Usability of Mobile Devices and intelligently adapting to a User's needs

Theme: Adapting to a User's needs

M-zones WP2

First Draft: 25-4-03

Stephen Greene and Jason Finnegan

Usability is defined in ISO 9241 (1998) as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction. Usability in relation to mobile services i.e. services that run on Mobile Phones and PDAs must look at the mobile user and surmise what interfaces for mobile services are appreciated and anticipated by the user. This paper will cover the area of usability issues when developing Mobile services. It will look at the mobile users and will propose a system that will store and utilise users' context information to help a user carry out a task. It will formulate a structure that allows applications to learn the significance and interrelations of people, places, objects and documents. By knowing the significance and relationships between objects and data, applications can always provide the user with the most relevant data and build up a context for user interaction.

1. Introduction

The primary focus of this paper is the issue of Adapting to a User's needs. The following are some questions that focused our research.

- Are current mobile service interfaces offering the full functionality to the user that they are capable of?
- Are the interfaces actually hindering the user's ability to complete their tasks?
- Have interface designers in general taken a user-centric design approach to their interfaces?

This paper will attempt to answer these questions and set out a suggested solution for some of the usability issues.

People come to new mobile devices having evolved a set expectations (mental models) based upon their experiences of using PCs and telephones. For example: emails are often saved in files whereas telephone conversations are not recorded; unlike spoken conversation, actions on a PC can usually be undone etc. These different system properties lead to assumptions about how technology will work. Designers need to uncover these assumptions and build systems that act in predictable (intuitive ways).

We believe it is possible to build a system that can adapt to a user's needs and that the system can also adapt to a user based on their contextual information in real-time. We are attempting to build an Intelligent User Interface (IUI) system to prove this. Maybury (2001) defines a Intelligent User Interface as:

> *"Intelligent user interfaces specifically aim to enhance the flexibility, usability, and power of human-computer interaction for all users. In doing so, they exploit knowledge of users, tasks, tools, and content, as well as devices for supporting interaction within differing contexts of use. "*

There are many definitions of what an Intelligent User Interface is and to what extent intelligence is used in to enhance the interface. Not all intelligent user interfaces have learning or problem solving capabilities. Many interfaces that we call intelligent focus on the communication channels between the user and machine. These interfaces often apply new interaction techniques such as speech processing, gaze tracking or facial recognition. The IUI that we will develop will use an intelligence system to generate the context information that the interface will use to automate some tasks.

Elhert, Patrick (2003) describes the driving focus of IUIs "*is that they are designed to improve communication between the user and machine"*. The end goal of our IUI system is user adaptivity which includes all techniques that allow the human-machine interaction to be adapted to different users and different usage situations. There will also be limited user modelling which covers techniques that allow a system to maintain or infer knowledge about a user based on the received input.

This paper is divided up into two sections; the first section details the usability problems we are trying to solve and the second section gives an in-depth look at our IUI system, the Ambient Intelligence Engine (AIE), which we have developed.

The first section will give an overview of the usability problems that a mobile device suffers from. It will give an example of some research already carried out in the usability area. It will look at the user interface in particular, both the screen size and the issues surrounding it and also the mobile device's user input. It will explain the importance of getting users usage patterns for interface design.

The second section will introduce our attempt to develop an IUI system that adapts to a user's needs. The problem of an IUI as described by Rakotonirainy (2002) is that "*it requires that tasks be aware of the surrounding environment, both physically and conceptually*". This information is the task's context and can fundamentally change the way a task is completed by a user or computer. Most definitions of Context are quite broad e.g.,

"…any information that can be used to charaterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves."
(Abowd, Dey, Orr, & Brotherlon 1997)

From this definition we can see that an Application needs to know about people places and objects relevant to the User and the task or Application itself. It cannot be expected that the application programmer could have any knowledge of these things when developing the Application, so the Application needs to be told about these things or learn them for itself. Our system allows for autonomous self-configuring applications, Interfaces that will change for different users, or situations. This in turn allows for a reduced functionality interface but one that adapts to the user's contextual environment. It is designed to help a user complete a task but not take over the task completely. This section will describe the reasoning for the approach we took. The sample applications – email client, scheduler program - we developed to test our system will be described in detail.

