

Achieving Non-Intrusive Environments for Local Collaboration

Martin Jonsson, Calle Jansson, Peter Lönnqvist, Patrik Werle
and Fredrik Kilander

The FUSE Research Group, Center for Wireless Systems, KTH
DSV, Electrum 230, SE-164 40 Kista, Sweden
{martinj, calle, peterl, werle, fk}@dsv.su.se

Abstract. The way people use computing artefacts in local collaborative settings is problematic. The main reason for this is that the tools that are being used to a large extent are designed as personal tools. Constant intrusions on the collaborative task are imposed from lacking support for collaboration, external non-related events, parallel non-related work and the breaking of different kinds of social norms. In an attempt to deal with one of these issues, the external events, a test bed environment for non-intrusive notifications was created. This environment consists of both hardware and software components and makes it possible to deliver short messages in different modalities, on public or private displays, and with different renderings.

1 Introduction

A lot of people, in a typical working day, spend big parts of the day together with other people in different kinds of meetings, or similar local collaborative situations. For quite a long time these settings has been fairly free of advanced technological tools. The tools used have been typical office tools, such as paper, pencils and a chalk- or whiteboard. Not even the PC revolution has had any major effects on the meeting situations, at least not until recently. In the latest few years though, we have seen a great increase in the density of technological artefacts present in collaborative settings. The major reason for this is the rapidly developing mobile technology, which makes it possible for people to bring advanced tools for communication and computation wherever they go, also in meetings. There are laptops, which can be connected to the Internet via a wireless LAN, PDA's, mobile phones and all sorts of new types of information appliances that are constantly being developed.

The nature of these artefacts present at the meetings is that they are primarily *personal tools*, containing different kinds of services with the primary goal to allow people to continue a work process while moving between different situations and places. So being in a meeting, a person can at the same time be connected to various distributed contexts, carrying out work that is not related to the meeting situation. Having these tools we might even be *expected* to be connected to our distributed contexts, performing work or being available for communication.

Another aspect of these personal artefacts is that they are also being used as tools for the local collaborative task. Somebody might be taking notes using a word processor, or

exchanging documents via the email client, or hook his or her laptop to a projector, providing a shared focus of attention. Most meeting rooms in themselves have very little stationary computing artefacts (eventually a PC hooked to a projector), making the personal artefacts the primary tools supporting the collaborative work.

All of the above results in a situation where the collaborative task becomes very sensitive to intrusions, both from the open channels to other contexts (mobile phones, email, IRC etc.) but also because of lacking support for the collaborative process.

A common example of intrusions is the use of mobile phones at meetings. A ringing phone creates quite a large intrusion in the local setting, and actually answering the phone might be even more intrusive. A common solution to this specific problem is to forbid the use of phones at meetings, shutting down the connection to external contexts. People wanting to participate in other context often create this kind of intrusions, and the intuitive feeling is that this is a behaviour that should not be encouraged. We believe however that this behaviour don't have to be a problem per se, and our aim is to try and see how we can improve the possibilities to work with parallel activities without interfering to much with the main task of the group.

We mean that the intrusive behaviour to a large extent depends on the design of the hardware and software. The artefacts that are currently being used are mainly designed as personal tools, to be used when alone and not involved in any local social or collaborative activities. Both the software and the hardware is therefore very focus demanding, effectively stealing attention from the surroundings.

The need for computers to become less focus demanding was earlier discussed by Mark Weiser in the vision of Ubiquitous Computing [7], which describes a world where computer resides everywhere around us, embedded in the environment, enhancing our lives in different ways. An important observation in this vision is that for environments with a high density of computer artefacts the computers cannot be designed like the PCs of today. Weiser mean that we must make better use of our peripheral processing abilities, and envision something called "calm computing" where technology stays invisible in the background until we need it [17].

Another aspect is that of ambient media [18]. With ambient media the goal has been to integrate information with physical space, often in the form of more or less "tangible" presentations. The Play studio in the Interactive Institute has a similar approach using the notion of "Slow Technology"[16], where for example *informative art* is used to convey information in ways that promotes moments of concentration and reflection.

In this paper we present an analysis of intrusiveness in collaborative settings, as well as some design principles to support the development of services appropriate for local collaborative work. We also present an implementation of an environment handling a specific type of intrusions, caused by notifications to persons in a collaborative setting. This environment for non-intrusive messaging consists of both hardware and software components and makes it possible to deliver short messages in different modalities, on public or private displays, and with different renderings.

