Presented at the 2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002), May 21-24 2002, Berlin, Germany

The iDorm – a Practical Deployment of Grid Technology

Anthony Pounds-Cornish, Arran Holmes
Intelligent Interactive Environments Group, University of Essex, UK
apound@essex.ac.uk, acholm@essex.ac.uk

Abstract

A purpose built environmental testbed for Intelligent Interactive Environments research, based upon student accommodation (iDorm) is described. It is argued to be a practical example of Grid technology. The testbed combines several different networks and provides a common protocol to act as a gateway between the different sensors and effectors in the iDorm. A variety of interfaces to allow users to control and monitor the different services are described and critiqued. Future plans for extensions to the iDorm are described along with future plans for research experiments using intelligent agents as monitoring tools and interfaces themselves.

1. Introduction

The estimated number of microprocessors built in 2001 is over 8 billion [1] and today, computers outnumber the world's population. This continual rise in the presence of technology means computing, networking and connectivity plays an increasing role in our lives both at home and at work, whether we want it to or not. These circumstances cause the human cognitive load to increase drastically – the majority of everyday tasks now involve interfacing with technology of varying complexities.

The IIE Research Group at the University of Essex [2] is researching new ways for mankind to interface with the technology that surrounds them. These interfaces offer ubiquitous, novel methods of communicating with the environment both locally and remotely. They provide a generalised interaction model enabling the user to control the surrounding technology regardless of its type or communication protocol.

The ideas and concepts produced by our work have been brought to fruition in a purpose built environmental testbed nicknamed the iDorm (Intelligent Dormitory), which is used by our group to run experiments. The iDorm represents an application of the Grid architecture, whereby the effecters and sensors found inside represent different services within the room. These services, having different functionality, represent the problem of how to access this grid of services regardless of physical location.

This paper will outline how the iDorm was designed and the reasoning behind those decisions. It will compare the different communication networks that are used in the iDorm. It will also cover the different methods of interaction with the environment, how they relate to each other and how the remote and local access to the service Grid is achieved. The final part of the paper includes details of our future work including using the results taken from the iDorm and using them to design and implement a larger, multi-room environment and experiments involving a student living in the iDorm for an extended period.

2. The iDorm

The Intelligent Dormitory was an office in the Computer Science Department at the University of Essex. It has been converted into a student dormitory, the design of which is based on the campus accommodation at Essex; the same suppliers provided furniture and fittings. The iDorm's contents include a desk, bed, wardrobe, chest of drawers and a PC (Figure 1).

Figure 1. The Intelligent Dormitory (iDorm)

On first inspection, the iDorm looks like any other student dormitory. However, floor level trunking and a false ceiling hide a myriad of sensors and effectors as well as several different communication networks. These networks are connected to the switches and buttons on the wall that represent the standard interface to an environment found in most homes and offices. A list of these devices can be found in Table 1.

Table 1. Controllable devices in the iDorm

Light	Table lamp
	Bedside lamp
	Four ceiling spotlights
	Motorised vertical blind
Temperature	Heating Fan
	Cooling Fan
Access	Electronic door lock

The iDorm was built to fulfil the need of academics at Essex who had many research topics involved in intelligent inhabited environments. The ideas and concepts, ranging from interface design to intelligent environmental control agents lacked a framework to experiment in the real world instead of computer-simulated environments.

3. Communications

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

LonTalk is an off-the-shelf communications network designed for intelligent buildings [3]. 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 network is laid out in a ring arrangement; there is no central server. Each device is individually addressable on the network and is capable of sending and receiving messages from other devices connected to the same network. A list of standalone sensors attached to the LonTalk network is in Table 2.

Table 2. Sensors attached to LonTalk network

Light sensor	Cooling Fan
Occupancy sensor	Heating Fan
Smoke detector	Temperature Sensor
Humidity sensor	

These sensors are in addition to the embedded ones, which enable each effecter to report its status.

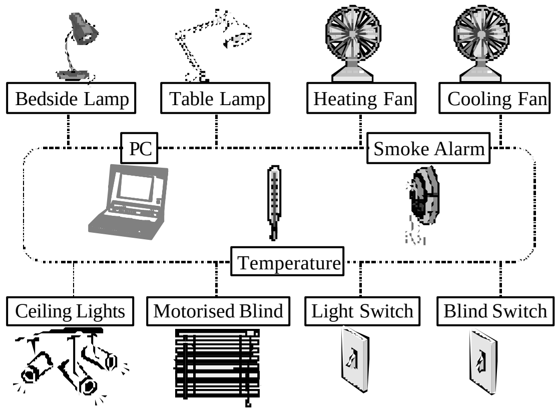


Figure 2. LonTalk network layout

The network is configured using design software on a PC. The PC is then temporarily connected to the LonTalk network and the design is embedded in all the devices. When the PC is disconnected, the LonTalk network runs unattended.

