact intelligent embedded-agents (intelligence integrated into computational artefacts) and computational architecture to assist with the above.

At Essex University, the authors have constructed a test-bed for ubiquitous computing work, the iDorm - an intelligent student dormitory. The main focus of the work at Essex concerns investigating methods for embedding useful amounts of intelligence into artefacts thereby enabling a type of ambient intelligence. In the oldest scientific traditions, Essex plans that the initial guinea pigs will be the scientists themselves. The first experimental subject is a Ph.D. student in the Computer Science Department who is researching into embedded-agents and intelligent inhabited environments.

2. Architectures – Systems, Artefacts and Agents

2.1 System Level

A typical ‘container’ environment for ubiquitous computing might be a house or office. In such environments there is wide scope for utilising networked computer-based products to enhance living conditions. For instance computers are sometimes embedded into building artefacts (e.g. lighting, heating etc); entertainment systems (e.g. DVD, TV etc); work tools (e.g. robot vacuum cleaners, washing machines, cookers etc), or safety systems (e.g. security, appliance monitors etc). Some of these artefacts can form part of the building infrastructure and are static in nature (e.g. lighting, HVAC etc), others will be mobile, (e.g. wearables), or nomadic (e.g. TVs or other temporary items). Environments in which computers are used to control building services are generally referred to as “Intelligent Buildings” [Callaghan 00], a paradigm that developments such as the “Disappearing Computer” programme promise to transform radically.

2.1.1 Macro Construction - Enhancing existing paradigms

Some of the most common ubiquitous computing environments are buildings (e.g. homes, schools, hospitals, offices etc). Traditionally, artefacts have a complete and self-contained functionality (e.g. a CD player, security system etc). One way of connecting these together to form macro systems is simply to embed a computer and network interface inside such artefacts, thereby allowing them to be interconnected and remotely accessed. Being a stepwise development of existing technology this is a popular approach to creating current commercial environments. Examples of networking standards employed in these effort include X10, CEBus, Lontalk, BACnet, NEST, EIBus and 1-Wire. Most of these standards support fully distributed or centralised computing models (or hybrids of the two). When designing building-based ubiquitous computing environments one initial consideration is that most buildings are sub-divided by the architect into areas that might usefully be monitored and controlled by a computer-based agent. That is, the architect has already partitioned likely personal needs and behaviours within the spaces as part of the room allocation (i.e. buildings may be regarded as being made up of *rooms* of different functional types such as cooking, sitting, sleeping etc). In this view the range of artefacts in our personal space are, to some degree, based around a *room* (i.e. our behaviour is often associated with the type of room that we are in, and thus so are our artefact system needs). Most large buildings have a great deal of concurrent human activity distributed widely throughout. Thus it is possible to vary the degree of distribution between fully distributed (each artefact being computationally autonomous) to centralised (a number of artefacts being dependent on the same computer). Whilst the former is the ideal of a longer term view of ubiquitous computing, frequently current commercial implementations take a more pragmatic view choosing to make the room the atomic unit. In the next section we will see the opposite view, where existing arefacts are deconstructed to make much smaller atomic units. One possible implementation of a room based distribution strategy is shown in Figure 1. Each room contains sensors and output devices, which are monitored and controlled locally by an embedded-agent. All these agents are connected together via a network, forming a decentralised architecture that enables building-wide collaboration.

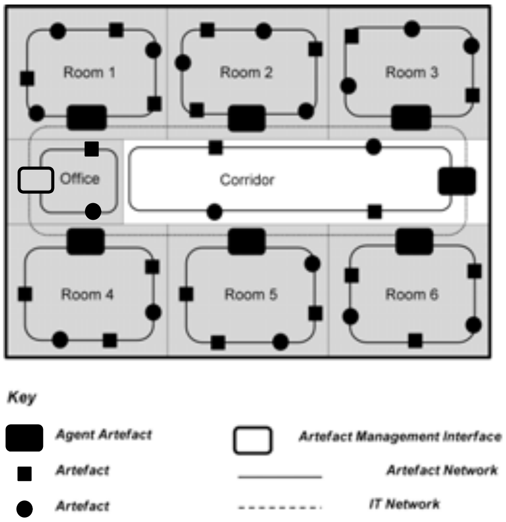


Fig. 1. A Room-Centric Macro Architecture

In this scheme we are therefore dealing with a number of parallel distributed agents, each of which is monitoring a room (or some artefact system decorating a person's personal space), and responding individually to whatever is occurring there. In this way, each agent is focused on responding as well as possible to *the particular needs of the person*, rather than finding an efficient way of satisfying the generalised needs of all the people in the environment (eg building). Of course, there are still some matters that require communication and co-operation between these distributed agents (e.g. responding to an emergency). The building-wide network allows the agents to *selectively share* their information when circumstances require, enabling them to make better decisions regarding situations that have a wider impact on the occupants and building, such as the presence of an intruder, or a fire, for instance. By utilising this approach, in which most of the control is localised to a particular room, inter-agent communication is minimised, resulting in network bandwidth requirements which are only a fraction of the capacity of most existing building networks. Alternatively, it is possible to localise control around other devices such as a cellphone (phone-centric), which has the added advantage of acting as a vehicle for transporting individual preferences and ID across different environments (in the room-centric model these are transported via a network).

2.1.2 Micro Deconstruction - A new paradigm

New paradigms necessarily question the conventional way of thinking about established practices. As the number of tailor-made discrete artefacts increases exponentially the concept of collaborating (atomic) artefacts is a new and powerful way of approaching the problem. In the case ubiquitous systems of the future, the question asked is whether the old technological and economic constraints that have led to current artefacts being largely manufactured as predefined rigid packages of functionality (e.g. Walkman) is still valid. We suggest that thinking in terms of atomic artefacts representing basic generic artefact functionality would afford the user more opportunities for emergent design and use (i.e. the current artefact rigid functionality packaging being an obstruction to user imagination, flexibility and personal design). Thus, we argue, by deconstructing conventional 'gadgets' and then offering flexible, modular, ways of associating their component artefacts into Artefact Systems, a whole variety of new modular devices can be generated. Similarly connectivity between devices of different scale can be facilitated so that modules of hand held devices can be interfaced with different scale modules to take advantage of whatever the environment offers. In such ways a mobile phone might use the speakers on a hi-fi or the image from a TV might be played on a personal organizer screen and so on.

2.1.2.1 Deconstructed Domestic Appliances

The following table lists the sort of domestically situated components that might be flexibly linked together (associated) across a variety of different scale components.

