In Proceedings of the IEE Workshop on Perspectives in Pervasive Computing, London, UK,25 October 2005.

USER INTERACTION IN A SHARED INFORMATION SPACE – A PERVASIVE ENVIRONMENT FOR THE HOME

P R Limb, S. Armitage, J S Y Chin, R Kalawsky, V Callaghan, P M Bull, H Hagras, M Colley

British Telecommunications, University of Essex, Loughborough University

Abstract

Connected technology in the home is growing rapidly due to advances in modern electronics. The growth of broadband is being followed by an associated uptake in home networks. At present the enabling technology tends to be visible and distinct (PCs, multimedia systems, Televisions, screens, etc). In the future, we predict the home will incorporate embedded systems and even greater networked-enabled shared resources resulting in a pervasive computing environment. Currently, there are a number of challenges to make this a reality. For example, how will users interact with the technology in a way that is intuitive and non-threatening; how will the wide range of devices communicate and coordinate their requirements and services? This paper describes research which is investigating that as devices in the home become more interconnected and announce capabilities over a network, then a new model presents itself where devices are decomposed into functional components which may then be reused in a much more flexible way. An architecture to support this model is proposed and presents a novel approach to device configuration called Task Oriented Programming (TOP) together with work on meeting the challenge of providing suitable user interfaces in the pervasive home.

INTRODUCTION

Trends in technology in the home

Over the last 5 years, we have seen a huge growth in broadband connections and use within the home. People have quickly come to realise the benefits of a fast, always on connection. This together with an ever-increasing quantity of consumer equipment, which either requires or is enhanced by a wired or wireless network means that the trend for a home network is following the broadband trend. This is confirmed in a report by Forrester (1) which states that 42% of PC owners having multiple PCs in the home with broadband also have a home network, whereas this figure drops to 16% for homeowners without broadband. Figures from the same report for single PC owners are 12% with broadband and 6% non-broadband. Reference to any recent computer-related magazine in a newsagent and you will see entertainment, multimedia, gaming devices and appliances on offer, together with routers, gateways, hubs and even networked storage devices. The latter trend being almost certainly a result of the popularity of digital photography and the need to store and retrieve easily large numbers of digital images.

Inevitably, these exciting technological trends, come with increasing complexity and confusion, especially for the new, non-technically minded home user. There is a long-standing anecdote about adults not being able to program the video recorder and leaving it to the children. On the positive side, user interfaces for such devices have become easier to use and technology like VIDEO Plus+® [1]have helped. However, growth of new consumer devices has far exceeded development of simpler user interfaces so the consumer is still left with a range of quite complex pieces of equipment each with their own interface style. Unfortunately, much of the functionality may not be available to the user because of the confusion caused by the complexity of system operation. As devices in the home become networked, this confusion becomes compounded with a range of proprietary interfaces and protocols.

Pervasive computing

The original vision of a person interacting with hundreds of devices connected via a wireless network is usually attributed to Mark Weiser (2), although he called it, in 1988, “ubiquitous computing”. As Weiser pointed out, this vision of course is in contrast to another topical research theme of the time, virtual reality. Instead of immersing people in a virtual environment produced by computers, bring the computing devices into the real world that people inhabit. Since then there have been many research projects following the ubiquitous computing theme and increasingly over the last few years people have begun to refer to the term “Pervasive ICT” as information and communications technologies merge.

It is this mix of increasingly complex and technology in the home with a desire to make technology more flexible and easier to use that has inspired the research project described in this paper. The vision that even in the home, devices will become smaller and more pervasive has motivated us to think how a future home network might be realised and, how will people interact with such technology? In the next section we elaborate on this vision and outline what our research has tried to achieve. After that we describe in detail the research and solutions we are investigating and this is followed by conclusions and suggestions for further work.

A VISION FOR PERVASIVE COMPUTING IN THE HOME

The General Vision

Pervasive environments are typically characterised by having a wide range and number of computing devices and sensors connected over a wired, or more commonly wireless network. The terms 'pervasive' and 'ubiquitous' imply that the technology is everywhere but also 'disappears' to some degree so that users do not feel invaded by it. Our vision is that this will be introduced into the home such that people can carry out simple or even quite complex tasks and actions. Users should not have to worry about the technicalities such as which device supports which file format or application, where a particular media file is stored, specific methods for configuration, etc. We extend this vision further by postulating that once devices are networked and are able to announce their capabilities and services then we can then 'decompose' devices into functional components, which can then be reused. This has two potential benefits:

- resilience – a device requiring a function that has failed can find an alternative on the network;
- *virtualisation* – 'devices' can be created from a

1VIDEO Plus+ is a registered trademark of Gemstar Development Corporation number of components, for example a 'TV' could be made up from a tuner device, a display device, speakers and a user interface.

