The Personal Bucket Organizer

Supporting Spatially Distributed Interaction on a Waste Water Plant

Tomas Sokoler, Thomas Binder, Jörn Nilsson, Nina Wetcke
Space & Virtuality Studio
The Interactive Institute
Beijerskajen 8, 205 06 Malmö,
Sweden
+46-406657107

E-mail: {Tomas.Sokoler, Thomas.Binder, Jorn.Nilsson, Nina.Wetcke}interactiveinstitute.se

ABSTRACT

The work of process plant operators involves continuously shifting between interaction with physical objects distributed throughout the plant and interaction with the digital representations of these same objects. These shifts between interaction domains are poorly supported by the centralized control room structure usually applied in process plants. We describe a system designed to smooth the transition between the two domains of interaction. The 'Personal Bucket Organizer' system is designed to support flexible and spatially distributed interaction with the plant through the use of a custom built handheld personal device hereafter named the Pucketizer. The Pucketizer design introduces the use of a Bucket metaphor as its organizing principle. The Pucketizer enables the user to, while walking around on the plant, a) establish digital links to objects on the plant, b) Audio annotate objects on the plant, c) Monitor the state of objects on the plant. The Pucketizer was designed with the active participation of operators working on a local waste water plant.

Keywords

Spatially Distributed User Interfaces, Augmented Reality, Ubiquitous Computing, Mobile Computing, Work Oriented Design

1. INTRODUCTION

The use of information technology on waste water plants, and in process control in general, has for many years been synonymous with the idea of having a centralized control room as the main gateway to information about, and control of, the plant. In the centralized control room architecture a server collects data from sensors distributed on the plant and presents the plant operators with mainly visual representations of the data.

The operator's main role is to monitor the plant's state via these representations and the alarms initiated by the system. Operator intervention in most cases involves physical inspection of the components on the plant. This implies a shift in the interaction domain from interaction with digital representations off the plant to interaction with the physical components. A strictly centralized control room model inherently precludes a smooth transition between these two domains of interaction. Further on, observing waste water plant operators go through their daily routines it becomes clear that physical inspection goes beyond a simple 'get an alarm, find the error and fix it' scheme. While walking around on the plant the operator uses all his senses, his expertise, and his accumulated knowledge of the plant to get a feel for the plant's current state. Interaction with the plant during inspection is not only a matter of highly focused data collection and hands on adjustment of physical components (vents, pumps etc.), but involves a more subtle mode of interaction simply taking in impressions from the plant. Peripheral awareness expressed as the ability to make use of informational resources in the environment on and near to subconscious level of attention seems to be an invaluable part of the daily inspection.

We propose a system that supports the process of physical inspection and attempts to make the transition between physical interaction with the plant and interaction with digital representations of the plant smooth.

2. THE WORLD OF PROCESS PLANTS

In early studies**,** automation of process control was expected to reduce the role of the operator to a 'machine-minder' with no need for manual skills that only intervened when process information deviated from specified norms [1], [4]. Zuboff [15] have later argued that computerized process control systems force operators to leave their manual skills behind to develop the more intellectual skill of operating a process through symbolic representations on a computer display. However, in more recent studies it has been argued that knowledge of manual operation and machinery and knowledge of computerized process control are two inseparable components of operator skill. The process operator rely very much on the ability to understand the process through various representations, where process information on computer displays is just one form of representation. In particular operators need the ability to bridge the gap from symbolic representations on computer screens to a detailed understanding of the machinery on a physical level coupled with tacit knowledge of process dynamics [8].

With this in mind we designed the Pucketizer system with active participation from operators at a local waste water plant. We started out following three operators going through their daily routines and video taped two full days of work at the plant. These rounds of observation and informal interviews were followed by two workshops held at the plant. The workshops included brainstorming, enactment of scenarios, and discussions centered around paper mockups and foam models prepared by the research group illustrating several design ideas [3]. Through these workshops researchers and operators developed a common understanding of the problem area and the design process converged towards a design concept for, what later on was to be named, the Pucketizer.

