CallMeSmart: A VoIP Softphone on Android based mobile devices using SIP

[a] Terje Solvoll, [b] Lorenzo Gironi, [c] Alain Giordanengo, [d] Gunnar Hartvigsen

[a,b,c,d] Norwegian Centre for Integrated Care and Telemedicine, University hospital of North Norway, Tromsø, Norway

[a,d] Department of Computer Science, University of Tromsø, Tromsø, Norway

[a] terje.solvoll@telemed.no, [b] lorenzo.gironi@telemed.no, [c] alain.giordanengo@viacesi.fr, [d] gunnar.hartvigsen@uit.no

Abstract – **The Wireless Communication Infrastructure represents a core for information sharing between health care workers in hospitals: Medical staffs work situation is highly mobile, and important information is constantly shared between health care workers to provide high quality service for the patients. Physicians carry mobile communication devices to be able to communicate in their mobile work, and often several wireless devices according to their role and responsibilities. This leads to a number of problems, especially regarding interruptions from these devices. Such interruptions are often due to the caller is unaware or ignoring the situation and context, in which their colleagues are. This can, and often does lead to severe medical consequences. This article deals with the CallMeSmart system (CMS); a communication infrastructure based on collection, analysis and dissemination of context sensitive information through a communication system based on smartphones and DECT devices, to improve the current communication backbones, and to reduce interruptions from mobile devices in hospital settings.**

Keywords - Context awareness; wireless devices; mobile communication; Interruption management;VoIP

I.

II. INTRODUCTION

Activities within hospitals and healthcare settings require reliable communication systems. Sharing information between colleagues, medical attendants, investigatory facilities and other resources, using wired/wireless communication systems is a necessity. This often results in a lot of communication events. Clinical questions are often complex and not clearly defined, and will therefore require frequent conversations and discussions [1]. Devices currently used to communicate at hospitals, are mainly pagers. Wired/wireless phones are less utilized, but also in use. Personal Digital Assistants (PDA) has also been tested for use in some hospitals [2]. More and more hospitals are using wireless phones based on Digital Enhanced Cordless Telecommunication (DECT) or Voice over Internet Protocol VoIP) and Session Initiation Protocol (SIP), like the devices used in [3]. These devices can both be personal and role-based, since communication in many cases is not aimed to one person, but to a role such as; 'the nurse on call', or 'the physician on the next shift' [4]. Because of this, some staff members are carrying multiple devices for different roles and purposes [2, 5].

However, mobile communication in hospitals has shown to suffer from poor practice and inefficiency caused by an insufficient infrastructure, especially when the communication need is urgent [1, 4, 6]. A more extensive use of mobile phones can offer a solution to this problem by improving accessibility and communication in hospitals [1, 6, 7]. Compared to the usage of pagers, important advantages can be achieved by offering two-way text and voice services. Providing smaller delays in communication may lead to improved patients care, and also to reduce the risk of medical errors [6].

Despite the advantages of mobile phones, there are also well-known downsides to the usage of these devices. The increased availability and accessibility can cause an overload of interruptions on key human resources, such as, senior physicians, or 'on call' staff [5, 8]. These interruptions can lead to a diversion of attention, errors, and may disturb in situations such as, in outpatient clinic, or in the operating theatre [5, 8]. A context-sensitive system can provide a solution to control availability and interruptions [5]. Context based on the phones' location, a person's role and schedule, interruptions can be avoided, and calls can be redirected to other available resources. Combining the personal and role based devices into one single device, will also offer an improvement to the mobile communication [3, 8].

In this article we present a prototype of a VoIP context aware softphone, based on the Android operating system, integrated in a complete context sensitive communication system for mobile communication in hospitals. The system is built on top of existing infrastructure, as explained in [9].

 ISBN: 978-1-61208-252-3

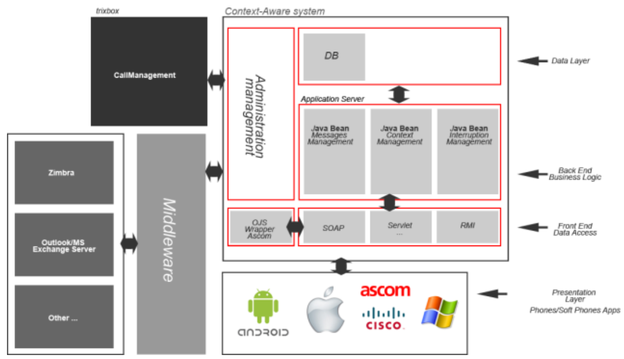

Figure 1: Overall system architecture of the context aware interruption management system, CallMeSmart [9]