2. User Interface

2.1 Influence of people's previous experience

Sacher and Louden (2002) wonder whether the interaction paradigm for mobile services will be based on PC use, telephone use, a mix of both paradigms, or something completely new. Research that has been carried out using the rules of the PC or phone paradigms cannot be readily applied to the mobile context. There has been little directly applicable research. However, Sacher and Louden have started a research project looking at a mobile service for teens. They found that services may need to support varieties of communication style dependent upon the context and recipient (e.g. for the teens in this project there were different structures for friends, school friends and parents). Recent work to exploit the potential of PDAs has shown how new interaction paradigms can underpin creative applications development. For example, Regan el a. (2001) have proposed a new interaction paradigm for handheld computing. They used multiple interconnected handheld devices to form a virtual shared workspace for children's learning.

The design of successful mobile services will be based upon discovering the most effective paradigm for usage. Usage is likely to depend upon a number of factors that cannot be predicted without empirical research using real-world simulations of these mobile systems. The process of discovery cannot be achieved through market research alone (asking people what they think they might do) or through applying the findings of people working in different (more limited) paradigms.

2.2 Problems with Input and Output

To give an understanding of the interface usability problems that mobile devices are currently facing, there follows a short section on the issues of the two types of mobile devices this paper is looking at.

Mobile Phones
The screen size on mobile phones is quite small compared to the average PDA. It's only recently that high-quality, full colour screens have been offered to the public at an affordable price.

The input method into a phone has been exclusively through the use of the keypad on the phone. This method can offer up to 22 keys. Although for navigation around a service on the mobile phone 6 keys are used; a 'select' key, a 'back' key and 4 keys for up, down, left and right movement of a cursor. Some of the newer phones use a mini-joystick to move the cursor.

PDAs

The screen size on a PDA is considerably larger than on a phone. With this larger screen size, the PDA can offer a richer experience than a mobile phone. The majority of new PDAs offer high quality, full colour screens. The resolution on the PDA has increased over time meaning that more information can be fitted on screen.
The input into a PDA is more complicated than a mobile phone as the user has the ability to use a stylus on many of the PDAs. The stylus can simulate some of the functionality of a mouse. The ability to 'point and click' is combined into a single 'tap' which can be used on icons on the device. Also on many PDAs there are some map-able 'shortcut' buttons available to the user to quickly jump to a specific service.

Michael Dertouzos (2001) has argued that spoken dialog should be the main approach for exchanges between people and machines, and vision should be the main approach for human perception of info from machines This based on the assumption that speech is the natural mode of two-way communication and vision is used mostly one-way – for taking in information – and only secondarily for generating visual cues that reinforce spoken communication. These arguments could be taken to apply even greater force more with regard to handheld devices given the small screens and crude input devices. When speech recognition technology comes to maturity, will people use keypads, pen input etc. with handheld/ mobile devices?

2.3 The mobile context

In order to provide real life testing of interfaces we envisage using PDAs, wireless-enabled or not, but in a mobile context. User tests have traditionally been carried out in controlled conditions, often in quiet rooms. However, mobile users will be in the corridor, canteen etc, and this impoverished user-attention environment, with high user distraction and social context represents the real-world use of mobile devices. Research needs to answer such questions such as how long do people stop to interact with their device? Do they use it while, literally, moving? We believe that information on use in these contexts will prove just as important as power and speed in determining the overall design of the services

2.4 Fitting a mountain into a teacup? The Palm Pilot

Haitani. (2001) contrasted usage patterns of handheld and laptop devices: people generally use handheld devices in short bursts to quickly access data, and then put them away; in contrast, laptops tend to be powered up more infrequently, but for longer sessions (e.g. working on a spreadsheet). Since this usage pattern is different, Haitani argued that the UI design should be different.