The importance of physical inspection

A central observation that emerged from our collaboration with the process operators was that physical inspection of the plant plays a crucial role. The round is not only establishing the individual 'mapping' between system representations and actual state of plant components. It is also helping the operators to maintain a shared understanding of the process.

Figure 1: The Operator takes several daily rounds
A process operator goes through 'his area' on the computer. He has his own logs to keep track of parameters that he knows are critical. Later he will walk through the plant to listen to and sense such things as pump vibrations, valve operation and sludge quality. If time permits, he does adjustments and optimizations.

Every operator is making one or more non-alarm driven rounds of inspection everyday, following a more or less fixed route through the area of which he has responsibility. During this round he uses all his senses, and is equally attentive to the operation of components as to the quality of the processed sludge.

During the round the process operators are also getting an overall picture of what is going on at the plant. They occasionally 'bump in' to one another and exchange information but they learn as much from interpreting traces of their colleagues' activities (could be tools left for later use or dismounted components).

Even though the operators typically follow a fixed path the 'points of interest' on the plant seems to be constantly shifting. A certain part of the plant may be out of operation, and this will cause the operator to pay particular attention to other parts that may be running 'heavy duty'. It could be that a component during the round has appeared to be 'ready for a breakdown', so the operator has to have an eye on that particular section of the plant. And it could also be that the sludge coming in has certain properties that put stress on certain parts of the plant.

Alarms are not always important

In the plant we worked with alarm messages are immediately sent to the operator responsible for the area. He receives the alarm on his pager if he is not at a monitoring station, and he has to sign off the alarm personally on the monitoring station. Despite this we found that alarm handling only plays a minor role in keeping the plant running. Very often sections of the plant are under repair or maintenance and this frequently causes alarms that do not call for action (see figure 2). In other situations the actions of the operators in themselves causes alarms e.g. because a level meter gives a false reading. On the other hand it is often so that operators try to foresee situations that may cause problems before an alarm or even a warning has been sent out. E.g. the cluttering of a pipe or a valve is best dealt with if the problem is detected before it is detected by the monitoring system.

For the operators the focus on alarm handling in the conventional design of control and monitoring systems appear to distract attention from a more deliberate focus on upcoming problems.

An experimental approach to problem solving

When process operators identify a potential problem in a particular area, they often engage in a series of experiments in order to find out what relevant measures have to be taken. If e.g. a pump vibrates excessively, an operator might choose to examine if a parallel pump will be able to handle the flow on its own. Such experimentation will often involve setting up a problem specific configuration of monitoring devices at different places in the plant.

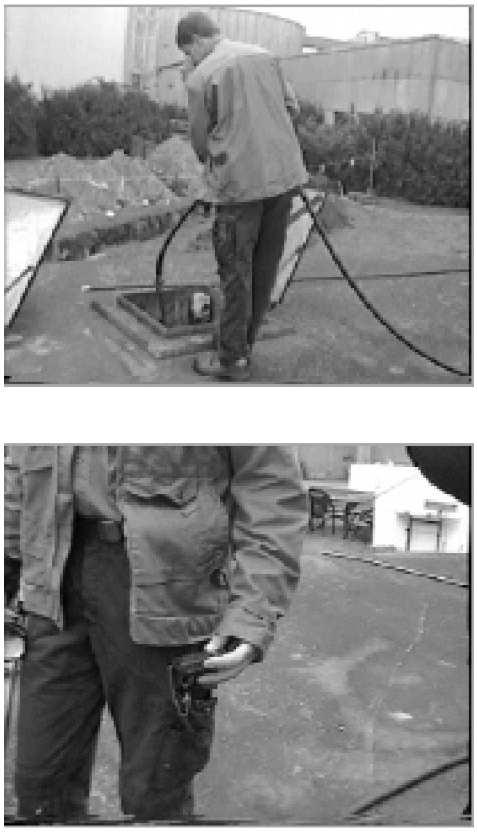

Figure 2: Alarms can often be ignored:
Most SCADA systems are designed for operators to act only on alarms. Reality is however often quit different. A lot of alarms are caused by well-known and unproblematic events. When e.g. a process operator flushes a tank to avoid sediments he triggers the level meter and gets an alarm. Even though he is on the spot, he can only see the alarm at his pager. To cancel the alarm he has to go to one of the SCADA workstations and log in.