This trend has already started with the emergence of flat panel displays and separate tuner boxes. For this environment to become usable, it must be:

- heterogeneous — non-proprietary, supporting many types of device and connectivity using open
- standard interfaces and protocols,
- distributed — devices contribute resources to the environment, and these resources can be combined to form ‘virtual’ devices — control should also be distributed to minimise dependency on a central point of failure, a key area of concern for consumers,
- auto-configuring — it must be easy for the user to add and remove facilities (actually, total auto-configuration is probably unachievable, but minimising user involvement should be the goal),
- self-healing — a reliable infrastructure is a key requirement — if resources disappear, either through failure or just because a user removes them, the environment should recover or at least degrade gracefully,
- secure — provide appropriate levels of privacy and security, which encompasses resource access
- control and security of information — there may also be safety considerations here, e.g. in a telemedical context.

Whilst this vision offers greater flexibility and cost savings to the consumer the product line of the white and brown goods manufacturers will have to change. Instead of bringing out re-packaged or re-styled products with different look and feel the manufacturer will need to offer basic functionality. The user interface (possibly branded) will be delivered by other parts of the system and internal functions will need to be made available to the network. Whether manufacturers will be willing to make available internal functionality that might be used by competitors, or whether they will be content for another vendor’s product to provide the user interface to their system remains to be seen.

THE PHEN PROJECT VISION

In this project we sought to explore issues relating to how devices and people might interact with pervasive networks in a home environment setting an aim embodied into name we gave the project; Pervasive Home Environment Networking (PHEN). At the middleware end of this work we have addressed devices and discovery protocols, finding that device descriptions are frequently minimal and very specific. In this work our views concurs with others that to progress towards our vision, descriptions need to be much richer, incorporating contextual information and ideas similar to those being developed for the semantic web (3), Horan (4). To these ends we have adopted an approach that introduces what we call an information layer, which supports the storing and retrieval of enriched device descriptions, ad-hoc data from users and applications, and a reasoning ability. From our experience we have found that the typical hierarchical file systems and monolithic applications, which are typical in the IT, space do not fit so well in the networked home. We describe in this paper an information layer architecture, which is based on a tuplespace solution. This is followed by a section describing our approach to placing the user in control and giving focus to the tasks the home inhabitants wish to carry out, rather then the technology through task oriented programming research from Essex University. As well as bringing some transparency (and hence trust) to the system, it also raises the possibly of empowering users to become designers of pervasive computing functionality thereby not just changing the computational model, but also expanding the usage model. While we are describing technology 'disappearing', there are inevitably points where users must and need to interact in a pervasive home environment. Consequently, it is important to have a flexible, context-aware approach to user interface design and implementation. This area of research from Loughborough University is described in detail. In the following sections we examine these issues in more detail.

THE PHEN ARCHITECTURE

The middleware (a development of UPnP and tuplespaces), task-oriented-computing (a task construction and execution engine) and the context aware interface (dynamic construction and termination) are built around a single conceptual architectural diagram shown in Figure 1, below. The figure shows the layered approach to the PHEN architecture with the coordinating middleware making up the PHEN-core supported by plug-in components which interface to the various home networking protocols such as UPnP and Bonjour. Above the core layer is the interface to TOP (see later) and other applications and user interfaces.

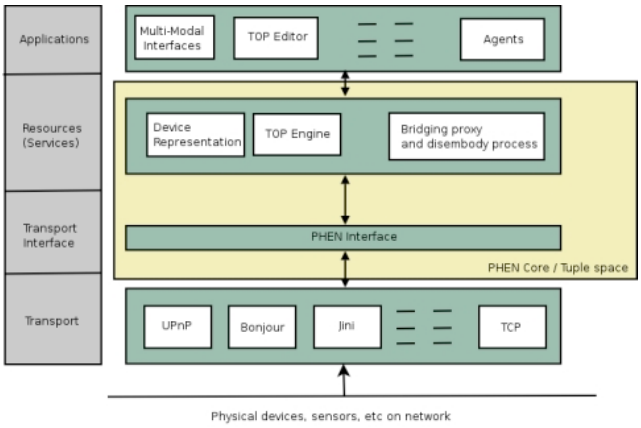

Figure 1 PHEN Architectural Diagram

Information Space

PHEN is centred on a shared information space that performs a number of roles:

- a repository for data from a variety of sources, e.g.:
  - information which users may want to store such as personal preferences
  - community descriptions
  - rule descriptions and rule sets
- a source and sink for data about devices and services available on the network
- a protocol adaptation layer which allows devices using a range of protocols to share and communicate
- an event model allowing services and applications to listen and respond to changes in state from other devices, services, etc.