He argues that the key to the whole user experience is to reduce the frustration of searching through menus by understanding user priorities and giving people what they want in terms of the visibility of frequently used items. He focused on optimising navigation to reduce the number of taps needed to access frequently used items. He argues that people are not too concerned about 'one more tap' for features used infrequently. An analogy would be your desk – if a frequently used item like a mouse or stapler was hidden away in a drawer this would likely induce frustration; in contrast, having an infrequently used item like a staple remover in a drawer (hidden) probably reduces clutter. In summarising this approach he says: " 'How do you fit a mountain into a teacup?' if that's question stumps you, you're still in the PC mindset"

2.5 Sonically-Enhanced User Input Buttons

Brewster proposed a possible solution to the problem of the user input on a mobile device taking over too much of the display screen. He believed that one way to overcome this problem was to introduce the extensive use of sound to help a user in the completion of their task. Brewster said "*The underlying hypothesis being that presenting information about the buttons in sound would increase their usability and allow their size to be reduced*". The idea is "*that presenting information about the buttons in sound would increase their usability and allow their size to be reduced*". Specific sound is used to indicate if a user has pressed a button, moved a pointer over the button or released a button press. He found that by reducing the size of the onscreen buttons while adding the use of sound, more data could be entered with the enhanced buttons and there wasn't significant drop in the quantitative performance. When he reduced the size of buttons even further, quantitative performance remained the same but it caused a significant increase in subjective workload. So while the user was able to enter the same amount of data, they had to spend more time focusing on the buttons. Brewster ran two formal experiments, one experiment was to determine the smallest size that people could work with and still remain productive. The second experiment was to determine the usability of the sound-enhanced interface in a real world setting. Brewster speculates that sound could also be used for the presentation of graphical information to a user e.g. a bar chart or a histogram graph. Brewster believes that the use of sound to enhance buttons is a must to increase the usability of the application.
"*If small buttons are to be used in the display of a mobile device (and there are many instances of targets on mobile device displays that are small) then sounds must be included to raise levels of usability as much as possible. If sounds are not used then performance will be significantly impacted.*"

2.6 Unanswered Questions

We believe that questions about Next Generation and other emergent technologies have to be answered with due regard to the human context within which people will use them. What are people's goals and needs and how will the technology fit with people's limited attention? The devices envisaged offer potential for augmenting a user's working memory - by using two modes of presentation (audio and visual). Will this be a significant factor in how such devices are used or will it be offset by the environmental conditions in which they are likely to be used? Also there can be significant levels of frustration associated with a lot of technology - will this disappear or, more likely, will it still exist and if so how much will users tolerate (in terms of download speeds, quality of images etc.) before they decide a particular solution is not for them?. These very human questions need to be addressed in order to ensure the success of the technology.

3. Intelligently adapting to user needs

The Ambient Intelligence Engine (AIE)

3.1 Why develop the system?

As described in the introduction, this proposed solution will attempt to ease the workload of the user when using an interface. It will attempt to take some of the monotonous actions of the user and automate them e.g. constantly moving new email messages into the relevant folders. This simple task for the system could potentially save a lot of time on the users' part.

The problem of recording context information by a computer system is well researched. In the Area of Context-Aware Systems much faith is put in the ability of sensors to tell computers about the world around them. However most sensors available today tell us little about what is happening in the real world, they are limited to reporting location, heat, light, pressure and sound levels. Even when all put together these sensors tell us nothing about what a person is doing and what they what to achieve. Microphones and video cameras perhaps with voice and gesture recognition may be able to do more, but they require a lot of processing power and further research. The advantage of using sensors is that they require zero user effort to interact with, so when used in conjunction with a more informed context system they can achieve the goals of invisible seamless computer interfaces, set out by Doucatel et. al (2000), in their Scenarios for ambient intelligence in 2010.

3.2 Relevance Theory

The main goals of an intelligent user interface support system would be, to learn through use, to make best effort guesses, to generalize and associate data and to put data into context. Google, the popular search engine shows many of these characteristics and a by looking at the PageRank formula that underpins this system we can see that the sum of the knowledge is greater than its parts.