Monitoring is here rarely restricted to observing control room information. Typically the operator has to set up monitoring devices on different components as well as monitoring the resemblance of data obtained from different places in the chain from sensors to computer monitoring system. Frequently the sensous perception of the operator of e.g. sound or smell on particular spots form an integral part of the diagnostic activity. Shifting between different domains of interaction introduces a discontinuity in the operators' workflow, because coordination of observations in the plant and information presented by the centralized control room system is poorly supported.

3. TOWARDS DISTRIBUTED INTERFACES

What emerged in our collaboration with the process operators was the incresingly clear picture of a guiding image in SCADA systems design that begs to be challenged. Rather than designing control room installations with a claim for the perfect centralised information support of a 'panopticon', we wanted to dissolve the static user interface with its fixed views of the process.

The aims were to make it possible for the operator to create and modify his own points of interaction on the locations and at the times of his own choosing. This would make it possible to get away from a situation, where the operator has to leave his current work context in order to obtain information or gain control.

Further more we wanted to support a more continuous transition between interaction with focal points selected during physical inspection and interaction with the corresponding representations in the digital domain. These aims eventually led to the Pucketizer design.

The Pucketizer and the Bucket metaphor

The Pucketizer design revolves around a Bucket metaphor for interaction with the plant. The underlying idea being that the operator while walking around on the plant can grab components of interest and group these components into one or more Buckets. Obviously its not the actual physical components that are grabbed and kept in the Buckets but a representation establishing a link to the components. The grouping of components within a Bucket is left entirely open to the operator thereby enabling him to create his own problem specific view of a possible interdependency between components. The Buckets are carried along and represents the operators personal collection of work activity focal points. The Buckets contains a minimal visual representation (icons) of the components collected and whenever the operator needs to take a closer look at a specific component the content of a Bucket can be 'poured' onto one of many displays distributed throughout the plant.

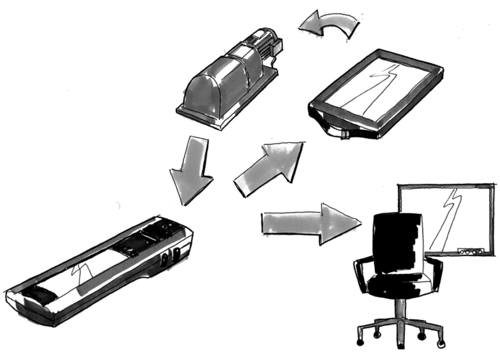

Figure 3: The Pucketizer System

The Pucketizer system consists of:

A handheld unit (the Pucketizer, see Figure 3) containing the Buckets. The Pucketizer serves as the operators interface to the plant and is used by the operator for the grab and pour operation on components. More operations available to the operator are discussed later.

The physical components already present on the plant including pumps, motors, vents, and numerous sensors.

A number of displays in different shapes and sizes distributed throughout the plant. Some of these displays are mobile and constantly travels the plant following the focal points of work activities. The displays serve several purposes as discussed later.

Figure 4: The handheld Pucketizer unit

Grabbing components into a Bucket and pouring components onto a display are seen as the two basic functions provided by the Pucketizer. Any interaction with components via the Pucketizer starts with the grabbing of a component. Figure 5 shows Per, an operator at the waste water plant, using an early foam model of the Pucketizer to illustrate how he would grab a component.