Conventional Macro Level Artefacts found in everyday households.	**Deconstructed Atomic Level Artefact(s)**	**Conventional Macro Level Artefacts found in everyday households.**	**Deconstructed Atomic Level Artefact(s)**
Mobile phone	Speaker	FM/AM Radio	Sound generator (dif to speaker)
Disk man	Amplifier	Cooker	Light source
Walkman	RF Tuner (AM/FM)	Fridge	Window detectors
Personal organiser	CD player	Game System	Door detectors
DVD system	DVD player	Fax	Movement detectors
MP3 system	Storage device	Photo copier	Bell push
Video camera	RF Speech Receiver	Clocks (inc alarm clocks)	microphone
Room lamp	RF Speech Transmitter	Answer phone	Speaker (sound transducer)
Heater	RF Video Receiver	Music Keyboard	Door actuator
Security System	Microphone	Metronome (electronic)	Smoke detector
Electric Door Entry	Display screen	Entry (door) Access System	Heat detector
Safety system (smoke etc)	Video player	Computer	MA detector
IPot (kettle)	MP3 player	Printer	Water level detector
Television	On/off switch	Toaster	Water temp detector
Satellite receiver	Light generator	Electric Knife	Temp detector
Modem	Temperature sensor	Mixer	Door status detector
CD System	Heat source	Shaver	Ring detector
Scales (food)	Bi-directional Storage (data)	Weight analyzer	Weight analyzer
Scales (person)	Bi-directional Storage (video)	Note Generator	Note Generator
Activity (e.g. PIR)	Bi-directional Storage (audio)	Scan Device	Scan Device
	Processor	Print Device	Print Device
	Program	Alphanumeric keyboard	Alphanumeric keyboard
		Voice recogniser	Voice recogniser

These may be usefully classified into 3 types, “Input, “Processor” and “Output”. Examples of Input devices include keypad, touch screen, pen or stylus-activated screen or palimpsest, voice-input microphone. Examples of Output devices include: video screen, speaker, and LED display. Examples of Processor units included amplifier, transmitter, receiver, tuner, media playing devices e.g. CD, DVD, tape, videotape, perhaps retrieving and storing video from hard disks etc. Learning (intelligence), is appropriate to output type artefacts where it would be used to develop rules controlling its output in relation to changes in the wider artefact system.

The point we wish to stress is that in a “*deconstructed appliance*” model, the user now has a richer and less constrained set of possibilities for assembling novel artefact systems (i.e. the possibilities for emergent use is considerably more). Later in this paper, when we discuss iDorm scenarios, we will develop this argument further.

2.2 The Artefact

There is a need to adopt a uniform approach to the structure of the software systems embedded within an artefact. This is necessary, as there are many different forms of artefact. For example, an artefact may or may not contain an agent. To maximise code reuse and to minimise special cases it is necessary to ensure that the software architecture adopted is as flexible as possible and thus offers the maximum benefits to the project.

2.2.1 A Generic Artefact

The following diagram shows the main functions of a ‘generic’ computationally based artefact model, as used in our work. From this it should be possible to derive all other forms of artefact. It is assumed in this diagram that the artefact computing hardware platform interfaces to the local artefact sensors and effectors and that it also provides an inter-artefact communications mechanism. It should be remembered that this diagram represents the functionality of a computational artefact. The structure doesn't imply or exclude any particular implementation.

Different classes of artefact can be formed by excluding particular functionality from the diagram, except that all artefacts contain (in one form or another) an artefact-OS and Hardware Platform. Thus an 'intelligent' artefact would contain the Intelligent Agent component, whereas a 'dumb' artefact would not. The inclusion of the Logic Function component allows non-intelligent artefacts (those that are neither 'dumb' nor 'intelligent') to be created. The Connection-OS handles associations between artefacts (This is equivalent to the GAS system in our EU eGadgets.work).

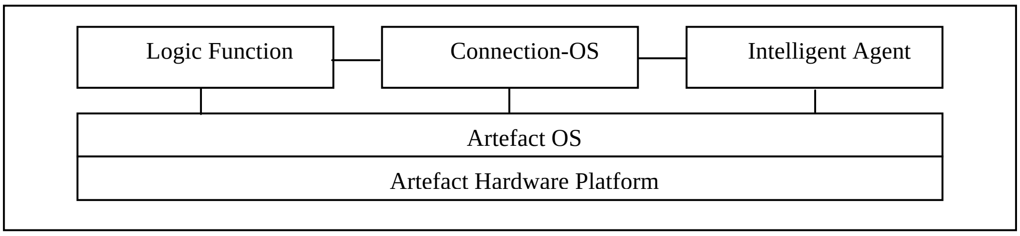


Fig. 2. A Generic Artefact Architecture

2.2.1.1 Component Interconnection Strategies

There are many strategies that could be adopted for the interconnection of the functional components within an artefact. Each strategy offers different set of strengths and weaknesses. The factor of primary concern when deciding upon which interconnection strategy to adopt must be how the chosen strategy impacts upon the usage of the artefact. Of particular concern is the need for the different forms of artefacts to be as functionally equivalent as possible. That is, it would be preferable to only have one version for each module (functional block) of software irrespective of which functional blocks the artefact contained. Thus, in this model, an intelligent artefact (one containing an agent) would have the same Artefact OS as a dumb artefact (one *not* containing an agent). Additionally, accessing the resources of a remote artefact should not be logically different from accessing the resources of the local artefact. There should not be a direct access method for local resources, and an indirect access method for remote resources. A logical access mechanism should hide the implementation of the physical access mechanism. The interconnection approach preferred in our work is shown in the figure below. The functional blocks are not directly connected to each other; the connections form a hierarchy. Only the Connection-OS connects directly to the Artefact OS. The remaining functional blocks only connect to the Connection-OS. This approach only requires two interface definitions:

- Artefact-OS to Connection-OS
- Connection-OS to Intelligent Agent/Logic Function

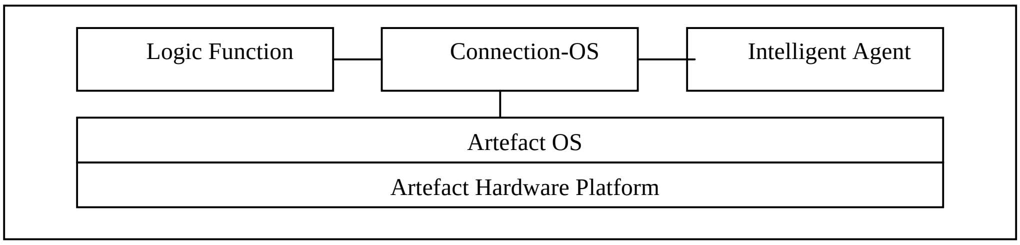


Fig. 3. A Minimised Connection Strategy

The hierarchical nature of this approach requires that the Intelligent Agent/Logic Function access both remote and local resources via the Connection-OS, which in turn accesses them via the Artefact OS. There is no direct access path from the Intelligent Agent/Logic Function to the Artefact OS. For an artefact operating in isolation this may represent an unwanted overhead. In this model, remote resources are accessed via peer-to-peer connections. Should the Intelligent Agent/Logic Function require access to a remote resource it does this via the Connection-OS, which communicates with the remote artefact to access the required resources. This approach follows the standard ISO layer model for network communications.

Of course, rather than operating on their own, systems are generally formed containing two or more artefacts. These artefacts will be interconnected in some user-defined way and share/offer resources to each other. In these circumstances the system described above both minimises the use of special cases and offers a (highly) flexible approach to interconnection with a peer-to-peer communications model and the possibility of using common function coding. However, as one of the functions of Connection-OS is to define artefact interconnections it would seem to be a natural extension of this role. Thus a 'dumb' artefact provides access to its local resources via connections managed by its local Connection-OS. Intelligent artefacts are able to access local resources via the local Connection-OS and, where connections to remote artefacts exist, are also able to access these resources via the same route. The types of connections between the artefacts (permanent, temporary, and backup) are then only known to the Connection-OS, which can manage them accordingly. The intelligent agent is then just has to process data provided by the Connection-OS.