The LonTalk network is very scalable, in the iDorm application it is used in a single room but there are commercial applications that run a single network across an entire building [4].

1-Wire

The 1-Wire protocol has been designed and implemented by Dallas Semiconductors [5]. It is designed for small-scale applications where the distances between devices on the network are relatively small. Unlike LonTalk, this network requires a central server that takes the form of a small piece of hardware called a Tiny Internet Interface board (TINI). 1-Wire devices are powered from the network and are very small (usually an area of less than an inch).

The iDorm's 1-Wire network runs through the trunking around the edge of the floor and through the false ceiling. At various points along the trunking, 12 temperature sensors are attached to the network. They are laid out to give a temperature gradient across the room and a selection of the sensors give readings from outside the four walls of the room.

The TINI board acts as a gateway to the 1-Wire network. It is addressable on the IP as well as on the 1-Wire network. A Java Virtual Machine is embedded on the board and the research group has written a small server that accepts HTTP requests for the temperature information and returns it as an HTML web page (Figure 3).

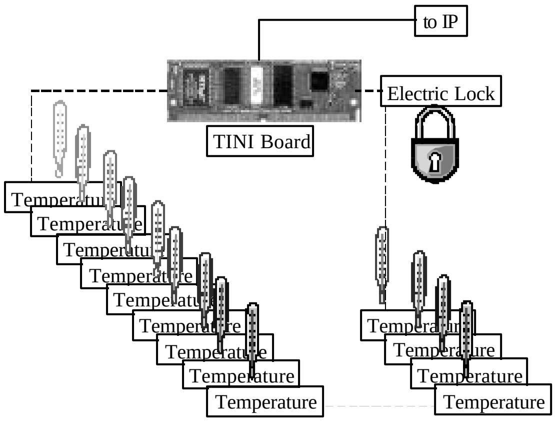


Figure 3. 1-Wire network layout

Because the devices all draw power from the same network, there is a limit to the number that can be connected to it. The advantage of the 1-Wire devices is that they are very cheap and very small. In contrast to LonTalk, the 1-Wire network also allows plug and play operation. The TINI Java program that the research group has written means that temperature sensors can be added to or removed from the network whilst the server is running without interruption to the temperature service.

IPv4

The group decided that a single network was required to be able to link the several 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 2) and the 1-Wire network (Figure 3).

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.

All of the interfaces that the IIE group has produced connect to the central server across the IP network. The interfaces can request information about the iDorm in the form of an empty XML tag pair. This pair then gets returned to the interface with the blank parts of the request filled with the appropriate data. The XML definition is stored on the central server and can be retrieved by any interface. This allows the definition to be updated without affecting the functionality of the interface mechanism.

Because the central server receives and transmits information using the HTTP protocol, we drastically increase the hardware platforms that interfaces can run on. These facts lead the group to design and implement a variety of interfaces to the iDorm, all of which use HTTP across IPv4.

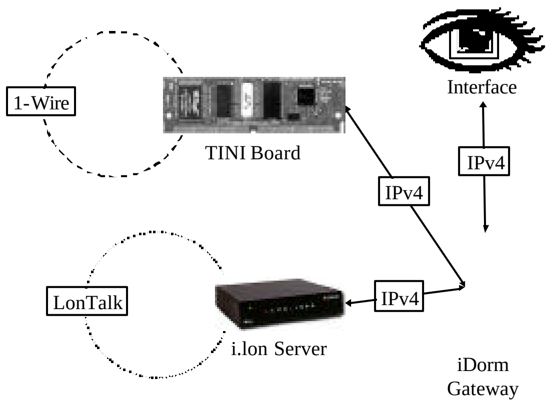


Figure 4. The high-level network

4. Interfaces

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

The Standard Interface

As mentioned previously, 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.

Each effecter on LonTalk is initially configured to listen to data packets from certain devices. For instance, the blind is initially configured to respond to the buttons on the right of the blind (see Figure 1). However, the network could be reconfigured to make the various buttons across the room control different devices.

Plans are underway to make this interface intelligent. The group is looking into ways of making a set of switches dynamic in what they control. For instance if a user is near a bank of switches and the system can work out they wish to change the state of the blind by monitoring their previous actions, the behaviour of those switches closest to the user will change to control the blind.