III. BACKGROUND

In general, mobile phones are currently not widely in use in hospitals. Only a few staff members carry a wireless phone due to the assumptions of that a phone is more interruptive than a pager [2, 5]. Before introducing wireless phones as standard hospital equipment, usability and user satisfaction are important factors to account for. A study, carried out by the first author, regarding the usage of mobile phones at St. Olav's Hospital in Trondheim, mid Norway (an early adopter of implementing wireless phones), observations and interviews showed that the users were unsatisfied by the current user interface of the phones [3, 10]. It kept them from using all functions of the system, especially the way messages were handled. The feedback from the interviewed and observed physicians was then used to design and develop a prototype for a context based communication system [9] which we have called *CallMeSmart*. Figure 1 presents the overall system architecture of CallMeSmart. The method used here was based on a participatory design process [11] and heuristic evaluation [12, 13], where we used input from the users to design and develop, and then tested the system with real users according to scenarios' from health care settings, adjusted the system according to the feedback, and then tested again, adjusted, and so on. Due to the fact that the cost of replacing a complete communication system will be enormous for a hospital, this system where developed on top of an existing communication system and infrastructure based on DECT, where we re-routed the signals from the DECT system to our context server. Then we used collected context data from the users to control the communication, and thereby avoid interruptions. We believe that utilizing existing systems and infrastructure will be cheaper and experience less user resistance, and thereby it is up to the user and management at the hospital, which devices the user should use and carry. This opens up for including new devices and features together with older communication systems and infrastructure, like including smartphones.

IV. MATERIALS AND METHODS

The phones subjected in this paper are commercial off-the-shelf Android based mobile devices; smartphones and tabs. Android based devices are already widely used both by private and professional users, which means that this is known devices and user interfaces for a lot of users. The devices were configured with the CallMeSmart (CMS) SoftPhone. The softphone is based on VoIP using SIP, offering voice and text services, but also role-based communication, alarms, and pager services, and are controlled by context information, based on definitions in [14], to control the communication and to avoid unnecessary interruptions. In this Section we present the subjected mobile devices, and the method used to develop and test CMS SoftPhone.

A. Mobile devices

1) Samsung Galaxy ace

This mobile phone was set up with Android 2.3.3 and TouchWIZ User Interface (UI).

2) Samsung Galaxy SII

This mobile phone was set up with Android 2.3.5 and TouchWIZ UI installed, but was updated to Samsung original Android 4.0.3.

3) Samsung Galaxy Tab 7"

This Tablet was set up with Android 3.2 and TouchWIZ UI installed.

4) Samsung Galaxy SIII

This mobile phone was set up with Android 4.0.4 and TouchWIZ UI installed.

5) HTC Desire

This mobile phone was set up with Android 2.1 and Sense UI installed, but was updated to HTC original Android 2.2 and 2.3.3. It was also tested with a rooted Android 4.0.4.

6) HTC Sensation XE

This mobile phone was set up with Android 2.3.4 and Sense UI installed.

B. Methods

The software engineering approach used to develop the context-aware system is based on the Unified Process. An iterative and incremental development methodology (also known as spiral development or evolutionary development) based on the ideas of Boehm [15] and Gilb [16]. This approach split the development process into a series of short mini-project, called iterations. The purpose of an iterative approach is to increasingly enlarge and refine a system within each iteration, in order to gradually approach the requirements of the targeted application. An iterative model does normally not start with a full specification of the requirements, but begins with specifying and implementing only the most important features, which are subsequently improved and adjusted to include missing requirements during next iterations. Each iteration includes:

- Requirements: Identified, collected and analyzed.

 ISBN: 978-1-61208-252-3

- Design: a software solution is designed by using Use Cases diagrams to capture the functional requirements, Interaction diagrams are used to define the interactions between software components and other graphical Unified Modeling Language (UML) notation models are applied to better define the overall architecture of the software.
- Implementation: Program the software described in the previous step, improving the system already developed.
- Testing: New developed features are tested in order to verify if they are consistent and without errors.

If the requirements are not met after these steps, a new iteration takes place.

The tests were carried out by simulating typical scenarios within health care settings, where the functionalities of the application were tested for quality and stability.