In the introduction we stated that this is a popular area for research so it is appropriate to give some background on related work. AutoHAN (5) from Cambridge University has a number of similarities with our approach. They used universal plug and play (UPnP) (6), which the team extended to include semantic descriptions of the application that has been built into the devices. This description may be coded in a language developed for the purpose, called AutoHAN Language. The AutoHan team have also worked on user interfaces, concentrating on a remote control style of architecture based on some novel devices using infrared. This is one of the key differences with our work, where we want to provide a more ‘invisible’ architecture and use novel user interface and control techniques through rules, learned behaviour and embedded agents. MIT and others (including HP, Philips and Nokia) have been working on Project Oxygen (7). The user interface aspects are more in line with our work; their proposal is to move away from the need for users to have to use specific devices for user interaction (the mouse, keyboard, etc.) and move to more natural methods of interaction including speech and gesture. HP also have their own research project called CoolTown (8). The focus is on extending the now common experience of using the Web to a more ubiquitous computing model using hand-held devices, using for example infra-red communication. It appears that HP have reduced effort on CoolTown specifically, and taking their research in different but related directions. The New York University has developed a framework for pervasive computing called one.world, Grimm (9), which specifically supports application development.

Tuplespaces

The information space is based around a tuplespace model, which is based on an original idea by David Gelernter (10) who developed the Linda system at Yale University. Tuplespaces offer a number of features, which are advantageous and lend themselves to PHEN. These are:

- associative memory – data is addressed by content rather than address. Data can be retrieved using a pattern which does not have to be fully defined (cf wild card searching)
- logically shared – the space appears to be one entity but does not have to rely on one physical memory store. This makes distribution of the space easier and avoids the weaknesses of a central store.
- generative communication – this means that data stored as tuples in the tuplespace are not tied to a single process but may be shared and have loose temporal coupling

For PHEN we have implemented a tuplespace in Java which allows us to store data as tuples. The tuples can be as simple as string-based name-value pairs, or the value can be Java objects providing a more complex data structure to be stored. Retrieval can be specific based on known name/values or can be based on a pattern. For example we might store information about an audio player as follows:

Tuple:
name=audioplayer1
location=lounge
serialnumber=01234567
ip-address=192.168.0.10
status=playing
format=mp3,ogg,wma

and a table lamp as:

Tuple:

name=tablelamp1
location=lounge
status=off
dimmable=false
control=X10

If the tuplespace is searched with a query tuple defined as:

Query Tuple:
location=lounge

then both tuples are returned. Alternatively, a more specific query would return one or other of the tuples. We have included an event model such that subscribing to the tuplespace based on a query tuple will inform via a callback function that informs the subscribing application or service when changes based on the query occur. The application can then make decisions based on that event.

The tuplespace read/write API follows closely the model defined by Gelernter, which is based on just four basic operations. Our API uses *write*, *read*, *readMultiple* and *take*. The first two operations write and read tuples, readMultiple will retrieve a list of tuples based on the query tuple pattern and take is like read but will also remove the tuple from the tuplespace.

Device and Service Discovery

Earlier, we described devices communicating with each other and looking beyond simple device description to the point where individual services within a device can be connected and re-used. This leads to device decomposition and thus the potential to create virtual devices from components on the network. A key enabler for the pervasive home environment is that devices and services announce their availability and capabilities. There are a number of methods available to achieve this including Jini (11), UPnP (Universal Plug and Play) (6), Bonjour (previously know as Rendezvous) (12), SLP (Service Location Protocol) (13) and Bluetooth (14) (Salutation has often also been cited, however the Salutation Consortium was dissolved in June 2005). Each of these protocols have similarities, differences, strengths and weaknesses, and it is not within the scope of the research being described in this paper to provide a comparison, such work has been reported before (15), (16).
Our research has primarily used UPnP for device and service discovery because it is a fairly simple protocol, based on HTTP and XML and we are already seeing devices available, which support UPnP (notably broadband router/gateways and networked cameras). As part of the information space architecture we have built a component which bridges between UPnP and the tuplespace. The reasons for this are twofold. Firstly, we recognize that there will be other discovery protocols so providing a bridging mechanism allows us to have devices communicate and share information via the protocol-agnostic tuplespace. Secondly we believe that implementation of PHEN requires a richer, contextual environment which device descriptions alone can not achieve. The bridge allows us to capture fundamental descriptions about the device and store that along with other information (possibly provided by other devices and sensors or by users) in the tuplespace. Since both UPnP and the tuplespace have eventing mechanisms, these can also be linked via the bridge. Devices which support UPnP which are connected to the network in our environment will announce themselves and be automatically discovered via the bridge and tuples capturing the information placed in the tuplespace.