It is important to note that the selection of the component to grab is done simply by pointing at the physical component without entering any symbolic reference to the components ID. This frees the operator from the cumbersome task of mapping physical components to their symbolic names before grabbing them. Standing in front of a component the operator already knows that this it the component he wants and going through any further component identification seems like a waste of effort. The notion of a collapsed name space [2] facilitating information management through links attached to physical objects has an immediate use in the Pucketizer system. The physical objects are already present on the plant and the existing central server contains the digital information, hence, only a tagging mechanism sensitive to the Pucketizer pointing (implementation discussed later on) needs to be added. Displays are seen as a subset of the component domain and pouring a Bucket's content onto a display is done by selecting the Bucket on the Pucketizer and pointing at the display. The components pointed at thereby determine whether the Pucketizer grabs or pours.

The Pucketizer as memorizer and annotator

The process operator is not only using the Pucketizer to grab components for later use. It can be seen as a memorizer device in the sense that it bookmarks and keeps a reminder of particular points of interest. In the prototype we have implemented we have even included the opportunity to monitor a core value of the memorized components.

Figure 5: Grabbing a component to a Bucket within the Pucketizer

As the components memorized in the Pucketizer have no need for further indexing it has been easy to include the opportunity for the operator to annotate the grabbed components. After grabbing a component the operator can attach an audio post-it note to it. The audio can be accessed as long as the component is present in one of the Buckets carried by the operator. Audio notes serves two purposes: making comments for the operator's own later use; and telling other operators about activities relating to the component. Each component has a 'voice mail box' attached and operators automatically gains access to the mailbox when grabbing the component. In this way the Pucketizer enables the operators to extend the practice of leaving traces of their activities 'on location'

In principle the Pucketizer system opens up for a more active configuration of process monitoring including temporary re-instrumentation. In one of the scenarios we developed together with the group of process operators, the Pucketizer was used together with a mobile display and a wireless fieldbus connection to set up local monitoring of electrical current and flow.

4. THE PUCKETIZER USER INTERFACE

We decided early on in the design process to build a customized Pucketizer unit as opposed to implementing the Pucketizer functionality on one of the commercially available PDAs. This decision gave us the freedom to specifically support the Bucket metaphor without having to force our ideas on top of a pre-existing general purpose interaction scheme. We also decided to build a prototype under the constraints of using standard off-the-shelf components. This meant that the possibilities for designing the visual information content shown in the 122 x 32 pixel display was strongly limited. We chose to strive for a 'flat' and simple design trying to avoid software buttons and menu hierarchies.

The Pucketizer is operated by the use of 6 buttons and a rectangular display shows the current state of Buckets and components in these Buckets. The 6 buttons have the following functions:

Bucket selection. By pressing this button the Pucketizer advances to the next of the 4 Buckets available. Whenever a Bucket is selected its components are shown in the Bucket display area.

Selection of components already held in the current Bucket. By pressing this button the Pucketizer advances to the next component in the current Bucket.

Grabbing a component. Pressing this button activates the Pucketizer's laser pointer. Holding the button down and pointing the Pucketizer at a physical component in the environment makes an icon of that component appear in the Bucket display area. Still holding the button down while moving the Pucketizer as you would move a search light scanning the environment the Bucket display area continuously shows you the icon of the last physical components pointed at. When the button is released an icon of the last physical component pointed at is grabbed and kept in the current Bucket.

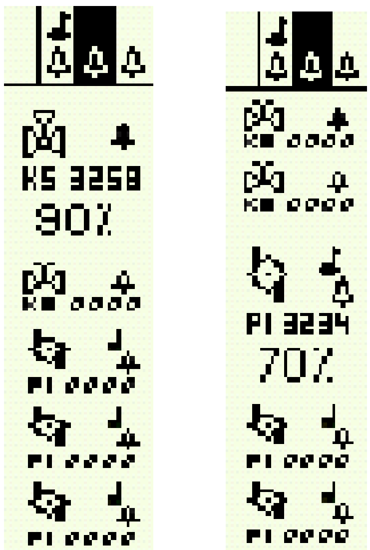

Figure 6: Pucketizer display

Removing a grabbed component. Pressing this button removes the currently selected component from the current Bucket.