1.1 What do we mean with intrusiveness?

A similar expression to intrusiveness is the notion of interruptions. This phenomenon has been examined in HCI literature by for example McFarlane [1] who defines an interruption as:

"Methods by which a person shifts his focus of consciousness from one processing stream to another"

Gille and Broadbent [2] have empirically examined when certain interruptions become *disruptive*.

While an *interruption* implies that a process stops or halts entirely, the notion of *intrusion* also could cover other negative effects on an ongoing process. McFarlane [1] defines intrusions as:

"Errors where people incorrectly perform actions from a pre-interruption task after task switching."

We see this definition as far too narrow for our needs. We would like to consider intrusions not only as actions by subjects, but as any kind of interfering events. We also don't see the previous process as the only source of intrusion, but could imagine that any parallel process could cause intrusions. Therefore we land on the following, rather broad definition of intrusiveness:

"The degree of interference with the realization of the main task of a group, caused by a number of intrusions"

Where intrusion is defined as:

"An occurrence of a process or task that is not intimately related to the current task of a group and that interferes with the realization of that task"

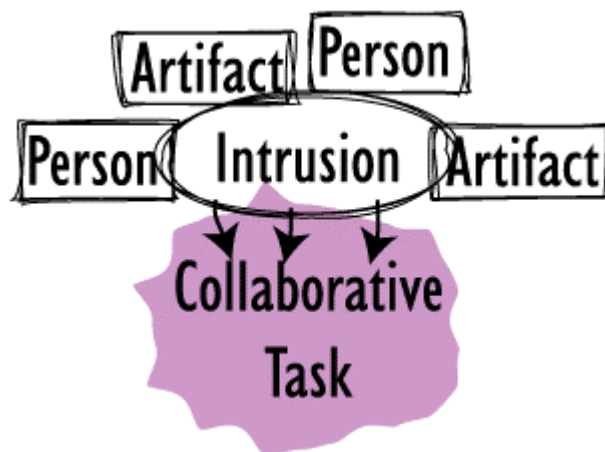


Figure 1: The intrusions originate in the behaviour of the local persons and artefacts and affects the collaborative task.

1.2 Sources of intrusion

The activities we are interested to examine are collaborative activities in small groups in computer rich environments. Such a computer rich environment with today's technology could look something like this:

- Personal devices: Laptops, handheld computers, mobile phones, etc.
- Public devices: Some computer connected to a projector.
- Personal software: email clients, web browsers, calendar, word processor, etc.
- Groupware: project management software, shared calendar, web browser or word processor on shared display

We have chosen to model the collaborative task from an Activity Theory perspective as described by for example Nardi et al in [3]. In activity theory, an activity is primarily described as a *subject* using a mediating *artefact* to transform an *object*. The subject is typically a person whereas the artefact could be a concrete tool such as a word processor, or more intangible entities such as methods or procedures. The object is the entity being transformed, and could be anything from a document to a person. In Activity Theory a collaborative activity can be described using Engeströms model for social activities [4]. In this model the subject is related to a community. The relation is mediated by rules, concerning the social structures in the community. The community has a relation to the object being transformed, basically mediated by a division of labour.

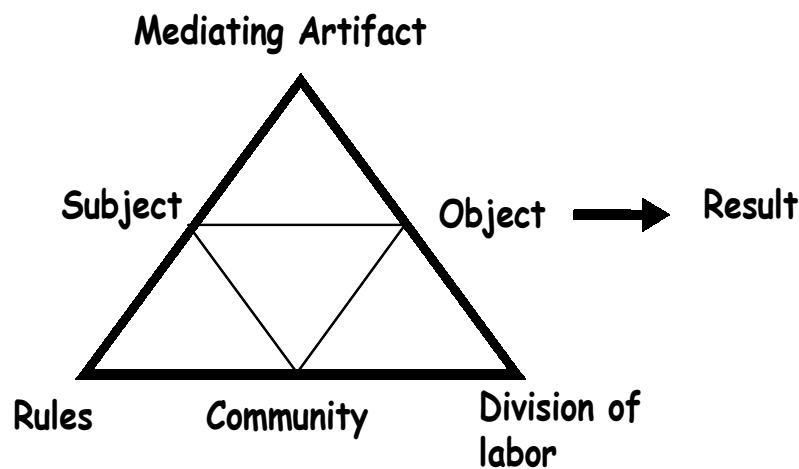


Figure 2: Engeströms extended model for social activities.