End-User Programming

In earlier work Essex University conducted extensive research into the use of autonomous intelligent embedded agents to produce autonomous self-programming environments in which the users roles and cognitive loading was minimised, thereby making the technology easy to use (17). However, whilst it was found that autonomous agents appealed to many people, their acceptance was not universal. Some lay-people distrust autonomous agents and prefer to exercise direct control over what is being learnt, when it is being learnt and to whom (or what) the information is communicated. These concerns are particularly acute when such technology is in the private space of our homes. Moreover, there are other reasons advanced in support of a more human driven involvement, such as releasing the creative talents of people by providing them with a means to become “designers of their own “pervasive computing spaces”, whilst at the same time, shielding them from unnecessary technical details. To explore this aspect of digital home work Essex has developed technique we refer to as Task Oriented Programming (TOP), which is loosely based on a combination of Programming-By-Example (PBE), pioneered by Smith (18) in the mid-seventies and Lieberman (19) in the 90s and Learning-From-the User (LFU), the paradigm Essex University has been developing for many years (17). It has some conceptual similarities to the notion of Macros but differs significantly to all these previous approaches in that spawns non-terminating processes and addresses the orchestration of real physically distributed computing devices rather than running on single processors or manipulating abstracted entities. It is based on a vision to put the user at the centre of the system programming experience by exchanging implicit autonomous learning for explicit user driven teaching. In this approach a user defines a community of coordinating pervasive devices and then "programs" it by physically operating the system to demonstrate the required behaviour (20).

Task-Oriented-Programming (TOP)

TOP was proposed and developed by Chin in 2003 as a means to address the issues of privacy and creativity in digital homes (21). In the TOP approach, the system is explicitly put into a learning mode and taught (by demonstration) how to behave by the lay end-user. For example the TV or sitting room light could be made to react to an incoming call on the telephone. Thus the telephone, TV and light coordinate their actions to form a new meta (virtual) appliance. The vision goes beyond linking only conventional appliances. For instance if a network capability is added to an appliance, it becomes possible to allow its functional units to be shared with others if appropriate. By exposing appliance functionality to other systems it is feasible to offer system features that would otherwise be impossible. Likewise, autonomous intelligent embedded agents are simply regarded as functional resources that the end-user can chose, or nor, in the system being created. Thus by way of a simple example, the audio amplifier in a TV could be made use of by the Hi-Fi system, or vice versa. Consequently, "virtual appliances" could be created by establishing logical connections between the sub-functions of appliances, creating replicas of traditional appliances, or inventing altogether new appliances. This decomposition of traditional appliances into their atomic functionality (either physically or logically) and later allowing users to re-compose "virtual appliances" (nuclear functions) by simply reconnecting basic atomic functionality together is the paradigm we called: "the deconstructed appliance" model. This has some conceptual parallels to the use of dis-aggregation of PC services in Microsoft's Easy Living (22) and aggregation of utility service provision being explored in the Centre for the Integrated Home Service Aggregation project CIHE (23) but differs significantly in that in PHEN addresses dis-aggregation of real physical embedded computer devices. The key to creating "virtual appliances" from decomposed functions is that of making connections between sub-functions so that a closed set of interconnected functions becomes a global set of functions (ie, it becomes a "community", or a collective of coordinating devices with a meta functionality). Clearly, this concept of "community" is not limited to decomposed appliances, but relates to any set of coordinated pervasive entities, whatever their functional or physical level (eg it could also relate to nano-scale or even micro-scale building-to-building environments). In general, a richer the pool of sub-functions will lead to greater combinations or permutations for the user to create new virtual appliances.

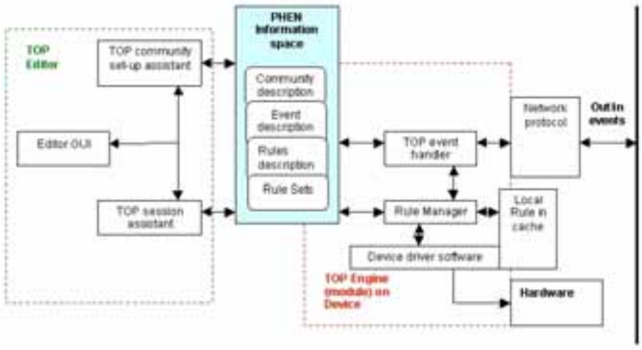

Figure 2 - The TOP Architecture

The TOP architecture, Figure 2, has two distinct modules; a "Top Editor" (to program the systems) and a "TOP Engine" (to execute the user generated rules). The first module, the TOP Editor has 2 components – (A) "TOP community set-up assistant" that allows the user to set up groups (communities) of devices that can communicate and coordinate their actions to produce some desired meta function (or virtual appliance), and (B) the "setup assistant" component that runs at the background managing TOP sessions; its role is to capture the user's on screen activity and generate data accordingly. The second module, the TOP Engine is a process that runs inside each and every networked device and executes the taught rules This module has three main components;

1. "TOP event manager", that handles events generated either from the internal components or across the network,
2. "Rule Manager", manages the addition and removal of rules from memory
3. "Local Rule cache", that acts as a temporary rule buffer whilst rules are being built by the user (i.e. while the user is still designing and experimenting with creating community functionality).