Leaving an audio note. Pressing this button initializes the recording of an audio note to be left at the component currently selected. Recording ends when the button is released.

Listening to an audio note. Pressing this button initialize the playback of an audio note found at the component currently selected. Playback ends when the button is released.

5. RELATED WORK

The work reported has been inspired by research in ubiquitous computing [12], augmented reality [13] and tangible bits [5]. There are currently numerous approaches to augmenting physical objects. The *Informative Things* approach is proposed by Barrett & Maglio as a new approach to information management [2]. Links are created between physical objects and digitally stored information giving the impression that the information is stored on the object and eliminating the need for creating and managing symbolic references to the information. In the described implementation floppy disks are used as objects with the ID stored on the disk, requiring no extra hardware to read it.

The *Insight Lab* is an immersive environment supporting teams in creating design requirements documents [6]. The connection of physical design documents to digital information is one element of the concept. Whiteboard printouts and paper documents are linked to associated multimedia data stored in a computer using barcodes as identification.

Barcodes are also used for tagging in *WebStickers* which is a low-cost method for associating web pages with physical objects [7]. A sticker with preprinted barcode is attached to the object, which is then linked to one or more URL:s. The links are stored in a networked server and the URL can later be retrieved by scanning the barcode.

Want et al [11] argue that, while the low cost of using for instance barcodes for tagging allow larger numbers of augmented objects and support multi-location use, the visual obtrusiveness of the tags and the akwardness of the readers limit their use. Instead they propose RF ID tags for augmenting objects already naturally occuring in the environment, providing a more seamless interaction by being unobtrusive, and still using inexpensive infrastructure.

In the context of process control the Pucketizer provides inherently unobtrusive tagging since the infrastructure for linking already is in place. Also, as mentioned above, the physical objects referred to are already in focus in the work activities of the process operator, providing a more seamless interaction with the environment. The Bucket metaphor also introduces the possibility for organizing the established links with the same device as used for tagging and annotation.

Another related approach is Pick-and-Drop [9] which is a direct manipulation technique allowing a user to exchange information in multi-computer environments. By recognizing ID:s of pointing devices an object can be picked up from one computer screen and dropped on another, much like physical objects are moved without the need for symbolic references to locations. The notion of Pick-and-Drop relates to the Pucketizer concept on a more abstract level. The Pucketizer allows the user to ‘pick up’ physical objects in the work environment and then ‘drop’ them onto different displays (or rather symbolic references to them). The idea of having various displays available in the work environment that are not regarded as distinct computers also corresponds to the notion referred to by Rekimoto [9] as ‘Anonymous Displays’.

Finally, the audio annotations of objects provided by the Pucketizer correspond to the notion of augment-able reality introduced by Rekimoto et al [10] where augmenting information can be created dynamically and attached to the user’s surrounding physical environment. The informaion is then shared by users with wearable computers and networking facilities. However, the ‘situated information’ can also be accessed with other technology, e.g. from a desktop computing environment using a digital representation of the physical environment.

6. SYSTEM PROTOTYPE

A functional laboratory prototype of the Pucketizer system was implemented with a custom built handheld Pucketizer unit controlled by an 8-bit micro controller, a standard PC running a JAVA application under Windows95, and hardware for wireless radio communication and identification of components.

Figure 7 shows the overall hardware architecture of the Pucketizer system and the communication paths between PC, Pucketizer, and the laser eyes used for component identification. The current implementation does not include small displays distributed in the environment but uses a standard PC monitor for the time being.

The communication between server and Pucketizer transfers digital data as well as analogue audio via two radio transceivers (BIM 814MHz). All user actions on the Pucketizer are reported to the server through the digital data channel and a copy of the current state of Buckets and components on the Pucketizer is kept on the server. A simple sliding window protocol is implemented on the server and Pucketizer to control communication of digital data. The digital data packet size is 10 bytes and transferred at 4800 baud.