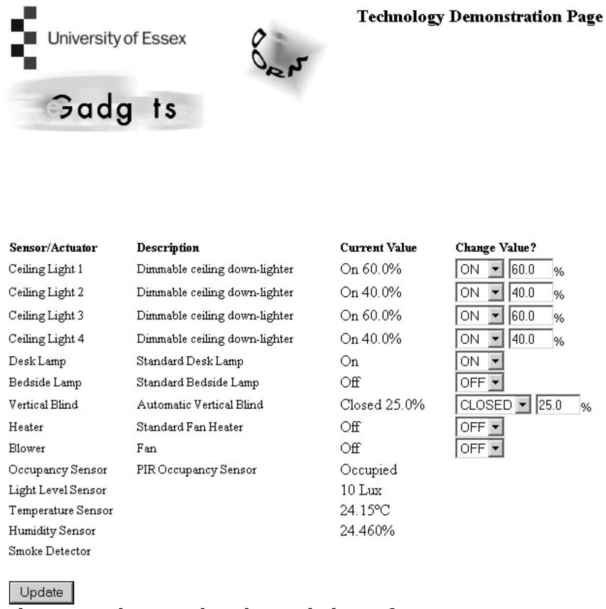
The Web Interface

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 (Figure 5).

This interface allows access to the iDorm's status and functions from any machine connected to the Internet. It means that the room can be controlled from a place of work or on the move from a laptop.

The iDorm central server deals with the security issues raised by this. It is set up only to respond to requests from certain machines individually identifiable by their network address. This list of "trusted" machines could be edited on the local server to remove or allow new machines to adjust the room's environment.

This interface cannot take advantage of the fact that the iDorm is dynamically extensible. Any new devices added to the room cause a re-write of the web interface. This led the research group to concentrate on a different interface that incorporated this extensibility.



Sensor/Actuator	Description	Current Value	Change Value?
Ceiling Light 1	Dimmable ceiling down-lighter	On 60.0%	ON 60.0 %
Ceiling Light 2	Dimmable ceiling down-lighter	On 40.0%	ON 40.0 %
Ceiling Light 3	Dimmable ceiling down-lighter	On 60.0%	ON 60.0 %
Ceiling Light 4	Dimmable ceiling down-lighter	On 40.0%	ON 40.0 %
Desk Lamp	Standard Desk Lamp	On	ON
Bedside Lamp	Standard Bedside Lamp	Off	OFF
Vertical Blind	Automatic Vertical Blind	Closed 25.0%	CLOSED 25.0 %
Heater	Standard Fan Heater	Off	OFF
Blower	Fan	Off	OFF
Occupancy Sensor	PIR Occupancy Sensor	Occupied	
Light Level Sensor		10 Lux	
Temperature Sensor		24.15°C	
Humidity Sensor		24.460%	
Smoke Detector			

Figure 5. iDorm basic web interface

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 6). 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. By clicking on the bedside lamp in the VRML, the user can toggle the status of the bedside lamp in the real world. The model is also sophisticated enough to update itself based on what changes are made to the room outside of its own control.

For instance, if someone else uses the standard interface to make changes to the lights, the lighting inside the model will update in real time. This is possible because the model uses Java to regularly interrogate the iDorm central server for the current status of the room. When it sees a change in the environment, it updates itself.

The drawback of this interface is that only physical environmental data can be shown. For instance light and blind position can be easily shown, however changes in temperature and the status of the heating and cooling fans are practically impossible to reflect in a 3D model – the user cannot "feel" the temperature when using the VRML. Current work on this interface includes for instance dynamically changing the colour of the walls in the model to reflect the temperature; red for hot, blue for cold. It is hoped that this will give a far better representation of temperature across the room than a table of numbers.

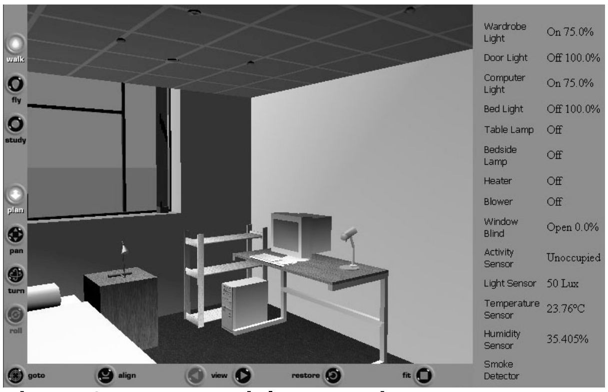


Figure 6. VRML model screenshot

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 view the current temperature across the room through a mobile device that supports WAP. Such devices include palmtops and mobile phones.