An intrusion as defined above would in this model be a process that interferes with the transformation of the object. It could for example be some malfunction of an artefact, something disturbing the subject, or frictions in the interpersonal communications. We have identified four major sources of intrusion for this kind of settings:

Insufficient collaboration support

The nature of the computer equipment used today in local collaborative settings is that they are giving much more support to individual activities than to collaborative ones. A tool that is not tailored for its actual usage is likely to create breakdowns in the work process. These breakdowns we see as one of the major sources of intrusion on the local collaborative work.

External events and notifications

Common for this kind of intrusions is that they originate in a context different than the local collaborative one. External persons might try to establish a connection with someone in the room, using telephone, chat, email, SMS, through the door, etc. Another type of such events is caused by active software components, such as email and calendar clients that will try and notify its users.

These events will create two types of intrusions. The connection attempt (noise, visual cues) will distract others than the recipient. The recipient might also have to deal with the situation in some way, thereby losing focus on the shared tasks.

In addition to distracting other people in the meeting notification might also annoy and irritate them. These feelings might affect the relation between the subject and the community in the model above, eventually causing longer-term effects on the collaboration between the participants.

Parallel individual activities

There is an increasing usage of personal mobile technology, including mobile phones, handheld computers and laptops. These devices provide the possibility to work with personal tasks, anywhere, anytime. It is becoming more and more common that this kind of technology appears in meetings or other collaborative settings. Sometimes used as a tool for the shared activity, and sometimes only used for personal tasks.

Working with personal tasks on personal devices however effectively steals focus from the shared tasks. The degree of attention needed to work with such an individual activity depends on two factors, the nature of the activity and the design of the mediating artefact. Some activities, such as writing, require a lot of our attention no matter what tool we are using, other activities such as browsing or searching for information could potentially be a rather peripheral activity if the design of the tool is appropriate.

Breaking social rules

By putting your mobile phone on the table you indicate that you allow your personal context to interfere with the collaborative task. There might however be an implicit social rule stating that the phones should be shut off during meetings. Being unaware of and breaking such rules affects the relations between the people in the group and might cause longer-term intrusions on the collaborative task.

2 Designing for non-intrusiveness

2.1 General design principles

Previously we identified different factors that we believe contribute a lot to the intrusiveness phenomena occurring in computer dense work environments. Here we will identify some general design principles that can prevent or diminish many of these effects.

Creating good models and explicit representations of work situations

At the core of mechanisms to diminish intrusiveness phenomena lie good co-ordination strategies. These in turn must rely on adequate models of the tasks being performed in parallel in each work environment (both in large and small scale), the characteristics of the participants, the physical properties of the work environment and the events that occur. We need also explicit and operable representations of these models, such that observations of a variety of contextual factors can be made and mapped onto those representations.

One particular problem with the handling of parallel tasks is the cognitive load when switching between tasks. If the services supporting different tasks can minimise the effort in switching between tasks, the effect of intrusions will decrease. The implementation of such functionality in services in turn has to be based on better modelling of tasks and work situations. This also holds for the personalization of work environments, facilitating the configuration or restoration of an appropriate work environment suitable for a particular group.

Modelling Intrusiveness and Non-intrusiveness

In the introduction we tried to give a rough model of what we mean by intrusiveness. This model has to be refined and also mapped into operational representations that can be utilised in e.g. a multi agent software system. Initially we are going to pursue this work along three tracks:

- **Modelling the social rules valid for a certain situation**, making them more explicit so that they can be shared both among internal and external human participants and supporting software components
- **Modelling the normal behavioural and perception patterns within a group**
- **Creating rough operational initial models** that can control the behaviour of our prototypes. An example of this is the introduction of the notion of *message impact*. The message impact corresponds to how likely it is, given the presentation properties, that the message comes through to its recipient, and to what extent the meaning of the message is conveyed properly.

Better support for local collaboration

To remedy the existing imbalance between technology support for shared local tasks and private and often distributed tasks the hardware and software support for local collaboration must be improved. In particular the following characteristics of collaborative environment have to be improved:

- **Shared focus** by co-use of public devices like interactive walls or tables
- **Simultaneous interaction** by co-use of input devices such as mouse and keyboard.
- **Transparent interaction** in the sense of uniform mechanisms to interact with information elements across physical and virtual boundaries
- **Ad-hoc device integration**. The different entities in the room should not be considered as separate artefacts but rather as components in a coherent dynamically configured system. This should also include the personal artefacts being used in the room.
- **Personalization of the work space** both during and between work sessions

A lot of work has already been done in order to try and create environments like this, for example in the Interactive Workspace project in Stanford [5] or in the iLand project at GMD, Germany [6], but none of them has explicitly targeted the issue of intrusiveness.