The server application is written in JAVA and uses native method calls and Windows DLL’s for communication with the PC soundcard and serial port. Besides answering requests from the Pucketizer the server maintains a component database including a directory containing the voice message boxes linked to each component. The server uses the serial port for digital data communication with the transceiver and the component identification hardware (laser eyes).

Audio notes are transmitted in analogue form from the Pucketizer to the server and delivered to the PC's soundcard after demodulation and amplification. The server JAVA application produces standard .wav-files of the recorded audio notes and saves these files in the voice message boxes linked to each component. Play back of audio notes takes the reverse path from .wav-file through the soundcard to the radio transceiver that transmits the audio in analogue form to the Pucketizer.

Basic Stamp II micro controller programmed in Basic. The current laboratory prototype has four laser eyes attached to the Stamp. Since no actual physical components (pumps, vents, etc.) are included in the laboratory prototype we attached the laser eyes to cardboard mockups of four components. The Stamp communicates the component ID to the PC using the PC's RS232 serial port.

7. CONCLUSION AND FUTURE WORK

We have described the Pucketizer system that was designed to smooth the transition between interacting with physical objects in process control and digital representations of the same objects. Main functions include establishing links to physical objects that are grouped in Buckets, remote monitoring of readings from linked objects, and the annotation of each link with audio post-it notes. The work has been carried out as a participatory design process involving process operators in a waste water treatment plant. In the process control context, the Pucketizer system opens up for a more dynamic and flexible configuration of process monitoring than provided in a traditional centralized control room context.

We have also come to the conclusion that the Pucketizer has generic qualities that could be further explored. The concept of using a handheld device for 'collecting' and grouping links to physical objects in order to later manipulate their digital representations in other contexts seems transferable to other application areas. The concept can also be extended to include linking to digital objects. In

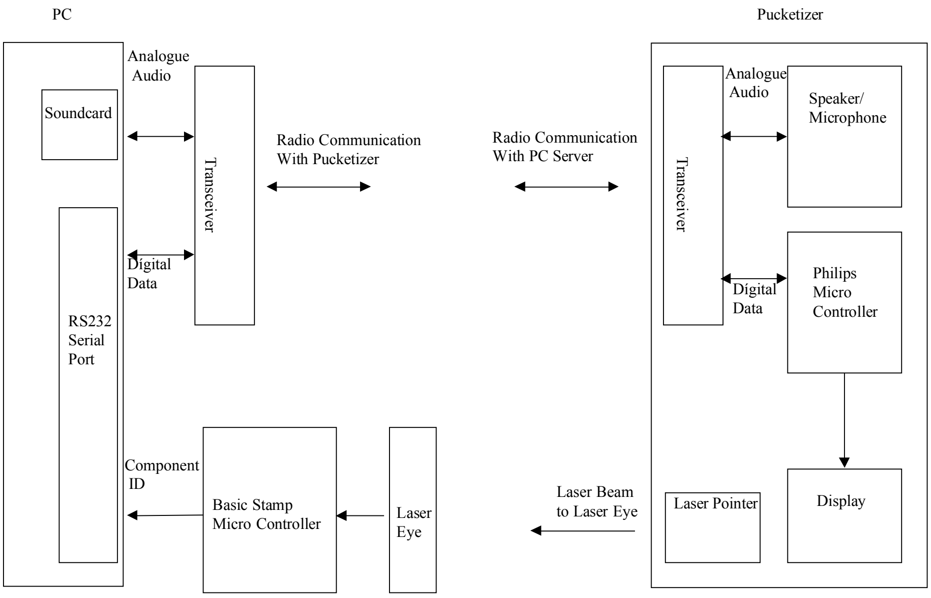

Figure 7: Pucketizer system hardware

The Pucketizer has an ATMEL 8955 8-bit controller at its core. The controller's main task is to control communication with the server. User button actions are detected by polling. The controller maintains a simple table of Bucket content and a predefined table of generic icons used when displaying grabbed components. The controller displays the icons through the display controller (SED 1520) in charge of updating the ASI-D-1223A display. Audio notes are recorded and played on a purely analogue basis and only involves the transceiver, an amplifier circuit, and the microphone/speaker.