Because of the difficulties in authenticating mobile phones as "trusted" devices, the WAP interface is simply a method of viewing the thermal status of the room. It is not possible to request updates to the room through this interface. However this interface is a good example of accessing the iDorm services remotely. Anywhere the portable device (e.g. mobile phone) has a signal, it is possible to check on the status of the iDorm.

The future for this interface includes creating a two-way system where changes can be made to the iDorm. Plans are also to have a text messaging functionality where the user can request specific information or changes without a continuous connection. Such technology could allow access to the iDorm based on the receipt of a text message from a certain phone, thus identifying the user based on their mobile device.

Figure 7. WAP Interface View

Voice Recognition Interface

Prof. Nikola Kasabov and Waleed Abdulla from the University of Otago in New Zealand originated a speaker independent voice recognition system. The research group is applying it using a room-based command set appropriate to the iDorm.

Based on Hidden Markov Models, the system contains fourteen commands (listed in Table 3). 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.

Table 3. HMM Voice Commands

Table lamp on	Table lamp off
Brighter	Dimmer
Blind open	Blind close
Window open	Window close
Window half	Room sleep
Light on	Light off
Door open	Door close

The voice recognition system is implemented in the MATLAB language and runs on the local PC in the iDorm. It is attached to a wireless microphone that allows the user to give commands regardless of their physical location. The advantage of this voice recognition system is that it is adaptable across different ages, gender and accents with little training. The current system was trained on 15 examples of each command from two people.

The problem with this interface is that although the effect of the commands is dynamic, the user must have prior knowledge of the voice commands for the room. Although the research group have attempted to make the voice commands intuitive, there is no convention for this sort of control mechanism.

The future plans for this interface include making the effect of the voice commands more dynamic. An example of this is using the outside temperature of the room in deciding whether opening the window will lower or raise the internal temperature of the room more quickly than using the air fans.

5. Conclusions

The different interfaces that have been implemented for the iDorm provide interesting conclusions about their use and functionality.

The iDorm, although limited in the number of sensors and effecters, provides an excellent starting point for more ideas about how best to represent information and control mechanisms within a local environment. It is a simple step to see that the interfaces to this room provide a wide selection of services and with the addition of more services inside the room; the iDorm could be a very useful experimental framework.

The application of a Grid framework in this room also means that how people react to different kinds of interfaces, indeed which interfaces they prefer to use in different situations can provide very useful results during usage experimentation.

The IIE group feels that with the increasing amount of technology that requires detailed interaction, the days of the basic switch and button are numbered. It is hoped that by experimenting with the iDorm and its interfaces, we can help move towards the new generation of interface design and thus play a part in the evolution of mankind's daily life.

6. Future Work

The IIE group is currently working on an intelligent agent that uses the services in the iDorm to monitor user's behaviour. By using this information, the agent is able to learn about specific user environmental preferences. Using this knowledge, the agent is then able to update the room's environment, pre-empting the need for the user to express their preference thus reducing the cognitive load.

Other work includes becoming part of the design team for an extension to the University of Essex's Computer Science department. The group's purpose is to incorporate similar services found in the iDorm into the new building. Bringing these services in at the design stage should make the added services physically invisible to the user. This brings the new building extension into line with the eGadgets research team, a subsection of the IIE group partnered with CTI in Patras, Greece and NMRC in Cork, Ireland who are funded by the EU's Disappearing Computer body.

The IIE group are also continually updating the iDorm to keep its services at the state of the art. The next sensor to be installed will be a computer sensor that can tell what the user is doing on the PC and even what kind of music they are listening to.

7. Acknowledgements

We would like to thank Graham Clarke and Vic Callaghan for their supervision and ideas from the design to the implementation stage of the iDorm.

We would also like to thank Robin Dowling and Malcolm Lear, without whose technical support the iDorm idea may have never been brought to fruition.

Finally, our acknowledgement goes out to the rest of the IIE group, whose ideas, support, late nights and early mornings have all combined to produce this experimental framework.

8. References

[1] Metcalfe R, ACM1: Beyond Cyberspace, San Jose, California, March 12-14 2001

[2] IIE Research Group – http://cswww.essex.ac.uk/intelligent-buildings/

[3] Echelon Corporation – http://www.echelon.com/

[4] The Integer House – http://www.integer.echelon.co.uk/

[5] Dallas Semiconductors – http://www.dalsemi.com/