Specific design considerations

Here a number of design directions are summarised, that will be elaborated further in the next section:

- Viewing all public and private devices available in a work environment as potentially shared and useful for all parallel tasks by all participants
- Utilising the periphery of the work environment as much as the current focus point, making devices less focus demanding and allowing for easy transitions between periphery and focus.
- Utilising mixed implicit and explicit representations of information
- Providing a rich set of modalities both for input and output and utilising mixed modalities for efficient interactions
- Providing a set of abstraction, sorting and filtering techniques
- Providing rich sensory observations as a basis for context dependency mechanisms
- Unifying the access to personal and collaborative support services.
- Allowing for easy ad-hoc configuration of all available hardware and software components
- Implementing efficient co-ordination strategies to allow for temporal, physical personalised synchronisation. This also includes negotiation schemes to create co-ordinations optimal for the whole group.

2.2 Specific design aspects

We have identified a number of design aspects or dimensions of relevance for handling intrusiveness problems when designing interactive systems for situations where many parallel tasks are interleaved.

Personal – Public

One interesting dimension to explore is whether interactions should occur on personal or public devices. The normal case has been to use personal devices for private tasks and public devices for common tasks in a particular work environment. We envision that future work environments to a larger extent will contain public devices that are free to use for a diversity of private and public tasks more or less temporarily. Such devices include computer screens of various sizes, speakers and various input devices. This development presupposes flexible and ad-hoc typically wireless mechanisms for associating devices in a work environment. In general a free use of resources in a work environment will improve the situation of work. More specifically from a non-intrusiveness aspect, a well designed and mixed use of public devices for private tasks and use of private devices for common tasks may break today's sharp boundaries between these kinds of tasks. Each particular design must of course also handle the privacy issues raised.

In focus – Peripheral

The next dimension is the systematic mixed use of focused and peripheral interactions. In the vision of Ubiquitous Computing [7] a big issue is the notion of calm computing. One of the main ideas behind calm computing is to make better use of the ability of the brain to process information that is not currently in the focus of your conscious mind, but rather in the periphery. An obvious example is how we process the text on street signs, which we ignore most of the time but which appears out from the background when we need it. Similar ideas are at the core of concepts like ambient media, where information is added to the surrounding environment in a very subtle way. One way to utilise this is to invoke interactions on displays in the room that are not currently in the focus of attention. This can be compared to the email symbol appearing on the bottom of the PC screen, as the bottom of the screen can be regarded as being in your periphery when your main focus is on another section of the screen.

Degree of explicitness

The interactive use of abstract symbols and sounds as bearers of information might steal less focus from an ongoing work process compared to explicit texts either read or listened to. The use of implicit messages is also related to the concept of ambient media discussed above. There are however some disadvantages with using implicit messages. One disadvantage is the amount of information that can be conveyed is rather limited. By mapping symbols and sounds to properties of a message you can convey information about e.g. the message type, its importance, receiver etc., but it might be very hard to convey any longer or richer messages in an understandable way. Another obvious

problem is the issue of interpretation. If the symbols should be interpreted by its receivers they must either use intuitive metaphors easily understandable by anyone, or else the meaning of the symbols must be learned and known before using them. As we consider multi-user environments the mapping of meanings to symbols must be personalised.

Choice of modality

While the last dimension had to do with how implicit the representation is, this one has to do with the choice of modality. We assume here that an interactive work environment is provided with input and output devices that facilitate flexible choices of modalities for the interaction. Dependent upon the situation sound may be a better way of conveying information non-intrusively than vision. If the current task mostly involves visual processing, an audio-based queue is less likely to interfere with the current task than a visual cue [8]. Apart from the fact that the environment should provide as large a spectrum of modalities as possible a careful design with many parallel input and output options for the user is requested. In our first prototypes we will only utilise visual and audio interfaces, but further on we will also include haptic interfaces, e.g. vibrating output devices and gesture recognition input devices.

Information filtering

Maybe the most obvious way of decreasing intrusiveness is by filtering the parallel interactive streams. There is a lot of work done in particular with respect to email filters where messages are analysed and categorised with respect to sender, message subject, keywords in the message body, and we will use existing filters or just assume the existence of a filter. In our first prototypes we assume the existence of a filter for private interactions where each person can affect how severe his or hers messages will be filtered. The strength of filtering could be varied by e.g. changing the number of filter rules that are applied. Each such personal filtering preference will also affect the filters of the other participants. A common strength of filtering will be negotiated using software agents with some additional knowledge about e.g. personal long term preferences.