V. RESULTS

CallMeSmart SoftPhone is a context-aware SoftPhone, based on the Android operating system, specifically designed for hospital usage. The system have been tested through scenarios', experienced during the first authors fieldwork [3], in our context-sensitive laboratory at Tromsø Telemedicine Laboratory (TTL) hosted by Norwegian Centre for Integrated Care and Telemedicine (NST). Together with the context-aware system, CMS [9], on which it relies on, it is able to change its behavior according the context of the user carrying the device. It supports three different operating modes which automatically is controlled by CMS, but can be manually overridden by the users: "Available", "Busy" and "Pager Mode". The functionality is the same as on the DECT phones in [9]. The Available mode makes the phone fully reachable both for calls and messages with the ringer on. In busy mode the phone receives only calls that have been forced by the caller for emergency reasons. And in pager mode, the phone can only be paged through standard text-based messages sending the callers number/name. The CMS SoftPhone also provides a Bluetooth Tracking module, which allows automatic

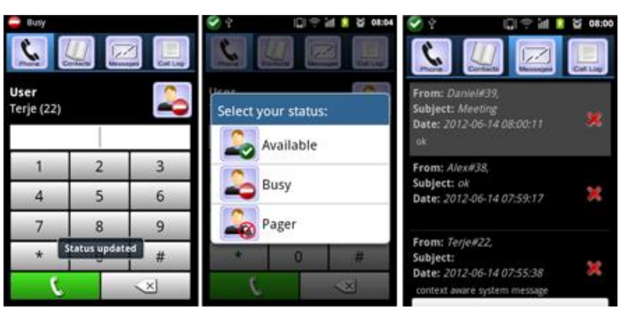

Figure 2: a) CallMeSmart SoftPhone keypad, b) The dialog box from which the user can switch the phone's operating mode manually, c) the Message List

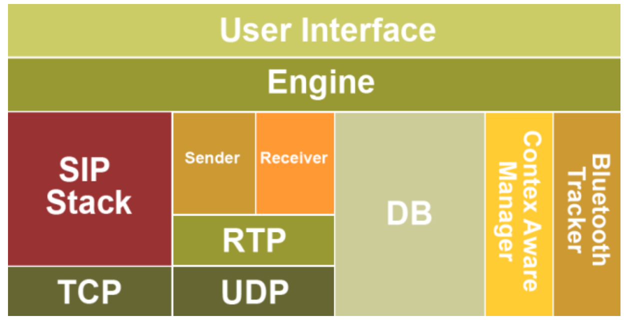

Figure 3: Overall software architecture of the CallMeSmart SoftPhone

tracking of the phones through Bluetooth sensors. The information about location is used by the Context-aware system in order to perform location based interruption management. This is used in the same way as the DECT tracing sensors in [9].

During the development, we put particular emphasis on optimizing the battery usage of the software, since having a long battery life is a mandatory requirement for devices targeted for hospital usage. Among other challenges we addressed, the most important one has been finding the right balance between computational power required by the software, audio quality perceived by the users and number of features introduced by the first version of the prototype of CMS SoftPhone.

Figure 2 shows some screen shots taken from the application's UI:

a) the keypad from which the user can access the operating modes of the phone through a suitable button-icon located on the right side of the display.

b) the dialog box which allow the user to manually switch the operating mode of the phone.

c) the CallMeSmart Messaging system, which enable messaging between other CallMeSmart enabled devices and the Ascom DECT phones. CallMeSmart implements in addition a basic Contact list and a CallLog.

Besides the UI, the others components characterizing the application are: the MjSip SIP stack [17], used for setting up and closing calls, Audio Sender/Receiver for managing the streamed audio, Context-Manager which provides services for communicating with the context-aware system, and the Bluetooth Tracker for tracking the device's location.

The Engine of the application bounds and coordinates together with the previous components, and provides the link between the UI and the rest of the application. Figure 3 shows an overview of the CMS SoftPhone software architecture.

A. *Audio Player/Receiver*

CMS SoftPhone manages the phone's audio through OpenSL ES library [18]. OpenSL ES is a cross-platform audio API tuned for embedded systems which provide the developers with a framework for accessing native audio functionality on a wide range of mobile devices, through a

 ISBN: 978-1-61208-252-3

common API. Among other Native Libraries, this is supported by Android OS since version 2.3 [19]. The use of OpenSL ES allowed us to implement an Audio Recorder optimized for low CPU usage and an acceptable low recording latency, which is a well-known problem between Android developers [20].

We addressed the unpredictability and variability of the network conditions by implementing an adaptive jitter buffer on the phone's receiver, which is a common solution. This solution is adopted for VoIP clients to remove the jitter in the arrival times of the packets. The objective of a jitter buffer is to keep the buffering delay as short as possible, while minimizing the number of packets that arrives too late, by dynamically adjusting the buffer's size containing received audio packets, which are waiting to be decoded and played on the phone's speaker. Our implementation takes in consideration the number of packet loss as well.

One of the problems we had to face was related to Acoustic echo. It occurs when there is a feedback path between a telephone's microphone and the speaker. Moreover acoustic echo can be caused by multiple reflections of the speaker's sound waves back to the microphone from walls, floor, ceiling, windows, furniture, and so on. We faced the echo cancellation on the new Android devices running Android 4.0 (IceCream Sandwich) by using a feature, which allow to tune the microphone of the phones for Voice Communication [21].