The original Google formula , which has since been updated, by S. Brin and L. Page (1998)

$$\Pr(a) = 1 - d + d \sum_{i=1}^{n} \frac{\Pr(T_i)}{C(T_i)}$$

Pr = PageRank
D = dampener, Static value between 0 and 1, set by Google
C = number of links on a page

A WebPage's PageRank is calculated by looking at all relevant pages that link to a that page, their contribution is their PageRank dived by the number of links on their page. When used in conjunction with text matching techniques it consistently returns the most relevant page on virtually any topic. A weakness of this version of the formula is lack of any learning on the part of the system.

In our system we propose an extension of these ideas, with data Nodes replacing web pages, linked together by variable strength links. These links connect relevant or contextual data together. By using 'Topic map' like types as data nodes we can then store and retrieve data of a particular type in order of relevance to a existing data node. Biezunski et al (1999)

3.3 How it works

The Ambient Intelligence System that we have designed takes information from Applications as the user is doing their daily work, and stores it away in a centralised Prolog 'Network'. This Prolog network forms the Core of the Ambient Intelligence Engine which filters, parses and handles data between the Core and Applications.

The Core holds two distinct types of information, Nodes and Connections. The Nodes represent some real life person, place, thing or concept. New nodes are created in the network for example, when a user records a meeting in their PDA's diary (new Meeting Node), or sends an email to someone (new document Node).

Connections are what link the nodes together and are based on the fundamental concepts of Fuzzy logic and Neural Nets. Each connection links a single Node to another Node and has a strength value between zero and one hundred. These connections are created whenever two nodes interact in any way and are strengthened or weakened depending on how useful applications find the connections to be. There can only be one direct connection between a pair of nodes, but there can be any number of indirect connections, eg from node X to node Z and node Z to node Y.

Each node is of a specific type, it must represent a person, a thing or concept of some sort. There is a structure in each node, a number of fields holding relevant data about that 'thing'. New Node types can be defined by applications, allowing the AIE to be extended into areas by new applications while continually adding to the knowledge of the overall system.