Co-ordination and frequency modulation

One phenomenon that may affect the perceived intrusiveness is the co-ordination of the interactions both with respect to time and the available artefacts in the room. Unique to this aspect is a temporal point of view, where interactions may be clustered more or less densely or spread out with a maximal frequency of occurrence. Interactions may also be sorted both with respect to task and person. Lastly interactions may be routed through devices with specific modalities. Many of the sub-mechanisms needed have been touched upon above in relation to the topics of filtering and choice of modalities, but the specification of the overall co-ordination mechanisms is an important design task also dependent on the quality of mechanism for context-awareness (see below).

Degree of context-awareness

The possibilities for adequate co-ordination schemes are dependent on mechanisms for context-awareness, where context includes both the physical status of the work environment, the participants, the status of the tasks etc. Our intention of modelling the work process going on in the room has been discussed above, so here we focus on the physical aspects of the room. The first element here is the existence of different kinds of sensors. One example is microphones to measure the level and distribution of noise in the room. Other natural elements are measurements of locations and movements within the room. Of course all these observations only make sense in the context of an at least minimal model of the work process in the room in terms of which they can be interpreted. Examples of situations where physical context-awareness can influence the degree of intrusiveness is the detection of suitable points in time to deliver messages (e.g. a suspected break) and suitable devices (e.g. adjacent to a person) to use for the particular interactions. Previous attempts to create context sensitive notification, has often focused on location as the most important context factor, such as in [9].

3 A notification environment implementation

We have chosen to use the design issues from above to target a certain type of intrusions. These intrusions are disturbances caused by the process of delivering notifications or short messages to members of a group currently involved in a meeting. With notifications we mean short messages of several different types, such as calendar event notifications, short or truncated email, SMS messages, etc.

A typical such intrusion might be the delivery of an email to one of the persons in the meeting. This event causes several interferences with the meeting. One such interference is the loss of attention from the receiver of the message. His or hers attention is required firstly to manipulate some artefact to open an email program, and then to read the actual message. Another interference is the effect on the other participants in the meeting. They might first be disturbed by some notification sound or visual cue, and later by the fact that the receiver's attention is not focussed towards the meeting. The complete intrusion should consider the effects on both the receiver of the message as on the other people in the room.

A prototype environment was developed where hard and software was carefully designed in order to try and minimize the intrusions from incoming notifications. In the scenario for the prototype a small number of persons are having a work meeting, working on a document on a shared screen. While they work they are constantly given notifications of different kinds. These notifications appear both on personal devices and on public devices in the room. The participants can collectively affect with what impact the messages are being presented in the room by turning a knob residing in the middle of the table.

3.1 Physical setting

The setting is a meeting room with a table, some chairs and a wall display. Embedded in the table are small screens, one for each place around the table. On the middle of the

table there is a “knob”, looking like a typical volume control. To personalize the embedded screens, a user pushes a personal identifier device into a receptor connected to the screens. A projector is also used to project information against one of the walls. A device that resembles an analogue clock is hanging on another wall. A pair of speakers is also present in the room.



Figure 3: The notification environment, containing embedded table displays, shared focus display, and iClock display, displayed at the Disappearing Computer Community jamboree in Zurich 2001.

3.2 Software Services

The software services used in the prototype are implemented entirely in Java. The level of sophistication and independence of physical hardware available in the current versions of the language makes this approach both feasible and convenient.

Apart from the vendor-specific libraries provided for the TINI-board, the remaining software is built on the standard APIs and classes published by Sun Microsystems. This includes Jini [10], the server and client dynamic brokering protocol, which we have used extensively to create a dynamic service environment with a minimum of manual configuration and Java RMI for inter-object communication. For the terminals, the graphics and sound APIs so far provide both necessary and sufficient support for our ambitions. The different services in the prototype are:

Notification message originators – where messages for human users are created. The originators can be self-contained programs or parts of a larger system.

Message terminal services – where notification messages are displayed:

- The iClock service is running on a computer that hangs on a wall. The iClock shows a clock most of the time but can also fade in text messages every now and then
- The sBanner is a service that uses a part of the projector display area to display messages on a banner.
- The iBanner is a service for the embedded table displays that shows text messages and optionally rings to indicate that there is a message displayed.
- The Wisp is an audio service where the speakers constantly are playing a “soundscape” and small shifts in the sound are used to convey messages.

Impact controller service – Participants use a knob in the middle of the table to influence the impact with which the notification messages should be displayed.

Message router service – a context sensitive message routing service, which decides where a message should be presented.