The laser eyes used for detection of the Pucketizer laser pointer is implemented using a photo transistor and a schmitt trigger circuit controlling the clock pin on a digital flip/flop. Reading and resetting the flip/flop is done by a an interactive workspace, as described by Winograd & Guimbretiere [14], with shared digital objects visible on a wall-mounted display for group interaction, the Pucketizer could allow each participant to collect digital objects in their personal buckets for later use.

Future work involves implementing the display side of the Pucketizer system and evaluating the prototype system in process control contexts. We will also further explore the generic qualities of the Pucketizer concept in other use contexts.

8. ACKNOWLEDGEMENTS

The authors want to thank Henrik Janssen and Lars Malmborg from Nef Engineering for their work on the hardware design and CAD drawings for the Pucketizer handheld unit. Sincere thanks are also due to the process operators at Sjolunda waste water plant (Malmö, Sweden) for their stimulating cooperation in the design process.

9. REFERENCES

1.Bainbridge L. The Process Controller. In Singleton W T (ed.) *The Analysis of Practical Skills*. MPT Press Ltd, Edinburgh 1978.

2.Barrett, R., and Maglio P. P. Informative Things: How to attach information to the real world. *Proceedings of UIST'98, ACM Symposium on User Interface Software and Technology*, pp. 81-88, October 1998.

3 Binder, T. Setting the Stage for Improvised Video Scenarios. *Proceedings of CHI'99*, Pittsburgh, 1999.

4.Crossman E. R. F. W. Automation and Skill. In Edwards Elwyn & Lees Frank P (eds.) *The Human Operator in Process Control*. Taylor & Francis Ltd, London 1974.

5.Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proceedings of CHI'97*, pp. 234-241.

6.Lange, B. M., Jones, M. A. and Meyers, J. L. Insight Lab: An Immersive Team Environment Linking Paper, Displays and Data. *Proceedings of CHI'98*, pp. 550-557.

7.Ljungstrand P., and Holmquist L. E. WebStickers: Using Physical Objects as WWW Bookmarks. *Proceedings of CHI'99.*

8 Perby M. The Art of Mastering a Process – on the Management of Working Skills (in swedish). Gidlunds forlag, Smedjebacken 1995.

9 Rekimoto, J. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. *Proceedings of UIST'97, ACM Symposium on User Interface Software and Technology*, pp. 31-39, October 1997.

10.Rekimoto, J., Ayatsuka, Y. and Hayashi, K. Augment-able Reality: Situated Communication through Physical and Digital Spaces. *Proceedings of ISWC'98, 2nd. International Symposium on Wearable Computers*, 19-20 October, 1998, Pittsburgh, Pennsylvania.

11.Want, R., Fishkin, K. P., Gujar, A., and Harrison B. L. Bridging Physical and Virtual Worlds with Electronic Tags. *Proceedings of CHI'99*, pp. 370-377.

12.Weiser, M. The Computer for the 21st Century. *Scientific American*, 265 (3), 1991, pp. 94-104.

13.Wellner, P., Mackay, W., and Gold, R. Computer Augmented Environments: Back to the Real World. *Commun. ACM*, Vol. 36, No. 7, July 1993.

14.Winograd, T. and Guimbretiere, F. Visual Instruments for an Interactive Mural. *Proceedings of CHI'99, Extended Abstracts*, pp. 378-385.

15.Zuboff S. In the Age of the Smart Machine - the Future of Work and Power. Heinemann Professional Publishing, Oxford 1988.