The Core is written in Prolog with Nodes specified as follows

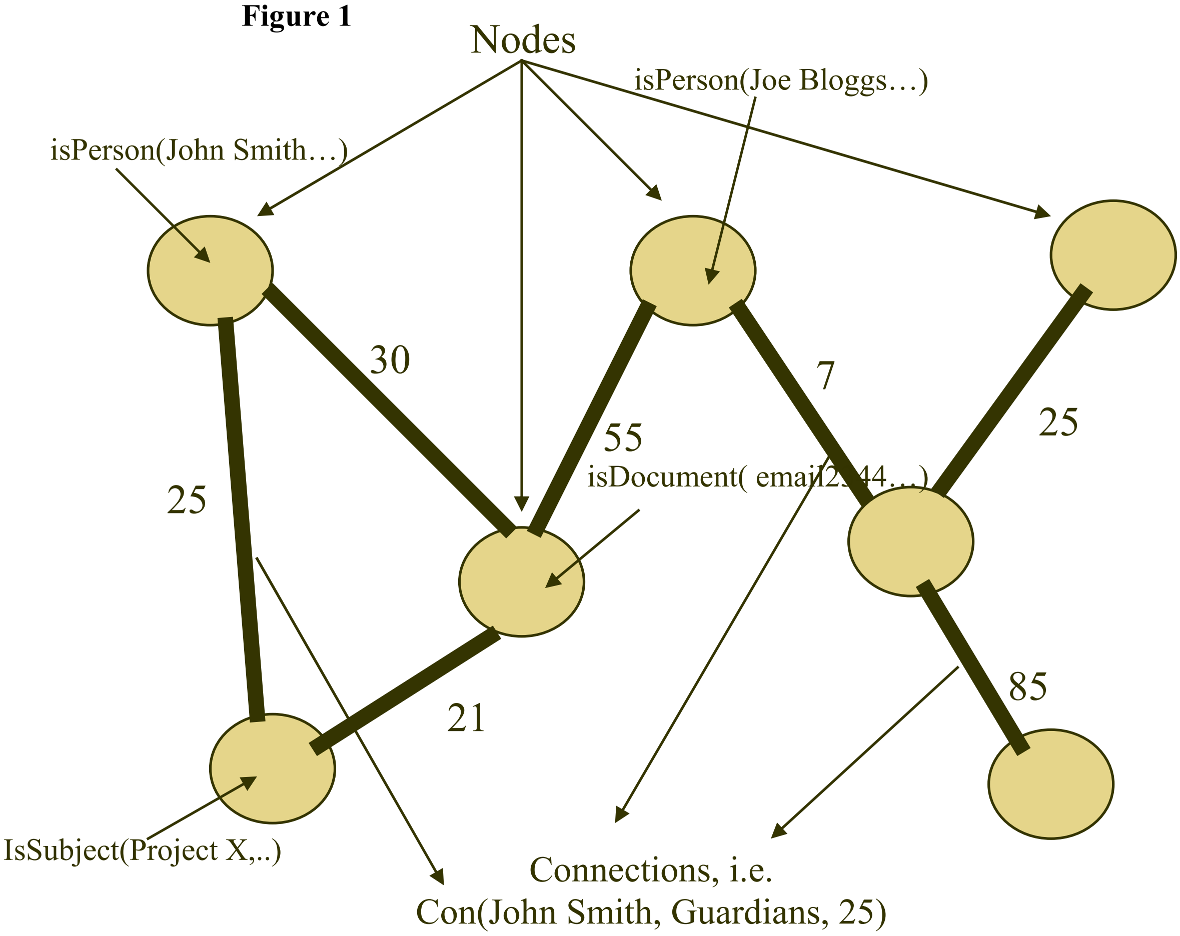
isPerson('Joe Bloggs','Joe','Bloggs','jBloggs@tssg.org',76843).
IsMeeting('Progress meeting','M-zones','A.N.Other','Joe Bloggs%' ,'1048600800000','1048604400000','lsmall meeting room',m1544).

Connections between Nodes are specified like so

con('Joe Bloggs',m1544,37).

Using this structure it is easy to list all Nodes of a certain type connected to a particular Node, and list them in order of total connection strength.

Figure 1



3.4 The Connections

Connections between Nodes are bi-directional associations, i.e. they have the same value in both directions, which differentiates this network from other concepts such as Bayesian Networks and Fuzzy groups (Fagan R., 1996) (Jensen, F. 1996). The connection both associates 'John Smith' to 'Project X', and 'Project X' to 'John Smith' by an equal 'strength' of 25. However all connection strength are relevant only when compared to other connections between similar types of Nodes. This means that the connection of 25 between John smith and Project X wouldn't be very important to him if he has several other connections to other projects with higher strengths. However as far as Project X is concerned, John Smith may be the most strongly associated person and therefore very important to it.

3.5 Linked and Multiple connections

In our model we propose 'relevance' have similar implication characteristics as probability, which states that given three facts A,B and C
if prob(B) given A is true =70%
if prob(C) given B is true = 40%
then prob (C) given A is true= 28%

'Relevance' in our model ignores direction so we break from the probability model by also stating that
Relevance(A) given C =28%

From this we develop the rule 'conl' for Linked Connection that goes through a third node, the value for which is equal to the product of the strengths of the two parts, divided by 100.
This means if

```
con(A,B,70).
con(B,C,40).
```

then

```
conl(A,C,28).
```

If there are several different 'paths' connecting two nodes then the total connection strength is calculated by taking all paths into account, and also how many different paths there are. The issue of how deep to look, how many maximum links in a connection to look for would have to take into account the processing overhead involved. This means that queries could be prioritised with more important queries able to look that one link further.

The ability of this Network to take multiple items of information, aggregate them and then answer queries in excess of what its been explicitly told is what makes this system useful.