3.3 *Message originators*

The *message originators* provide the input of messages to the system. The originators can be self-contained programs, such as a calendar application or parts of a larger system, such as an email client. Typically they would build upon some existing messaging application/technology, where they would pick out the messages, transform them into the right format and then send them to the message router.

Since they require a message router to pass each message to the proper destination, the message originators are clients of the message router service. Through Jini, the originators obtain a reference to the router service and submit their messages for routing. There is a problem of choosing the right message router if several is available on the network. In the fuseONE system [11] a solution of this problem were presented, where location information is used to choose Jini services in the same room as the user.

3.4 *Impact controller service*

The “knob”, situated in the middle of the table is really a potentiometer connected to an Ethernet enabled embedded java computer (a TINI-board from Dallas Semiconductors [12]). The knob’s position is intended to represent the participant’s common agreement on how willing they are to be disturbed. The idea is that the participants should be offered a possibility to control with which impact the messages should be displayed in the room. This could depend on e.g. the type of meeting, the state of the meeting, who the participants are, etc. The level of impact that the knob generates is called the *I-value*.

The router service subscribes to events from the knob publisher service. These events inform the router of the current setting of the knob and thus the desired level of impact.



Figure 4: The impact controller “knob”, situated in the middle of the table.

3.5 *Message terminal services*

The *message terminal services* appear to the message router as a set of services all implementing the same service interface. All information about the terminal, and hence

any routing decision, is obtained through this interface. Since the terminals appear as Jini services, terminals can be started or stopped at will. The router is automatically informed of the changes and can adapt appropriately.

The router and terminal services have also been written with an interoperability policy that can be paraphrased as 'mutual suspicion'. This means that each part supports the protocols of interaction but does not depend on the other part to honour it correctly. We believe that this approach, although slightly demanding, will be rewarding in the long perspective as it increases the robustness of the system.

When a message is delivered to a terminal it may take several seconds to actually render. It is therefore important for the router to learn when a terminal is ready to receive the next message. The protocol between a terminal and the message router supports two (possibly concurrent) modes of interaction. The router can poll each terminal to see if it is ready to receive a message. In addition, the router can also register a remote callback with the terminal. The callback gives the terminal a way to inform the router that it is ready to receive a message, busy or in the process of shutting down.

iClock

The iClock service is displayed on a computer screen that hangs on one of the walls. It shows a clock most of the time but can also fade in text messages every now and then. With the iClock service we want to examine a service that is public, peripheral, visual, and that displays the message explicitly. Our hypothesis is that a glance at the iClock is just as little intrusive as a glance at an ordinary wall clock (often present in meeting rooms). The iClock has in this prototype been assigned very low I-values, resulting in that the iClock is one of the few terminals that display messages when the I-value is set to very low. If the I-value rises above a certain level, the clock starts to chime discretely each time a new message is being displayed. The service thereby has several “modes” with different corresponding I-values. All the different terminal services have different modes in a similar way.



Figure 5: The iClock, first showing the time and then fading in a message for one of the participants in the room.

sBanner

The sBanner is a service that uses the upper part of the projector display area for displaying messages. The messages body scrolls over the screen while the receiver info blinks slowly. With the sBanner service we want to examine a service that is public, in focus, visual, and that displays the message explicitly. Our hypothesis is that a message being displayed on the sBanner is very intrusive, since the projector display area often is where the participants have their shared focus. The sBanner service therefore has been

assigned higher I-values. The sBanner service can change its behaviour with rising I-value, by increasing scroll speed and the frequency of the blinking of the text.

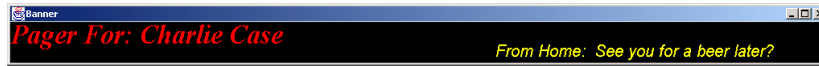


Figure 6: The banner service displaying a message.

iBanner

On the screens embedded in the table there is a service called iBanner running. The screens are actually Compaq iPAQ computers equipped with Wireless LAN cards. The iPAQs (hence the iBanners) are ‘personaliseable’ by pressing a personal identifier into a receptor connected to the iPAQ. The person identifier devices are iButton memory capsules from Dallas Semiconductors [13].

The iBanner service is similar to sBanner; a service that on the upper part of the screen showing scrolling text, with the difference that the iBanner service will ring at very high I-values. With the personal banner service we want to examine a service that is private, visual, and that displays the message explicitly. Our hypothesis regarding the iBanner service is that it is quite intrusive when a participant read messages on her personal device, thus it has been assigned high I-values.