In summary, this artefact architecture provides a symmetrical structure for accessing local and remote artefact resources. It makes 'intelligent' and 'dumb' artefacts functionally equivalent for the common blocks, the additional functionality is provided by adding extra blocks at the top level. It also hides implementation details of the artefact interconnection structure from the higher-level functions, which is also desirable. This approach does however; require that the Connection-OS manage not only the interconnection structure but also access to local and remote resources.

2.3 Agent Architectures for Artefact Intelligence

Ideally, for the vision described in the introduction to be realised, people must be able use computer-based artefacts and systems without being cognitively aware of the existence of the computer within the machine. Clearly in many computer-based products the computer remains very evident as, for example, with a video recorder. Here the user is forced to refer to complicated manuals and to use their own reasoning and learning processes to use the machine successfully. This situation is likely to get much worse as the number, varieties and uses of computer based artefacts increase. Can technology, which is the cause of this problem, be harnessed to provide a solution? We argue that if some part of the reasoning, planning and learning, normally provided by a artefact user, were embedded into the artefact itself, then, by that degree, the cognitive loading on the user would reduce and, in the extreme, disappear (i.e. a substantial part of the computer's presence would disappear.). Put another way, the proportion of reasoning, planning and learning transferred to the artefact (collectively referred to as "embedded-intelligence") is a "cognitive disappearance" metric! Hence we view embedded intelligence as an essential property of artefacts for the cognitive disappearance of the computer and necessary to the successful deployment of new technology in the ubiquitous computing environment. For artefact intelligence we use behaviour based architecture (BBA) derived from the system of mobile robotics as a means of providing artefacts with some useful amount of intelligence. The inspiration for this stemmed a statement from Le Corbusier that "*A house is a machine for living in*" [Le Corbusier, 1921] which from the Essex team's work on robotics led them to extend this view to "*A building is a robot we live in*" [Callaghan et al 99]. The scientific basis of this was that there are many close similarities in the problem specification for mobile robots and ubiquitous computing environments such as compact computation, being situated, embodied, real-time and the need to deal with what are essential non-deterministic problems.

In order for an artefact agent to respond appropriately, it needs knowledge about the environment (i.e. the artefact itself and the current context in any artefact system in belongs to) including people interacting with the artefacts. In other publications we have shown it is essentially impossible to create a useful model of these in advance [Callaghan 01A]; therefore the artefact-agent must acquire its knowledge in another way - through its perceptive capabilities, i.e. via sensors.

By gathering information from its sensors over a period of time, the artefact-agent can 'notice' how a particular person tends to behave in particular circumstances, and can then *learn* to "mimic" or replicate that behaviour itself. If there are sensors to distinguish between different users, the system is able to learn different behaviours for different people. So for example, an agent based light artefact might learn that Person A, who is only partially sighted, prefers a higher level of light than Person B, whose sight is normal. It could then adjust the lighting level appropriately, according to who was using the artefact at that time.

2.4 Some Research Challenges for Achieving Artefact Intelligence

Above we argued that transferring some cognitive load from the users into the artefact was a key element in achieving cognitive disappearance. This is not straightforward and we now describe some of the research challenges involved. A more detailed analysis of the issues is given in other work [Callaghan 01B]

2.4.1 The Issue of Physical Size and Cost

For physical disappearance artefacts will need relatively small low-cost embedded computers (possibly based on application specific micro-electronic fabrication). For example current (2002) specifications might be: Cost: £20-£50, Size: <2^2cm, Speed: 1-10MHz, Memory: 1-2 MB, I/O: 10-50 I/O channels. Traditional artificial intelligence (AI) techniques are well known for being computationally demanding and therefore unsuitable for 'lean' computer architectures. In addition traditional AI techniques have proved too fragile to operate real time intelligent machines such as robots. As a result, even implementing simplified traditional AI systems on embedded-computers has proved virtually impossible. This has led researchers to look at alternative paradigms such as behaviour-based methods from robotics. With recent technological advances in nano-technology, more difficult issues are raised in terms of lowering the cost of potentially very expensive nano-electronics that, by their nature, are likely to be used in greater numbers.

2.4.2 The Issue of Distribution

In most disappearing computer style scenarios, computer based artefacts are able to form ad-hoc groupings which work together to achieve some higher-level purpose. From an AI viewpoint this raises questions such as:

1. How is AI (agent) functionality and computation distributed (e.g. what is the computational granularity of artefacts, are they computationally and functionally autonomous)?
2. How are associations to other artefacts formed and recorded (i.e. does each artefact decide and record its own associations or is this centrally managed and recorded)? Such associations are critical to group co-ordination, synergy and learning.
3. How are the dynamics of artefact mobility and failure handled (how do artefacts choose between competing services or cope with the removal of a service)?
4. How is group control and contention arbitrated (is there a master artefact in overall charge or is this devolved)?
5. How do artefacts/embedded-agents communicate with each other (what is an appropriate and compact language to support the expression needed for generalised intelligent-artefact communication and co-operation)?

2.4.3 The Issue of Mobility

Artefacts can be mobile to differing degrees. For example a mobile phone follows the users movements through a variety of environments. If it were to collaborate with sets of local agents then its presence in their group may be fairly short. At the other extreme there maybe fixed computer based artefacts in buildings (e.g. HVAC systems), which are effectively permanent and static in nature. There are also intermediate levels of mobility such as that of a CD player brought into a building by an owner, which may be there for a number of weeks, or years, before being moved. Clearly the technical infrastructure has to deal with these varying dynamics of mobility and association. The following table summarises these possibilities.

	Centralised	**Distributed**
Static	Orchestration of groups of fixed artefacts by a single centralised computer	Anarchical (co-operating, self-organising, non-hierarchical) collaboration of groups of fixed autonomous artefacts
Semi-Static	Orchestration of groups of temporally located artefacts by a single centralised computer	Anarchical collaboration of groups of temporally located autonomous artefacts
Mobile	Orchestration of groups of continuously moving artefacts by a single centralised computer	Anarchical collaboration of groups of continuously moving autonomous artefacts

2.4.4 The Issue of Dimensionality and Temporality

The quality of agent decisions is limited by its knowledge of the world. The agent gets its knowledge from sensors directly attached to it and from other agents (i.e. indirectly from their sensors). Which set of sensor information is sufficient for an agent to make a particular class of decision? Consider a simple heating controller, why does the room's occupant alter the heat value? Is it related to the current temperature, his current level of activity, what he is wearing, where he is in a room, where he has just been? We may decide that it is based upon current temperature and therefore could operate with only one sensor, but later discover that an agent that used only one sensor was not working very effectively. At the other extreme we could decide we should sense 'everything' and then let the agent learn which of these sensed values was important. Clearly in this latter situation the agent would be able to make better-informed decisions and adapt to changing criteria. In addition this problem exposes a central dilemma, what is the best mechanism for selecting relevant sensory sets for agents? Is it the designer or the agents themselves? The problem with a designer is the assumption that people know best what the intelligent agent needs; but is this true? We would argue that it is better to provide a large set of sensory inputs to agents and provide learning mechanisms to let them resolve which of the stimuli is important for any given decision wherever possible. Whilst this latter argument may have some appeal it carries with it a penalty, the need to compute using large sensory input vectors. Thus, large sensory sets are an issue for intelligent-artefacts. One solution is the development of mechanisms that allow embedded-agents to "focus" on sub-sets of data relating to specific decisions or circumstances. An additional problem is that of time and sequences. Often the reason an action is taken is not simply related to the current state of the world, but to the sequence of states that led up to the most recent event. Thus, an effective embedded-agent would need to be able to deal with temporality.