3.6 Neural Nets and Fuzzy Logic

Fuzzy logic is a system created by L.A. Zadeh to allow computers handle situations where exact values of either right or wrong, 0 or 1 are not appropriate. Fuzzy logic allows variables to be any value between 0 and 1, allowing a computer to express an answer as either right or wrong and to map to ambiguous words such as "Tall" "Cold" etc. The AIE system takes into account some of the principals of fuzzy logic in its method of returning its best guess at the correct answer to a query. A query returns a fuzzy set of objects which have variable strength connections to he set. The set could be empty so and this must be handled by the Application making the query.

Neural Nets are made up of a network of interconnected nodes, with each connection having a variable 'weight'. The typical usage of a neural net is in solving a problem with many known input variables, and a number of possible outputs. The neural net is then trained to solve a particular problem involving these inputs/outputs by either

being given the correct answers (supervised training) or trying to group results together itself (unsupervised training). While neural nets show good results when trained to solve a particular problem, there is an inherent amount of inflexibility means they are "Limited to those kinds of things that can be passed on through imitation" J. L. Voss, (2000). The AIE system uses the principals of Neural nets in strengthening connections that produce correct results and weakening connections that produce wrong results or are ignored by the user repeatedly.
One of the main goals of the AIE system is take information learned from one domain and apply it to another with overlapping elements.

3.7 The Intelligent Applications

A central strength of the AIE System is that without imposing a strict structure on the information being stored, it is possible to share and refine information between many different Applications and Users. In our current implementation of the AIE system we have a Calendar/scheduler Application called OnTime and an Email client (with email sorter) called SmartMail.

We choose the email client because of the abundance of test data, and first hand experience of email overload. The email client has access to rich context information, people in general will sort their emails into folders based on certain keywords that they look for, for example moving all emails that contain the keywords java, swing and awt into a "java" folder. This pre-sorted content is a great source of context information if a capable application is used to extract this context information.

The email client looks at the structure of a user's account and determines the criteria that a user used to sort their emails into the individual folders, these folders are taken as subjects and the keywords for that subject are then saved into the Ambient Intelligent Engine (AIE). A connection is made between the subject and the user. At this stage the AIE has a list of users connected to subjects which have associated keywords. When the email client is used to sort a user's Inbox folder, it passes details (To address; From address; Date; Folder) about each email message to the AIE, the AIE queries its connections and returns back to the email client a list of keywords that the email client should search the email for. The email client will keep score of the number of times a keyword appears in the email message, giving a higher score to occurrences in the subject line, the reason being we believe a keyword in the subject line is a possible indication of the content of that email. Once the keyword searching is complete, the email client will pass the results back to the AIE. The AIE then determines which subject i.e. folder that the email should be moved into. The email client will then move the email message into the folder, creating it if it doesn't exist. An option is available to the user whereby all new incoming emails are automatically sorted by the email client using the AIE for context information.

The critical difference between SmartMail's sorting and standard email sorting is SmartMail uses aggregated information extracted from other users sorting behaviour and applies this to users' who share common subjects. This central pool of rich context information significantly reduces the amount of effort required to sort email.

The scheduler application was selected as it is a great source of time relevant information, such as where and when people plan to be. The scheduler is an effective way of simulating sensor information and location tracking systems that are currently in development.

Both applications function as you would expect from 'non-intelligent' applications, but every action the user makes is added to the total knowledge of the system. For example, if a user schedules a meeting about a particular project using OnTime, and puts down some other Peoples names as also attending the meeting, then all those involved get a connection to the project topic (or stronger connection if already exists). One effect of this change is that next time the user receives an email from any of those people; the chance of it being sorted into that projects folder is increased. The next time the user schedules a meeting about that project there is a greater chance those people will already be selected as attending.

The Applications could connect to a central AIE server that is used by a whole company or department, or each user could run a personal server. By using a multi-user server the AIE can share information between users, and also increase the speed at which the system learns by generating more data. If the AIE is run as a personal server and even if only used with a single application, it is still gives improved usability to most users. As users acquire more mobile device, synchronization between them is becoming an important issue, and the AIE can serve as a personal synchronization service.