Wisp

The WISP service, also described in [14] is the source of the sounds you hear from the speakers. The speakers are constantly playing a “soundscape” where small shifts in the sound will be used to convey messages. Each recipient is associated with a specific set of sounds and the idea is to point out to the recipient that there is a message for her. With the WISP service, we want to examine a service that is public, peripheral, audio based, and that has a more ambiguous representation of the message than if it was spelled out in plain text. In this prototype, the Wisp service works as the last way out; if there are no other services suitable for presenting the message, the message is being routed to the Wisp service. The Wisp covers the whole range of possible I-values. The speaker volume is adjusted accordingly to the current value. Some sounds are also exchanged to better fit the actual I-value.

3.6 Message router service

The Message router service is a continuation of the work with Active Documents presented in [11] The idea of Active Documents is to take off from the agent-programming paradigm [15], and turn documents into autonomous mobile agents and by that give them some useful qualities. A document should, for example, be aware of its content and the intention with it, and be aware of the context it is operating in, e.g. its receivers (who, why, preferences about formats, physical surroundings, etc.). The documents are active in the sense that they are autonomous (act independently), reactive (react on changes in the environment), and proactive (have their own goals and plans).

The message router receives all incoming messages and for each message makes a decision regarding which terminal service the message should be displayed on. The router is both a service and a client. It appears as a service for message originators and is a client of message terminal services. During its operation, the message router service uses Jini to subscribe to notifications of terminals as they come on-line or exit. This functionality, and the abstraction offered by the Java remote interface mechanism, makes it possible for the router to operate in a continuously updated view of the resources available to it.

The message terminal services appear to the message router as a set of services all implementing the same service interface. All information about the terminal, and hence any routing decision, is obtained through this interface. Since the terminals appear as Jini services, terminals can be started or stopped at will. The router is automatically informed of the changes and can adapt appropriately.

When a message is delivered to a terminal it may take several seconds to actually render. It is therefore important for the router to learn when a terminal is ready to receive the next message. The decision regarding which terminal that should display a specific message depends on several conditions. For example, the routing of the messages depends on the position of the 'knob' (situated in the middle of the table), the message will only be sent to a terminal that supports an impact that does not exceed the wanted value. There could also be meta-data associated with the message that could influence the routing decision, such as a private flag (the message will only be routed to a personalised device and not to a public). A terminal must also support the message's content type etc.

The message router persistently looking for new possible terminal services suitable for displaying the queue of incoming notification messages. It also get informed about changes in the context, such as that a person has entered the room or that the knob's position has been changed, and react accordingly to this information.

4 Conclusions

The system described has been taken to an initial user testing and extensive demonstration. The research team's own initial use of the system and the possibilities of demonstrating it to a wide array of users has given us some interesting input on how to further design a non-intrusive collaborative environment, both on the physical and the logical level.

Routing data to public and/or peripheral displays has been received with great interest and is mostly seen as a useful and interesting way of handling information. Especially the concept of the iClock, which to a high degree is unobtrusive and also can be perceived as an 'object d'art' and not only an information display and therefore be more easily accepted in a room where several participants meet. This differs from the public banner in that way. The public banner is only an information display and therefore more commented upon as 'ugly', 'intrusive' and 'irritating'. The design with the scrolling text is also found at such places as subway and train stations, advertisement scrolling screens

and the like and might thus have connotations with activities related to such (very public) environments rather than a workplace.

The idea of routing one person's information to a public display also raises very high demands on information filtering so that a specific user's possibly very personal information does not end up on a viewing screen where everybody can share it. Some commentators have suggested this, and that they never would allow any of their personal information to be routed in such a fashion. This taken into consideration, the possibility for any user to flag his or her information as private at all times must be implemented.

The way to display information on a personal display in the prototype described above is somewhat crude. We have taken the concept of the scrolling banner and translated this to be displayed in the same manner on a personal device. This to create a consistency between the different information sources in the room. The personal banner running on a handheld computer shows only private information. There is consistency, but there might be too much of a possibility for others to see what is shown on such a display so the design concept might have to be revised. For the user it might be good to build upon a known concept of information flow, therefore GUI ideas taken from instant messenger design (ICQ [19], AOL instant messenger [20], MSN instant messenger [21]) have been discussed and might be implemented before deploying a full field test. In systems like these any message that is sent to a user also stays in the user's environment until the user acknowledges it, something that has been asked for by users of the described system. The system, as well as other users, could communicate with the user in this fashion. When intelligent agents take part in the work 'behind the scenes', this might also be a known and easily adopted way for the user to interact and understand them, agents otherwise somewhat 'faceless' and hard to understand.