2.4.5 The Issue of Non-Determinacy, Intractability and Dynamism

Traditional AI is based around the so-called Sense-Model-Plan-Act (SMPA) architecture. In this there is an assumption that the world the agent acts upon can be abstractly described by either a mathematical model or some form of well-structured representation. In addition, it is usually assumed that the state of the world can be sensed reasonably reliably and compared to the abstract representation so as to reason or plan about the world. This approach works reasonably well for some forms of problem e.g. chess playing programs where many of these axioms hold true, but completely fails in robotics and other applications that involve an intimate relationship with the physical world. The reason that traditional AI fails in such physical applications has been well described by others [Brooks 91] but a simplified explanation would be that the assumption that the world can be accurately sensed and modelled (the key axiom of SMPA) does not hold. Fortunately, robotics has generated a potential solution for this type of problem that works by discarding the abstract model and replacing it by the world itself; a principle most aptly summarised by Rodney Brooks as, "the world is its own best model". This AI school is known as "new AI" or perhaps more meaningfully "behaviour based AI".

2.5 Types of Artefact Intelligence

At a high level there is a gross distinction between intelligence that concerns an artefact's effectors (taking input data from physical sensors, locally or remotely situated) and intelligence that concerns communication connections with potentially many different sources (taking an input date from connection topology databases or network performance monitors). The former type of intelligence can be seen to be a control-oriented agent whilst the latter is an information-oriented agent, each requiring very different agent techniques.

Control-oriented agents, can exploit sensor data at both an Artefact and Artefact-System level. At an individual artefact level they use an internal self-learning control loop between their sensors and effectors producing an intelligent artefact. At a multi-artefact level (Artefact-System) the agent in each artefact uses connections to remote artefacts thereby becoming a better artefact (decisions are better informed). An implicit consequence of each artefact using other artefacts to determine its action (at least in part) is the orchestration of collective Artefact-System behaviour (as local artefact actions, which contribute to the overall Artefact-System behaviour, are informed by the state of other artefacts).

Information oriented agents use information on the state or performance of connections (from network or plug databases) to repair or improve the inter-artefact connectivity with the aim of making a better performing Artefact-System.

The roles of intelligence, as outlined above, can be thus summarised as:

- Artefact Intelligence – to make an individual artefact a better controller (local adaptation) from fixed set of inputs within the artefact itself.
- Artefact-System Intelligence – to make artefacts work with whatever set of Artefact-System associations a user decides to give an artefact (which may be less or more than optimum). The agent would determine how to operate with what it is given or to adapt to changing connections.
- Connection Intelligence – a mechanism, distinct from the artefact actuator control, working with connection information (i.e. associations between artefacts) to proactively and autonomously set or finds better/replacement associations.

In addition to describing the types of agent that are possible (as described above), there is the important matter of how agents learn. Agents that do not explicitly involve the users in a forced interaction are termed *non-intrusive* whilst those involving explicit user interaction are termed *intrusive*. Intrusive learning can be useful for accelerating system set-up, whilst non-intrusive learning is more desirable in life-long learning situations. This can be summarised as:

- Non-intrusive learning – an implicit learning mode where the normal use of the artefact or Artefact-System is used to teach the artefact agent user needs (this is a form of life-long learning).
- Intrusive learning – an explicit learning mode where the user is involved in an accelerated interactive cycle teaching the artefact agent user needs (a form of initialisation or set-up phase)

Thus artefact-based intelligence has both levels and modes. A fuller view of possibilities for artefact intelligence is presented in the following table:

	Sphere of Operation		
Degree of Intelligence	**1. Artefact**	**2. Artefact-System**	**3.Connection-OS**
A. Dumb	A situation where artefacts are not interconnected (i.e. are stand –alone). Control/functionality of the Artefact is programmed into Computational Logic.	A situation where artefacts are interconnected to form Artefact-Systems using an association editor or by physical means. Control/functionality of the artefact is programmed into Computational Logic.	A situation where artefacts are interconnected to form Artefact-Systems using an association editor or by physical means. There may be some auto-repair/configuration based on simple procedural operations. Control/functionality of the artefact is programmed into Computational Logic.
B. Intelligent	A situation where artefacts are not interconnected (i.e. are stand –alone). Control/functionality of the Artefact is provided by intelligence (an agent) in the artefact.	A situation where artefacts are interconnected to form Artefact-Systems using an association editor or by physical means. Control/functionality of the artefact is provided by intelligence (an agent) in the artefact.	A situation where artefacts are interconnected to form Artefact-Systems by an intelligent autonomous process (e.g. distributed to each artefact). Control/functionality of the artefact is provided by intelligence (an agent) in the artefact.

2.6 An Embedded-Agent Architecture for Ubiquitous Computing Artefacts

As a first step we have developed an agent (see figure 5) that makes significant progress towards meeting the challenges described above. It uses a BBA approach from robotics that is computationally compact, operates in real-time and is able to cope with the non-determinism. It also learns to particularize itself. In more details the embedded-agent developed by Essex is based on the use of fuzzy logic implementation of a Behaviour Based Architecture (BBA) and a novel Genetic Algorithm (GA) / CASE based learning method referred to as Incremental Synchronous Learning (ISL), itself derived from an earlier generic CASE inspired method known as evidential learning. Figure 4 depicts the internal structure of the artefact agent. It is clear that, in order for an agent to autonomously particularise its service to an individual, some form of learning is essential. In our agent learning takes the form of adapting the dynamic "particularised use" behaviour's rule base, according to the user's actions. To do this we utilise an evolutionary computing mechanism based on a novel hierarchical genetic algorithm (GA) technique, which modifies the fuzzy controller rule-sets through interaction with the environment and user. Each behaviour comprises a fuzzy logic controller (FLC) that has two parameters that can be modified, a *Rule Base* (RB) and its associated *Membership Functions* (MF). In our learning we modify the rule-base. The behaviours receive their inputs from sensors and provide outputs to the actuators via the c*o-ordinator* that weights their effect. When the system fails to have the desired response (e.g. an user manually changes an effector setting), the learning cycle begins. The system then remains with this set of active rules (an experience) until the user's behaviour indicates a change of preference (e.g. has developed a new habit), signalled by a manual change to one of the effectors when the learning process described above is repeated. In the case of a new user the *Contextual Prompter* gets and activates the most suitable rule base from the *Experience Bank* or if this proves unsuitable the system re-starts the learning cycle above. The *Solution Evaluator* assigns each stored rule base in the *Experience Bank* a fitness value. When the *Experience Bank* is full, we have to delete some experiences. To assist with this the *Rule Assassin* determines which rules are removed according to their importance (as set by the *Solution Evaluator*). The *Last Experience Temporal Buffer* feeds back to the inputs a compressed form of the n-1 state, thereby providing a mechanism to deal with temporal sequences. A key point to note is that, over time, the ISL method develops (and continues to adapt) a set of behaviours that are *tailored to that particular artefact system and its users*, by relying on information gathered from sensors instead of from a pre-programmed model. It is evident that the *basic* behaviour of the system must at least be equivalent to the behaviour of the artefact if the agent were not present. We refer to this fundamental behaviour as the *manual behaviour*. However, this fixed behaviour alone is insufficient as a minimum fallback, as there are some situations (such as emergencies) that the agent must always be able to deal with correctly - it must not have to wait to learn these over time. For this reason, the system includes some permanent, fixed basic rules, which ensure the agent always behaves safely and efficiently, and is able to handle such situations.