3.8 Developing an Intelligent User Interface

The actual direct effect the AIE System has on the applications varies depending on the requirements of the application, it can be designed to be an invisible helping hand, cutting down on the number of button pushes required to get a simple task done on a mobile device. However simply by changing which option is selected by default, can greatly increase the usability of the device.

Every application that wishes to make use of the AIE System must take several principals into account.

- The AIE is only to provide Context and to assist the User; every application should be capable of achieving its tasks in cases where the AIE can provide no relevant data.
- The AIE stores two types of information: facts and connections. The application must be able to handle both types of information correctly
- User Feedback, both positive and negative is vital to getting the AIE to learn through use.

3.9 Sorting and Filtering

As described in 3.7 filtering and sorting of mail is a practical use for the AIE system. Because the AIE system knows every project and subject your involved in or interested in it can sort your mail into relevant folders. This works by checking which ‘Subject’ Nodes are related to the sender and receiver of an email, and searching the email for keywords related to that ‘Subject’. It would be quite straightforward to create a ‘Spam/Junk email’ subject that would learn from previous emails, what is and is not ‘Spam/Junk email’ and filter them accordingly.

Sorting and filtering of other types of information is also possible. For example, if a project meeting was about to take place and a user informed the AIE system that they were attending, the AIE would attempt to return all relevant information e.g. emails, documents, news, anything that is stored about the project. Another example would be if a user is in a weekly meeting and has requested that only important or relevant calls, emails or instant messages are allowed to disturb them, the AIE would determine which communications are allowed through based on the content of the communication and who sent it.

3.10 Issues with Ambient Intelligent Systems

There are a number of issues that must be addressed when designing a system that tries to help a User, but also stays out of their way. It’s a balancing act between doing too much automatically and not doing enough. When designing applications for the AIE System it was decided to create all applications with the capability to work independently of the AIE, to have a fallback or default setting for everything. Then as the AIE becomes more stable from watching but not doing, it starts to reorder lists, make its best guess of the default value etc.

3.11 Future Work

The further development of the AIE system, which will be extended to include information from other sensors and devices, to help build a picture of what a user is doing at any given time. User Status or presence could be linked to the system via an Instant Messenger Application, Location tracking and linking unspecified conceptual locations with real coordinates would allow the System to build a virtual map of a user’s surroundings, without the need for a time consuming survey.
The AIE System could also be integrated into a complete Smart Space Management System if required, by extending to cover awareness of devices and Services.

The deployment of the system will need to address several issues:

- Scalability, we need to consider the constant context information input of several hundred if not thousands of users into the AIE system. Can we provision resources for this large scale storage requirement?

- There will be a need for extensive testing, both of the AIE system and the sample applications that use it.
- An AIE interface API will be developed which will allow developers access to the AIE system. The developer will be able to make calls to the AIE system requesting information or give the developer the ability to pass information to the AIE system.
- The issues of privacy and security must be looked at as the AIE system is storing a large amount of context information about a user. How can we control access to authorised applications only?
- The environment in which the AIE system applications run, will play a large part in the deployment as it will affect the extent that the issues mentioned above will be researched e.g. if the environment is an academic one the issue of scalability is not as important as the issue of testing, as they will be using only a small number of users/applications but they will be getting those users/applications to do a large number of tasks, so they will require a robust, bug-free system.

There is still a great deal of work to be completed to get the AIE system up to the level of a robust, deployable system. But the potential for the AIE system is great with its ability to enhance the interfaces of many applications. Plus its ability to store and organise context information could in the future be used by any number of applications.

4. Conclusion

4.1 Usability Concerns

This paper has covered some of the usability issues that are faced by mobile devices. It has looked at the two main areas where mobile devices have usability issues; Visual interface and user input. There have been many different solutions proposed to solve or alleviate usability issues. The sonically-enhanced buttons solution is just one example of a system whereby an attempt has been made to overcome the lack of screen size on a mobile device. There are several commercial attempts made to overcome the screen size problem. The application ThunderHawk by Bitstream attempts to offer the same browsing experience as on a desktop computer. It does this using by using a server/client model. The webpage request is sent to the server which fetches it and renders the page into a special format for the client.