To facilitate the information to be used within and beyond the community, the data format needs to be standardised so that all other parties in the network can understand it. Community configurations and behaviours described in such a way then become portable between environments and tradable commodities in their own right. Such descriptions can be used to provide default functionality for environments and appliances, so that programming from scratch is not necessary. This aspect of the work can be handled in various ways; for instance we have developed a supporting ontology (dComp) and a formal methodology for visualisation and analysis described in papers by Chin (24) and Zamudio (25).

Dynamic context-aware user interfaces

As discussed earlier in the paper, a key part of the PHEN vision is the decomposition of devices. Devices are broken down into components, which communicate using the PHEN tuplespace based middleware. Virtual appliances can then be formed by aggregating different components. This decomposition and aggregation of devices to form virtual appliances has implications for both the user interfaces and user interaction in general. It introduces the requirement for user interfaces to be dynamic (reconfigurable) so as to handle the dynamic nature of virtual appliances, which can be easily created, destroyed or reconfigured. User interfaces also need to be able to cope with multimodal interaction in the home environment because it is unfeasible to expect people to interact with all devices using just a keyboard, mouse and screen. To address these challenges Loughborough University have investigated the use of context-awareness and multimodality to provide user interfaces for devices in the pervasive home environment.

Context-Aware Multimodal Interaction

For computer based control systems and user interfaces to work efficiently in dynamic reconfigurable environments, such as the home, they need to be able to react to changes in system state. To achieve this they need to be context-aware. In effect the system needs state information to assist devices and applications in making decisions. Additionally for a system to be useable in the home requires that the user can interface with it in a natural, non-invasive manner. This requires that the system provide multiple modes of interaction (multimodality) which are suitable for interaction within a home environment.

Loughborough University have researched the use of multimodality and context-awareness to provide dynamic user interfaces for systems within the home. This work has tied together a number of different projects to produce a multimodal context-aware architecture for providing interfaces for services and devices. The architecture has provision for gathering contextual information, providing user interfaces, and multimodal input. To help user interaction particular contextual information needs to be captured. The system needs to know where users are and the goal they are trying to achieve. To aid in locating users sensors in the home such as PIR sensors input data into the PHEN tuplespace, this can then be easily accessed and interpreted to provide user location information to services and devices. Obtaining the context of the user's goal is achieved through a voice driven ontological/context-based menu system (developed as part of another project). Due to the nature of the system being used within the home environment multiple modes of interaction are required for the system to be effective and users to accept the system. Therefore, the system developed allowed for a number of input and output methods. The input modes for users were; hand held remote control, voice, touch screen and the modes of output were; set-top box / TV, LCD screens, voice, small digital displays, mirrors and ambient displays. These input and output devices are implemented as PHEN devices which allows for them to be easily swapped in and out. The developed architecture allows for the registering and removal of PHEN devices.

XUL based Interfaces

In addition to the multimodal context-aware interaction architecture a simple XUL (26) user interface architecture was developed. XML User Interface Language (XUL) is an XML based language for creating rich graphical user interfaces. Originally developed as part of the Mozilla project (27) XUL allows interfaces to be specified in XML separate from the programming logic. This allows for user interfaces to be dynamically created or modified without the need to change a programs source code. The appearance of XUL interfaces can also be quickly and easily modified using CSS, XSLT, XPath and DOM features. Separating the user interface description from the application has the advantage of allowing GUIs to be modified quickly and having multiple GUI designs. This modification allows branding to be added or the 'look and feel' to be easily changed.

The XUL based interface system allowed PHEN services to place a tuple into the tuplespace, which contained a XUL description of its standard user interface. Together with the XUL description the service places a serialized Java object into the tuplespace. On the client machine a simple Java application is run which uses the Luxor XUL toolkit to generate a Java Swing GUI from the XUL description. The application then uses the serialized Java object to connect the GUI to the PHEN service using the tuplespace for communication.