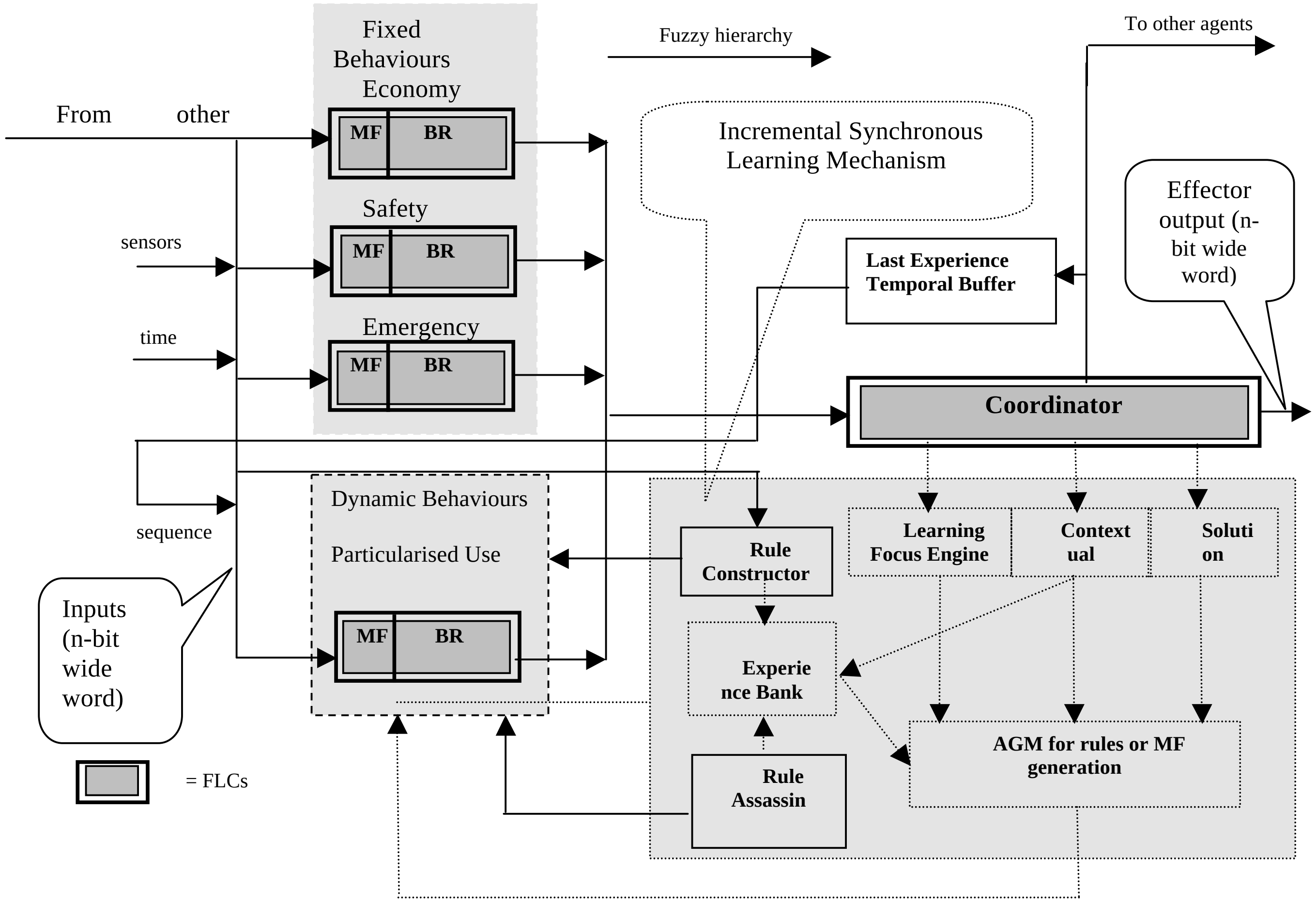


Fig. 4. Artefact Based Agent (UK patent No 99-10539.7)

In our work we employ three modes of agent interaction with the user. The first is a pre-emption mode where the agent continually tries to anticipate and set the artefact's state to meet the user's needs. The second, "assistance-on-demand," is where the agent shadows the users only offering assistance "on-demand". The third mode - "inverted" - is where the agent learns from the user's actions but doesn't attempt to pre-empt his needs; instead the agent flags departures from the normal. This mode is especially useful where one needs to design an agent that looks for a abnormal behaviour (as can be the case with subtly deteriorating medical conditions). The "inverted agent" mode is potentially popular as, from the author's experience, the public more readily accepts agents that aim to increase personal safety or help with medical problems. This capability arises as a consequence of the agents particularisation methodology that creates rules that fits an individual behaviour rather than developing rules that generalise across numerous users, or behaviours. The underlying principles and implementation are described in depth in other publications and readers with an interest in learning more about these methods are referred to that work [Callaghan 01A].

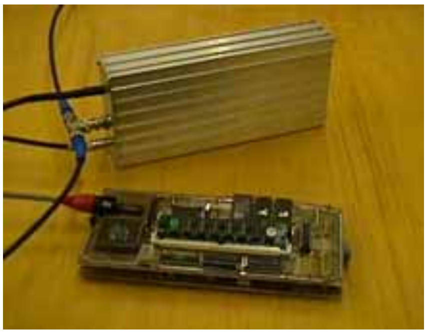


Fig. 5. The Essex Prototype Artefact Agent

Multi-artefact (multi-agent) operation is supported by making compressed information about the current state of the agent available to the wider network. The compressed data takes the form of which behaviours are active (and to what degree). The general philosophy we have adopted is that data from remote agents is simply treated in the same way as all other sensor data. 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, which regards remote agents as simply more sensors (albeit, sophisticated sensors) and differs from message based coordination typified by Agent Communication language (ACL) models. This is a large and complex subject beyond the scope of this paper but we refer interested readers to our work concerned with intelligent-building and agent communication languages [Cayci 2000].