Screen size will never get much better than they are now; if they did then they would no longer be as mobile. So application designers for mobile devices will have to take the weaknesses of the mobile device into account when designing the interface. There are many forms of user interaction available to them to exploit from full colour screen to a speech recognition. If they can adopt a multimodal approach then this will allow for a rich user experience.

4.2 Ambient Intelligence Engine

This paper has also proposed a solution to the issues of adapting to a user's needs. The development of an Intelligent User Interface system is our main goal, that is a system that allows an interface adapt to a user's needs. The interface can use contextual information to help users complete their tasks. A benefit of the interface is that not only can it use context information but can also be used to record a user's contextual information. This allows for the constant learning of a user's usage patterns albeit at a lower level, we are recording what the user has done, not how they have done it. Then the system offers the best options to the user based on their recorded context information.

The AIE system can also do more than helping in the development of an Intelligent User Interface. It can be used to provide and store contextual information, which could allow the AIE system be used in conjunction with a Speech Recognition system for example. The Speech recognition could pass the subject that the user is currently working with e.g. project name, to the AIE system which would determine the context the user is working in and send back list of keywords, people names, Locations and other contextual Data that Speech recognizers normally have problems with.

References:

ISO 9241 (1998) - Ergonomics requirements for office work with visual display terminals (part 11 - guidance on Usability) -

Maybury, M.T. (2001) .Intelligent user interfaces for all.., in User interfaces for all: concepts, methods, and tools., Stephanidis, C. (editor), Lawrence Erlbaum Associates, Inc., Publishers, Maway, NJ, USA.

Ehlert, Patrick. (2003) Intelligent User Interfaces Mediamatics / Data and Knowledge Systems group Department of Information Technology and Systems Delft University of Technology, The Netherlands

Bergman, E. and Haitani, R (2000) Designing the PalmPilot: A Conversation with Rob Haitani, Chapter 4 in Information Appliances and Beyond, Eric Bergman (Ed.), Morgan Kaufmann

Dertouzos, M. (2001). The unfinished revolution: Human-centered computers and what they can do for us. Place: NY: HarperCollins

Regan L. Mandryk, Kori M. Inkpen, Mark Bilezikjian, Scott R. Klemmer, and James A. Landay. "Supporting Children's Collaboration Across Handheld Computers." In Extended Abstracts of CHI 2001, Conference on Human Factors in Computing Systems. Seattle, WA, April 2001. pp. 255-256

Sacher, H.and Loudon, G. (2002) Uncovering the New Wireless Interaction Paradigm. Interactions ACM January and February 2002.

K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten and J.C. Burgelma, Scenarios for ambient intelligence in 2010 (ISTAG 2001 Final Report), IPTS, Seville, 2000.

Michel Biezunski, Martin Bryan, and Steve Newcomb, editors. ISO/IEC 13250, *Topic Maps*, URL:http://www.ornl.gov/sgml/sc34/document/0058.htm.

S. Brin and L. Page, "The anatomy of a large scale hypertextual web search engine," in *Proc. 8th Int. WWW Conf.*, Brisbane, Australia, Apr. 1998, pp. 107–117.

Grimm et al., Programming for Pervasive Computing Environments, University of Washington Technical Report UW-CSE-01-06-01, June 2001.

Jensen, F. 1996   An Introduction to Bayesian Networks, Springer Verlag,

Fagan R., 1996, Combing Fuzzy information from multiple Systems, IBM Almaden Research Center

Stephen Brewster, Overcoming the Lack of Screen Space on Mobile Computers, Glasgow Interactive Systems Group, Department of Computing Science, University of Glasgow.

John Greaney, P2 Usability issues and Mobile devices NOMAD, Dun Laoghaire Institute Art Design Technology, Dun Laoghaire, 2003.