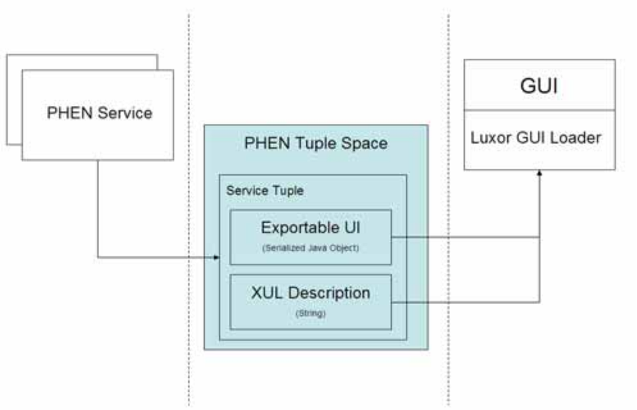

Figure 3 – XUL based interface generation

IMPLEMENTATION & EVALUATION

The iDorm

The intelligent dormitory (iDorm) shown in Figure 4 is a real pervasive computing test-bed comprising of a large number of embedded sensors, actuators, processors and networks in the form of a full size apartment (a digital home). It is a multi-use, multi-user space providing areas for different activities such as sleep, work and entertaining. It contains the normal mix of furniture found in a typical home. The iDorm was built from the ground up to be an experimental pervasive computing environment with many special structural features such as cavity walls/ceilings containing power & network outlets together with provision for internal wall based sensors and processors etc. There are numerous networks in place ranging from wired and power-line through wireless to broadband and high-bandwidth multi-mode fibre connections to the outside world. All the basic services are electrically controlled wherever possible (eg heating, water doors etc).

Figure 4 – iDorm2

A common interface to the iDorm and its devices is implemented through Universal Plug and Play (UPnP) which is an event-based communication middleware that allows devices to plug and play thus enabling automatic discovery and configuration. A gateway server is used to run the UPnP software devices that interface with the hardware devices on their respective networks. Our experimental agent mechanisms are built on top of the low level UPnP control architecture enabling it to communicate with the UPnP devices in the iDorm and thus allowing it to monitor and control these devices. Figure 5 shows the logical network infrastructure of the iDorm.

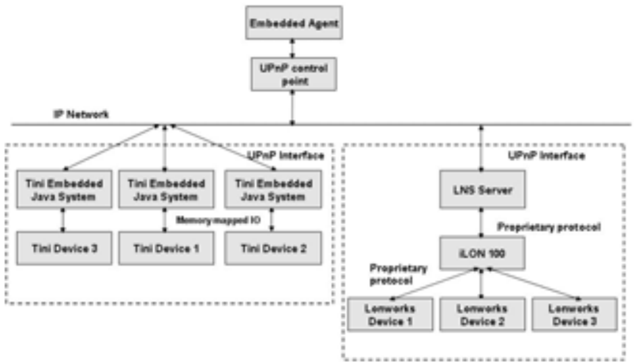

Figure 5 – iDorm2 logical network infrastructure

TOP Evaluation

The TOP evaluation consisted of eighteen participants drawn from a diverse set of backgrounds (eg housewives, students, secretaries, teachers etc). Their genders were 10 females and 8 males and their ages ranged from 22 to 65. The participants also formed a multicultural group including Asians, Europeans and Australians. All participants have some computing experience (ie. they knew how to use a mouse). Whilst 21.3% of the participants had a very good knowledge of programming, 57.4% of them have none at all. For the evaluation sessions they were given five sets of devices (drawn from a set of lights, a telephone, smart sofa and an MP3 player). During the evaluation, no specific tasks were set for the participants but they were encouraged to create their own desired environment based on the devices available. The evaluation was preceded by a 20 minute training session and evaluated via a proceeding usability questionnaire.

Figure 6 – TOP evaluation

The questionnaire was developed, with the assistance of Chimera (a socio-technical research unit based on the BT campus; see www.essex.ac.uk/chimera) to assess the participants' views on the usability of TOP. It consisted of a set of seventeen statements, measuring attitudes over six usability dimensions: "The TOP Concept", "User Control", "Cognitive Load", "Information Presentation", "Affective Experience" and "Future Thoughts". The questionnaire used a five-point Likert scale with responses from "Strongly Agree" through to "Strongly Disagree". Each of the usability dimensions consisted of a series of statements (from 2 to 4) with each offering a range of ratings (from 1 to 5). A higher rating score on the dimensions contributes towards the greater usability of TOP.

TOP Evaluation Results

It is not possible to present all the evaluation data, nor the results of the comprehensive analysis carried out using the SPSS software package in this paper, therefore highlights of the results will be presented in the following paragraphs. A summary of overall user ratings for the six dimensions are displayed in figure 7 and table 1.

In contrast to alternatives, the TOP approach gives people explicit control of what is learnt and when, a feature that was felt to be important to enable such technology to be accepted in the most private spaces of our lives; our homes. One of the penalties associated with explicit control is the amount of manual effort involved. At the outset of the TOP work, one of our contentions for TOP is that people would enjoy the manual experience of programming and not find it relatively easy. Both of these assertions were supported by the evaluation as, in terms of enjoying the experience, the mean of the affective dimension was 4.6 (the highest rating) indicating people greatly enjoyed the experience of programming using TOP, whilst the cognitive load dimensions had an overall average of 4.3 indicating people found the process relatively simple. In fact, it was found that after a brief 20-minutes training, 88.9% reported that they used the controls with ease and 83% of participants were able to use TOP to create their desired environments with little or no assistance.