3.0 The iDorm – A Testbed for Ubiquitous Computing and Ambient Intelligence

We have constructed an intelligent dormitory (iDorm) at the University of Essex to experiment with the systems described above. Being a student dormitory it is a multi-use space (i.e. contains areas with differing activities such as sleeping, working, entertaining etc). The occupant of the room (a student) is free to decorate his room with whatever artefacts he chooses (computer and non-computer based, passive and active). Because this room is of an experimental nature we have fitted it with a liberal placement of sensors (e.g. temperature sensors, presence detectors, appliance monitors etc) and effectors (e.g. door actuators, appliance switches etc), which the occupant can also configure and use. Our expectations are that the occupant would chose to decorate his personal space (the room) with a variety of artefacts ranging from building service devices such as heaters to entertainment systems such as CD/TV.

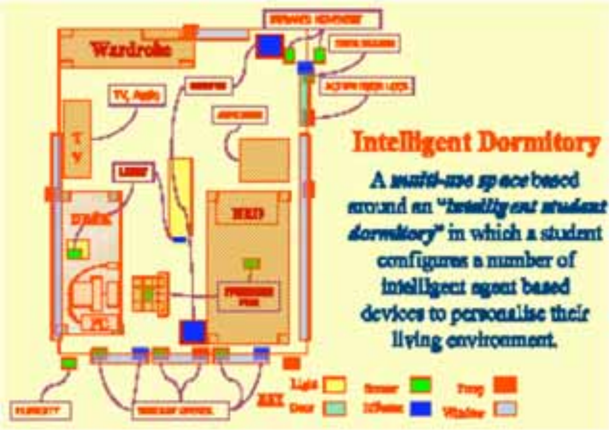


Fig. 6. Views of the iDorm

3.1 iDorm Based Scenarios for Embedded Agent Assisted Ubiquitous Computing

An illustrative scenario is as follows. The student moves into the dormitory, which contains some existing artefacts (mostly connected with the room infrastructure) but brings other more personal artefacts with him. He then runs a configuration program on his PC that allows him to set up associations between sensors and effectors.To take a mundane example concerning the room's infrastructure, the student might set an association between a light switch immediately inside the door and a number of room lights. In addition he could personalise this space by deciding to associate the same light switch sensor to his radio, so that the radio switches on whenever he enters the room. He then continues until he has associated together all the sensors, effectors and artefacts that interest him. Having set up a basic artefact association the occupant may then choose to switch the artefacts into an active online learning mode (or leave them as manually set). In general the room artefacts function as non-agent based systems, interacting with the user through conventional controls (no special embedded-agent controls are necessary and the user is essentially unaware agents exist, or that this is anything other than a normal environment). In the active mode artefacts monitor their use, in relation to the state of their local world, programming themselves to satisfy the occupant by doing what he habitually and persistently wants (i.e. not simply learning random whims of a user but rather learning long term persistent requirements, what we call 'learning inertia' in the embedded-agent research we have undertaken). At the same time as learning habitual and persistent user requirements, the embedded-agents also respond immediately to any command made by the occupant. Thus after some time has passed the intelligent-dormitory will learn how to configure and operate the constituent intelligent-artefacts to the benefit of the occupant. Of noteworthy mention is the linkage to the "inverted agent" mechanism discussed under the agent section. As the learning method used particularises to individuals, it learns the users usual behaviour and inherent to the method is the detection of abnormal (potentially new) behaviour. This can be used to good effect in scenarios aimed at detecting unsafe or unusual situations or subtle deterioration resulting from a progressive medical condition. This description is not comprehensive in coverage, and clearly speculative in places, but we hope it helps expose some of the issues and gives a feel for the type of operational issues and possibilities involved. In more general terms, consider a room populated with the following computer based networked artefacts.

Switches	Heater	Amplifier	Tone generator	Computer keyboard
Lights,	Temp Sensors	Speaker	Door access	Processor
Light-level sensor,	TV Tuner	Screen (video)	Activity monitor	
Clock	Radio Tuner	Music keyboard	Storage	

By associating various items together it is possible to make all sorts of artefact system. Some simple examples might be:

Artefact System	Associations	Scenario Description
Room Light	Switch, light, light level detector, clock	Mimics conventional light control. If Agent is in light actuator can learn preemptive control
Room Heater	Switch, heater, temp-sensor, clock	Mimics conventional heater control. If Agent is in heater actuator can learn preemptive control
TV	TV Tuner, amplifier, speaker, screen, clock	Mimics conventional TV control. If Agent is in Speaker & Screen effectors can learn preemptive control (e.g. volume)
Video Recorder	RF TV Tuner, screen, clock, storage	Mimics conventional video system. If Agent is in storage effector can learn preemptive control (e.g. regular recording)
DVD	DVD player amplifier, speaker, screen, clock.	Mimics conventional DVD control. If Agent is in Speaker & Screen effectors can learn preemptive control (e.g. volume)
Digital Piano	Music keyboard, tone generator, amplifier, speaker, clock	Mimics conventional Electronic Piano control. If Agent is in Speaker & Screen effectors can learn preemptive control (e.g. volume)
Metronome	Clock, tone generator, amplifier, speaker	Mimics conventional Electronic Metronome. If Agent is in Speaker & Screen effectors can learn preemptive

		control (e.g. initial beat, volume)
Security	Door access, activity, tone generator, amplifier, speaker	Mimics conventional security system using door access to trigger tone generator that is amplified and emitted via speaker. If Agent is in Speaker & Tone Generator effectors can learn preemptive control (e.g. what tone to associate with what person/event)

From the simplest configurations it is possible to see that emergent use (open ended user design) added by agent learning is possible. To illustrate this consider the following simple set of artefacts which the user might associate together:

- RF TV Rx
- Door access
- Amplifier
- Speaker
- Light in hall

If the artefacts were agent based then the amplifier might learn from user action that whenever the doorbell was activated the user turns down the TV sound (i.e. the amplifier) and turns on the hall light. To extend this scenario slightly there might be a camera at the door, which could be used to display a picture of who was at the door inset on the TV. On a similar tack, when you receive a text message on your mobile phone the message might be displayed inset on the TV. Another example would be a mobile phone that accepts forwarded calls from the home phone when the user is outside of the latter's audible range. This would require association between an artefact able to tell the location of a user, a phone artefact and a mobile phone. Another use of a similar associate would be to add a doorbell into the Artefact System. This would enable the mobile phone to buzz when someone was using the doorbell and the user is outside of the audible range of the buzzer. Clearly an almost endless and much more complex set of artefact systems could be constructed with very varied user use.