The idea of creating a soundscape, (WISP [14]) using information as a generator of sound is found interesting by several users. There are caveats to take into consideration here though; such as the semantics. Using sounds of nature, such as bird song or the rustle of leaves to convey an information signal to a user is something that requires training the user to do a mental mapping of a special sound to a special occurrence. There might also emerge cacophonous sounds if lots of information is routed to this channel at once, resulting in high degrees of disturbance and irritation. The idea though is liked by many, found as having something of a soothing and stress-lowering value. We will further gently explore it.

This far, real field-study with a fully implemented environment has not yet been conducted. The research team is planning one in the near future, where the setting is at university campus and the participants will be second-to-third year students taking part in project-based courses. These students are today all equipped with the artefacts mentioned above (e.g. mobile phones, wirelessly connected laptop computers and handheld computers) and will in their project-based courses be working in smaller teams, consisting of up to eight participants. We plan not to do these studies as lab oriented pre-defined tasks for the groups, but rather as ethno-methodological observation of the real project work conducted in the environment.

5 References

1. McFarlane, D. Interruption of people in humancomputer interaction: A general unifying definition of human interruption and taxonomy. Technical Report 5510-97-9870, Naval Research Laboratory Report, 1997.
2. Broadbent D. and Gillie T. (1989), *What makes interruptions disruptive? A study of length, similarity and complexity*. Psychological Research, 50, 243-250.
3. Nardi, B. (1996) *Context and Consciousness: Activity theory and Human-Computer Interaction*, edited by Bonnie A. Nardi, The MIT Press, Cambridge, Massachusetts, London, England
4. Engeström, Y. (1987). *Learning by expanding*. Helsinki, Orienta-Konsultit.
5. Fox, A., Johanson, B., Hanrahan, P., Winograd, T. Integrating Information Appliances into an Interactive Workspace. IEEE Computer Graphics and Applications, vol 20, No. 3, May, 2000, 54-65.
6. Streitz, N.A., Geissler, J., Holmer, T. and Konomi, S. *ILand: An interactive Landscape for Creativity and Innovation*” In proceedings of ACM Conference on Human Factors in Computing Systems: CHI 1999: ACM Press. Pp. 120-127, 1999.
7. Weiser, M. The Computer for the 21 st Century. Scientific America, 265(3) (1991) 94-104.
8. Anderson, J.R, *Cognitive Psychology and its implications*, Worth Publishers and W.H Freeman, 2000
9. Espinoza, F., Persson, P., Sandin, A., Nyström, H., Cacciatore, E. and Bylund, M. “*GeoNotes: Social and Navigational Aspects of Location-Based Information Systems*.” in the Proceedings of the Ubicomp 2001 Conference, Atlanta, Georgia, September 2001.
10. Jini is a product from Sun Microsystems <<http://www.sun.com/jini>>
11. Patrik Werle, Fredrik Kilander, Martin Jonsson, Peter Lönnqvist and Carl Gustaf Jansson, “*A Ubiquitous Service Environment with Active Documents for Teamwork Support*”, in the Proceedings of the Ubicomp 2001 Conference, Atlanta, Georgia, September 2001.
12. TINI is a product of Dallas Semiconductor Corp
<<http://www.ibutton.com/TINI>>
13. iButton is a product of Dallas Semiconductor Corp. <<http://www.ibutton.com>>
14. Kilander, F., Lönnqvist, P. *A Weakly Intrusive Soundscape for Intuitive State Perception*. Position paper at the continuity in Future Computing Systems workshop at the I3 2001 Spring Days, Porto, Portugal.

15. Jennings, N. R., Wooldridge, M. (editors): Applications of Intelligent Agents. In Agent Technology: Foundations, Applications, and Markets. Springer-Verlag (1998)
16. Redström J, Skog T & Hallnäs L. Informative Art: Using Amplified Artworks as Information Displays. In Proceedings of DARE 2000, Designing Augmented Reality Environments, ACM, 2000, Elsinore, Denmark.
17. Weiser, M. & Seely Brown, J. (1996). *Designing Calm Technology*. In: PowerGrid Journal 1.01. Available at:
<http://www.powergrid.com/1.01/calmtech.html>
18. Ishii, H. & Ullmer, B. (1997). *Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms*. In: Proceedings of CHI '97, pp. 234-241. ACM Press.
19. <http://www.icq.com>
20. <http://www.aol.com>
21. <http://www.microsoft.com>