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean	
					Lower Bound	Upper Bound
Conceptual	113	4.3186	.53894	.05070	4.2181	4.4190
UserControl	191	4.1990	.59134	.04279	4.1146	4.2834
CognitiveLoad	155	4.2710	.57332	.04605	4.1800	4.3619
InformationRetrieval	112	4.4107	.54613	.05160	4.3085	4.5130
AffectiveExperience	240	4.6083	.50596	.03266	4.5440	4.6727
FutureThoughts	83	4.1687	.76221	.08366	4.0022	4.3351
Total	894	4.3602	.59489	.01990	4.3211	4.3992

Table 1 - Overall TOP User Ratings

Table 1 – TOP evaluation Dimension Rating

In more detail, where there was variation on cognitive loading across user groups, it was relatively minor and cross-section tests suggested this might be related to age (older participants feeling a slightly heavier cognitive load than their younger counterparts) but more research would be needed before such a conclusion could be confirmed. In general the “Information Retrieval” dimension (how well information was presented to the user) scored the least but was still in excess of 4 indicating that overall people found TOP to be both useful and enjoyable.

Of the differing groups of participants, the non-programmers were the key target users for TOP. Thus we were pleased to find that this group performed well with only 4 out of 199 cases evaluated having negative responses (2%). Remarks recorded from this group included: “*I just feel like right now I want to sit down for a lot longer and try out all sorts of environment that I could possibly create*!” and another one : “*I can really get quite keen on it*” were just the few examples of the positive subjective judgements.

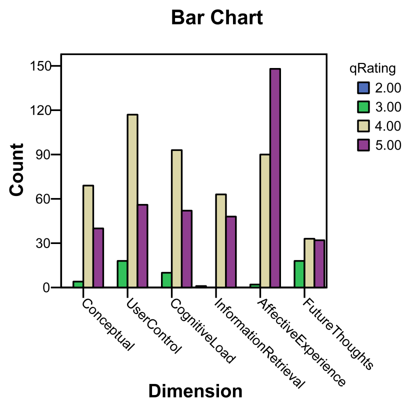

Figure 7 - TOP evaluation Dimension Rating

None of the participants found it difficult to understand the basic principles of the system. A remark from one participant stated “*I thought the basic principles themselves are very simple and straight forward. I felt I could easily grasp the basic principles*” was typical of many users. Given this particular remark was from the group with only average experience of computing and no programming skills at all, this was particularly heartening (one original goal for TOP/PHEN was to facilitate use by non-programmers). Overall 83.4% of all participants found TOP intuitive to use and 94.4% of all participants stated they felt rewarding when they used the system.

Thus, based on the results gathered, TOP was concluded to be a success in meeting its original goals for enabling non-specialists to be able, and to enjoy, programming coordinated actions of distributed embedded computer systems that make up PHEN.

Middleware Demonstrator

In addition to the TOP evaluation just described, demonstrations of device and user interaction, coordinated by the tuplespace have been developed. One of the first was based around a multiple occupancy home environment and a few audio player devices. This was able to demonstrate how commercially available audio players could be integrated into a pervasive environment such that for example, when a person enters a room, music plays according to their personal preferences. If another person joins them, then the music can automatically reduced in volume. Device availability, status and location; user location and preferences, play lists, etc are all stored as tuples. The original version of this demonstrator had 'hard coded' rules (eg for reducing the volume as described). Work is now in progress to encode these rules separately from the application logic and link this with the work on TOP. A number of ways of achieving this are being investigated.

SUMMARY

In this paper we have described work in progress but, even at this intermediate stage of our project PHEN project has successfully demonstrated the value of a set of heterogeneous services and appliances being distributed or linked through non-proprietary connectivity using open standard interfaces and protocols. The benefits of distributed devices contributing resources to the environment such that these resources can be combined to form ‘virtual’ devices is an important factor for the future integrated home environment. User access or control has been shown to be efficient and easy to comprehend with the distributed system. This minimises dependency on a central point of failure, a key area of concern for consumers, Moreover, our original concern that users may become confused or unable to comprehend this new way of thinking about devices in the home has been shown by our user evaluation to be unfounded (rather the opposite, users have enjoyed the freedom, creativity and control these tools provide). Whether manufacturers of future white/brown goods will be willing to make internal functions available or to let go of their interface with the user (through the traditional user front panel or display) remains to be seen. However, it is likely that the user will still be able to control their devices through traditional user interfaces but the real benefit comes from the ability to link network enabled devices together. PHEN will facilitate new user interaction paradigms by adapting to specific user needs or changing contexts and enabling a whole new population of non-technical users to become competent programmers of pervasive computing systems. Ongoing research is underway to develop the knowledge representation space, diversify the modal interaction of TOP and extend the existing PHEN architecture and apply this to exemplar environments.