More complex examples could include those in the following table:

Artefact Systems	Description
Comms World	Communication and Entertainment Artefact Systems (e.g. TV, Video, Phone) learn to work together to interrupt and deliver information to the appropriate context. For instance it might learn to SMSs, Security Camera Views on screen or mute sound when incoming phone call if occupant is using TV Artefact System at time etc
Study World	A system in which constituent output artefacts (e.g. lights, timers) learn to activate themselves according to the users needs for desk-based working. For example the desk Artefact System would learn to change lighting conditions depending on whether the focus of the work was paper or computer based
Care World	Allows occupant to teach room what is normal behaviour and signal occupant (and external carer) when abnormal activities above some level encountered
Music World	Allows the occupant to teach the room how to configure itself (lighting and sound) for different types of music; and different positions in the room (e.g. bed, desk, armchair). It would also learn to configure itself for playing of music.
Eco World	The Eco system would use the agents in the energy guzzlers to aim to learn how to maximize environmental comfort whilst, where there is no conflict, reducing energy consumption). This would also serve as an example of fixed behavior use in artefact-based agents.
Games World	The Games system would seek to couple itself to general environment Artefact Systems (e.g. lights, sound) to provide a more immersive and context sensitive environment.
Etc Etc Etc	Limited only by user’s imagination

3.2 iDorm Technology

The Intelligent Dormitory (iDorm) is a test-bed for ubiquitous computing environments based on a university student dormitory. The iDorm provides a variety of technology aimed at providing a rich and flexible testbed [Holmes 02, Pounds-Cornish 02]; the following providing an overview of the networking and interfacing technology. The iDorm uses three main communication protocols to allow its devices to communicate with each other. Such a variety of networks and protocols were chosen because any successful intelligent agent produced for the iDorm can be shown to be network independent.

LonTalk is an off-the-shelf communications network designed for intelligent buildings. It is a twisted pair network, similar to IP that comes in two flavours – one that provides power to the devices through the network and another that requires devices to have an external power supply. The majority of the sensors and effectors inside the iDorm are connected via a LonTalk network.

The 1-Wire protocol has been designed and implemented by Dallas Semiconductors. It is designed for small-scale applications where the distances between devices on the network are relatively small. It is addressable on the IP as well as on the 1-Wire network. A Java Virtual Machine is embedded on the network board and the research group has written a small server that interfaces sensor/effector information via HTML .

Fig. 7. One Wire and Lontalk architecture in iDorm

The iDorm uses a single network (IPv4) to link the different networks together. This allows a common protocol to be produced that all interfaces could use to communicate with the iDorm. There are several distinct advantages to this approach:

- The first is that a common interface immediately creates a scalable environment. More sensors can be added to existing networks or entirely new network protocols can be added to the iDorm without having to re-configure every other network that communicates in the room.
- The second is robustness. More than one network can provide similar information, if one fails the other can seamlessly provide that information. For example, the iDorm has temperature information available on both the Lonworks network (Figure 6) and the 1-Wire network.
- The third advantage is that a common interface doesn’t limit an interface to a certain way of expressing data. If all the iDorm’s environmental information is available as simple states and values then it is entirely up to the interface designer as to how and in what format that data is used.
- The fourth advantage is that of security. If the iDorm’s information is available through a single communication protocol, it is far easier to decide whether the client is entitled to receive this information. This entitlement can be decided on anything from identification or time. The group uses the latter concept to timeshare access to the iDorm when more than one experiment needs to run at one time.
- The fifth advantage is that a common protocol allows a dynamic interface to be created. An example of this is the voice recognition interface explained later in this paper.
- The sixth advantage is that the processing power required to gather information from the room is greatly reduced by placing the onus on the common protocol to provide the information. This system reduces the amount of processing required from the interface.

The protocol that has been produced is an XML definition for the iDorm. All information requested from the iDorm must go through a central server. This server communicates with the iDorm’s LonTalk and 1-Wire network across IP using HTTP requests to get environmental information and request changes to the states of the effectors.

Fig. 8. A high-level View of the iDorm network

3.3 Interfaces

The group designed several interfaces to deal with the problem of being able to control the room with as few constraints as possible:

3.3.1 The Standard Interface

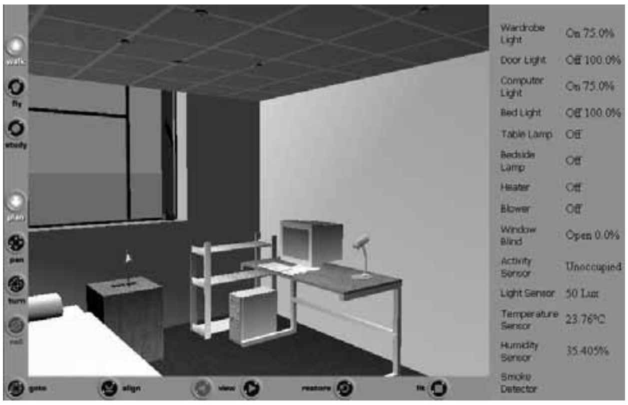
There are normal switches mounted on the walls in the iDorm that control all the effectors (lights, blind, heaters). However, these switches are not directly connected to the device they control. Each switch and button is a device on the LonTalk network. As such, it transmits a data packet across the network when it has been pressed.

3.3.2 The Web Interface

A small web page has been created which is accessible from any machine running a web browser. It shows the current status of the iDorm that automatically refreshes. The user can select the changes they wish to make to the environment; click on the "Update" button and the room will change. Because the web page is very simple and very small, it is possible to view it on smaller web enabled devices such as a palmtop.

3.3.3 The VRML Interface

This is a hybrid system that marries the Virtual Reality Modelling Language with a Java interface controlling the iDorm. The VRML interface takes the form of a scale three-dimensional model of the iDorm and its contents (Figure 9). It allows the user to move through the model on any computer with a suitable VRML viewer. It is possible to interact with the 3D representations of the devices inside the iDorm.



Wardrobe Light	On 75.0%
Door Light	Off 100.0%
Computer Light	On 75.0%
Bed Light	Off 100.0%
Table Lamp	Off
Bedside Lamp	Off
Heater	Off
Blower	Off
Window Blind	Open 0.0%
Activity Sensor	Unoccupied
Light Sensor	50 Lux
Temperature Sensor	23.76°C
Humidity Sensor	35.405%
Smoke Detector	

---iDORM---

The temperature is currently 23.19 degrees C.

The Desk Light is 1

Fig. 9. VRML and WAP screenshots

3.3.4 WAP Interface

This interface is a simple extension of the web interface. Because the iDorm central server can also support the WML language it is possible to interact with the iDorm on mobile phones.

3.5.5 Voice Recognition Interface

Nikola Kasabov and Waleed Abdulla from the University of Otago in New Zealand originated a speaker independent voice recognition system. The Otago research group is applying it using a room-based command set appropriate to the iDorm. Based on Hidden Markov Models, the system contains commands created by the user. Several of the command's behaviours are dynamic, depending on the current state of the room. For instance, the command "brighter" takes an average of the ceiling light levels, adds 10% to the value and sets the spotlights accordingly. This command means the ambient light level of the room can be controlled without having to give individual commands to each spotlight.