We are pleased to acknowledge the DTI’s NextWave Technologies and Markets programme which, via the “Centre for the Integrated Home Environment” supported the PHEN project. We also acknowledge the work, support and advice of Graham Clarke (Essex University), Alex Loffler (BT) and people from the Smart user Interfaces project also under CIHE.

REFERENCES

1. Forrester, September 2005, Consumer Technographics Survey, www.forrester.com

2. Weiser, M., 1991, "The Computer for the Twenty-First Century." Scientific American, 1991, 94-100, http://www.ubiq.com/hypertext/weiser/UbiHome.html

3. Semantic Web — http://www.w3.org/2001/sw/

4. Horan B., 2005, “The Use of Capability Descriptions in a Wireless Transducer Network”, Sun Microsystems Research Labs, Report Number: TR-2005-131

5. Blackwell A. F., Hague R., 2001, ‘AutoHAN: an architecture for programming the home’, Proceedings of the IEEE Symposia on Human-Centric Computing Languages and Environments, pp 150-157

6. UPnP, http://www.upnp.org/

7. Project Oxygen, http://oxygen.lcs.mit.edu/

8. Debaty P., Goddi P., Vorbau A., 2003, “Cooltown, integrating the physical world with the Web to enable context-enhanced services”, HP Technical report: HPL-2003-192, http://www.hpl.hp.com/techreports/2003/HPL-2003-192.pdf

9. Grimm R., Davis J., Lemar E., MacBeth A., Swanson S., Anderson T., Bershad B., Borriello G., Gribble S., and Wetherall D. 2004, “System support for pervasive applications.” ACM Transactions on Computer Systems, 22(4):421-486, http://www.cs.nyu.edu/rgrimm/one.world/papers/tocs04.pdf

10. Gelernter D., 1985, “Generative communication in Linda”, ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 7 , Issue 1, http://portal.acm.org/citation.cfm?id=2433

11. Jini – http://www.sun.com/jini/

12. Bonjour, http://developer.apple.com/networking/bonjour/

13. Service Location Protocol, http://www.ietf.org/rfc/rfc2608.txt

14. Bluetooth, http://www.bluetooth.com/

15. Bettstetter C., Renner C., 2000, “A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol”. In Proceedings EUNICE 2000, Sixth EUNICE Open European Summer School, Twente, Netherlands

16. Rekesh J., 1999, “UPnP, Jini and Salutaion - A look at some popular coordination frameworks for future network devices”. Technical report, California Software Labs, http://www.cswl.com/whiteppr/tech/upnp.html

17. Callaghan V, Colley M, Hagras H, Chin J, Doctor F, Clarke G, Programming iSpaces: A Tale of Two Paradigms, in book Intelligent Spaces: The Application of Pervasive ICT part of the series Computer Communications and Networks Steventon, Alan; Wright, Steve (Eds.) approx. 455 p. 162 illus. ISBN: 1-84628-002-8, Dec 2005

18. Smith, D. C., 1977, “Pygmalion: A Computer Program to Model and Stimulate Creative Thought”, Basel, Stuttgart, Birkhauser Verlag.

19. Lieberman H, “Your wish is my command”, Program by Example, Morgan Kaufmann press, 2001.

20. Chin J., Callaghan V., Colley M., Hagras H., Clarke G., 2005, “End-User Programming in Pervasive Computing Environments”, In Pervasive Systems and Computing conference,(PSC-05), Nevada, USA, June 27-30, 2005

21. Chin J.S.Y., Callaghan V., Clarke G., Hagras H., Colley M., 2004, “Pervasive Computing and Urban Development: Issues for the Individual and Society” UN Second World Urban International Conference on “The Role of Cities in an Information Age”, Barcelona, Spain.

22. Brumitt B., Meyers B., Krumm J., Kern A., and Shafer S. 2000, "EasyLiving: Technologies for Intelligent Environments". Handheld and Ubiquitous Computing

23. CIHE, “Centre for the Integrated Home Environment”, http://www.cihe.lboro.ac.uk/

24. Chin J, Callaghan V, Colley M, Hagras H, Clarke G. “Virtual Appliances for Pervasive Computing: A Deconstructionist, Ontology based, Programming- By-Example Approach”, IE05, Colchester, UK, 28-29th June 2005

25. Zamudio V., Callaghan V., Chin J., 2005, “A Multi-Dimensional Model for Task Representation and Allocation in Intelligent Environments”, The Second International Symposium on Ubiquitous Intelligence and Smart Worlds (UISW2005), Nagasaki, Japan, Dec 6-7th 2005

26. XUL, 2001, “User Interface Language Specification”, http://www.mozilla.org/projects/xul/xul.html

27. Mozilla Project, http://www.mozilla.org/