4.0 Summary

In this paper we have described an architecture (hardware, software, communication and agents) that we are employing on the EU DCI eGadgets project. The main goal of this project is to support user-driven design of ad-hoc assemblies of computer-based artefacts that will make up many envisaged ubiquitous computing environments. Embedding useful amounts of intelligence into artefacts was seen as an essential enabling technology to achieve the vision of the eGadgets project. This paper has presented the main architectural techniques being successfully deployed in the initial stages of the work. Especially noteworthy aspects of the work presented are:

- Behaviour based architecture techniques taken from the field of mobile robots are well suited to forming agents that can be embedded into artefacts.
- Particularised learning of individual's artefact usage is an especially important characteristic as it both enables the artefact and environment to be fully personalised, whilst offering some valuable side-benefits such as the inverted safety agent.
- Coordination in multi-artefact systems can be accomplished at a simple, but adequate level by treating other artefacts (and agents) as sensors.
- User selectable interaction modes give the designer and user more scope to adapt the operation of ubiquitous computing systems to individual needs. In particular, the metaphor "the user is king" is useful (vital in the mind of the authors) when designing artefact control and interaction systems.
- A modularised hierarchical software architecture of the type described in this paper is well suited to artefacts as it provides a consistent model for both local and remote communication, facilitates dumb and smart artefacts without special provision and is scalable across the range of artefacts and systems.
- Many of the techniques are scaleable and may transfer onto nano scale operations
- A student dormitory makes an excellent evaluation platform for ubiquitous computing and ambient intelligence work as it provides a compact multiuse space with occupants that are sympathetic to exploring new technology.
- In practical terms, HTML, WAP, VRML and XML provide a highly flexible and generic means of interfacing to ubiquitous computing environments.

We have also discussed how transferring some cognitive capabilities from people into artefacts provides a natural mechanism to facilitate the disappearance of computers as computers are increasingly embedded into our daily environment. We have also argued that embedded-intelligence can bring significant cost and effort savings over the evolving lifetime of product by avoiding expensive programming (and re-programming). In particular, if people are to use collections of computer based artefacts to build systems to suit their own personal tastes (which may be unique in some sense) then self programming embedded-agents offer one way of allowing this without incurring an undue skill or time overhead. However, whilst this paper strongly argues that integrating embedded intelligent agents into artefacts is highly beneficial, it also exposes several significant problems, many of which remain as research challenges. For instance, dealing with the problems of non-determinism, dimensionality and temporality in computationally compact environments are very challenging topics.

We also presented an overview of an intelligent inhabited environment in the form of the iDorm that we are using as a test-bed for intelligent artefacts in the EU DCI eGadgets project and for the CareAgent project (part of a Korean-UK Scientific Fund Programme) that includes co-operation between fixed agents and mobile robots. Our work is ongoing, in particular we are planning extensive trials over the summer (including significant habitation of the iDorm) and we intend to report the latest results and to show videos of this at the conference. We also look forward to reporting results of more experiments in the iDorm in a future paper.

Acknowledgements: It is a pleasure to acknowledge our eGadget partners Achilleas Kameas, Irene Mavrommati, Manolis Koutlis, Dimitris Riggas (CTI) and Kieran Delaney, Frank Buckley; John Barton (NMRC), whose thought provoking discussions have helped shape our thoughts on embedded-agent and computing architecture design for ubiquitous computing environments. In particular, we acknowledge Kieran Delaney's formative contribution to the section on deconstructed electronic appliance artefacts. We gratefully acknowledge the financial support from the EU FET "Disappearing Computer" and Korean-UK Scientific research programmes, which have enabled this work.

References

[Brooks 91] Brooks R, "*Intelligence Without Representation*", Artificial Intelligence 47, pp139-159, 1991

[Brooks 97] Brooks R. "*The Intelligent Room Project*" Proceedings of the Second International Cognitive Technology Conference (CT'97*)*, Aizu, Japan, August 1997

[Callaghan 00] Callaghan V, Clarke, G, Pounds-Cornish A "*Buildings As Intelligent Autonomous Systems: A Model for Integrating Personal and Building Agents*", The 6th International Conference on Intelligent Autonomous Systems (IAS-6), Venice, Italy; July 25 - 27, 2000

[Callaghan 01A] Callaghan V, Clarke, G., Colley, M., Hagras, H. "*A Soft-Computing DAI Architecture for Intelligent Buildings*", Journal of Studies in Fuzziness and Soft Computing on Soft Computing Agents, Physica-Verlag-Springer, June, 2001

[Callaghan 01B] Callaghan V, Clarke G, Colley M, Hagras H "*Embedding Intelligence: Research Issues for Ubiquitous Computing*", The 1st Equator IRC Workshop on Ubiquitous Computing, 13-14 Sept 2001, Nottingham UK

[Cayci 00] Cayci F, Callaghan V, Clarke G, "*DIBAL - A Distributed Intelligent Building Agent Language*", The 6th International Conference on Information Systems Analysis and Synthesis (ISAS 2000), Orlando, Florida, July 2000

[Clarke 00] Clarke G, Callaghan V, Pounds-Cornish A "*Intelligent Habitats and The Future: The Interaction of People, Agents and Environmental Artefacts*", 4S/EASST Conference on Technoscience, Citizenship and Culture in the 21st Century , Vienna, 26-28th September 2000

[Colley 01] Colley, M., Clarke, G., Hagras, H, Callaghan V, "*Intelligent Inhabited Environments: Co-operative Robotics & Buildings*" 32nd International Symposium on Robotics (ISR 2001), Seoul, Korea April 19-21, 2001.

[Davisson 98] P. Davisson "*Energy Saving and Value Added Services; Controlling Intelligent-Buildings Using a Multi-Agent System Approach*" in DA/DSM Europe DistribuTECH, PennWell, 1998.

[Finin 94] T. Finin, R. Fritzson, D. Mckay, R. McEntire. " KQML: An Information and Knowledge Exchange Protocol, "Knowledge Building and Knowledge Sharing, K. Fuchi and T. Yokoi (Eds), Ohmsha and IOS press, 1994.

[Hagras 00] Hagras H, Callaghan V, Colley M, Clarke G "*A Hierarchical Fuzzy Genetic Agent Architecture for Intelligent Buildings Sensing and Control*", RASC 2000 - International Conference on Recent Advances in Soft Computing June 29 & 30 2000, Leicester, UK

[Hagras 01] Hagras H, Callaghan V, Colley M, Clarke G, "A *Hierarchical Fuzzy Genetic Multi-Agent Architecture for Intelligent Buildings Learning, Adaptation and Control*", International Journal of Information Sciences, August 2001

[Holmes 02] Holmes, A., Duman, H. and Pounds-Cornish, A., "*The* iDORM*: Gateway to Heterogeneous Networking Environments*", International ITEA Workshop on Virtual Home Environments, 20-21st February 2002. Paderborn, Germany

[Minar 99] Minar N., Gray M., Roup O., Krikorian R., Maes P. "*Hive: Distributed Agents for Networking things*",. ASA/MA August 3, 1999. Cambridge, USA

[Mozer 98] MC 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.

[Pounds-Cornish 02] Pounds-Cornish A, Holmes A, "*The iDorm - a Practical Deployment of Grid Technology*" 2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002), May 21-24 2002, Berlin, Germany.

[Sharples 98] Sharples S, Callaghan V, Clarke G, "*The Application of Intelligent Building Techniques to Care Service Provision*", IEE Colloquium on Intelligent Methods in Healthcare and Medicine, York. Oct 1998.

[**Sharples 99]** Sharples S, Callaghan V, Clarke, G "*A Multi-Agent Architecture For Intelligent Building Sensing and Control*", Int'l Sensor Review Journal, Vol. 19. No. 2. May 1999

[UK Patent 99] Genetic-Fuzzy Controller, UK No 99 10539.7, 7th May 1999