B. *Audio Codec*

The default audio codec supported by CMS SoftPhone is G.711 (µlaw), which is one of the most common supported codecs among VoIP clients. It is a lossless data compression algorithm. Audio compressed with this codec requires a bandwidth of 64Kbps. The CPU requirements in order to compress audio with G.711, is fairly minimal. Due to the perceived good audio quality this codec provides, it is a mandatory choice for mobile devices targeting hospital usage, where in many cases the bandwidth supported by the network infrastructure is high and its conditions are well known a priories. The adoption of this codec is also justified by interoperability reasons; the Ascom DECT based communication system, in which our context-aware system integrates, and the majority of other VoIP based systems, supports G.711.

The tests performed over 2G and 3G network with G.711 were not satisfactory, due the noticeable disturbing audio artifacts caused by the low and unstable bandwidth of these networks. This forced us to test utilize other codecs. The tests performed with the codec; Speex [22] and iLBC [23], gave us better results in terms of audio quality. Over 2G and 3G, they proved to be better due to their strong compression algorithm.

Running Speex and iLBC over mobile devices required a significant amount of CPU in order to compress the audio samples. On Samsung ACE (800 MHz ARM processor), the CPU usage during a call reached the peak of 70%, while with G.711 the highest value we experienced was 13%. On Samsung galaxy SII (Dual Core 1.2 GHz Cortex-A9), the CPU peak reached 35% with Speex, while with G.711 only 4%.

The usage of this codecs over a series of long lasting calls could severely decrease the battery life of the mobile devices, and as a consequence, the operating time of a phone. It should be mentioned that these codecs are mandatory if the system requires communication over networks that do not provide a minimum guaranteed bandwidth. We decided to use the Speex codec only on 2G and 3G networks in case of emergency communication, and to keep the G.711 codec on other wireless networks. In order to solve compatibility problem between Android and DECT phones, which do not support the Speex codec, a solution where we are performing a transcoding on the media gateway, when a communication channel is set up between these two different kinds of phones.

C. *Bluetooth tracking*

We implemented the tracking of the smartphones by using Bluetooth adapters as sensors, residing on standard PCs placed inside the areas where we are simulating the system. The discovery of these sensors is performed on the phones, by the Bluetooth Tracker component (see Figure 3), which uses the Android Bluetooth API to connect the device's Bluetooth adapters with our Adapters. Once an adapter has been discovered nearby the phone, the Bluetooth Tracker retrieves its MAC address, and transfers the information to the CMS server, in which maps the MAC addresses of the sensor deployed in the testing environment with name and criticality of the area on which they are localized. The location is subsequently used by the CMS server for providing a location based interruption management system.

D. *Roaming*

Most of the Android phones we tested do not perform roaming within WiFi networks in a seamless way. The time required to switch between two WiFi hotspots is too high; in some cases more than 8 seconds, which making the switching between two WiFi hotspots noticeable during a call. We implemented a solution which keep searching for the best WiFi signal present around the phone, and re-associate the connection of the device to the best hotspot as soon as possible. Even with this approach, the time needed to switch between two WiFi hot spots was not fast enough in order to guarantee a continuous call by some phones, except on the rooted version of HTC Desire. On the HTC Desire it could be used, but this is not the optimal solution due to battery usage. On the Samsung Galaxy SIII, with Android 4.0 they have solved the problem of roaming in WiFi networks.

VI. CONCLUSION AND FUTURE WORK

It is a fact that the usage of mobile phones enables higher availability and accessibility, but also introduces a numerous of interruptions [5, 8]. This often leads to user resistance against wireless phones in clinical settings. Having this in mind we developed a context-sensitive system for mobile communication suited for hospital use, which provides the opportunity to control the availability, and thereby the interruptions [9]. The easiest solution is to introduce an already developed system, like the AwareMedia and the AwarePhone systems to Bardram et al. [24, 25]. This system is based on ordinary mobile phones using the GSM/3G network. A new hospital building up their infrastructure for mobile communication could make use of a solution like this, but we believe it is less expensive, and that the user resistance will be lower by utilizing an existing internal infrastructure. Also the idea of using already well known devices in clinical settings, made us look into what was possible to do with an infrastructure based on DECT phones and pagers [9]. From the laboratory experiments done by Gironi [9], with real users, the feedback was clear; the users wants a user interface more equal to conventional 3G/GSM mobile phones, which gave us the idea of including smartphones into CMS, using VoIP and SIP, resulting in CMS Smartphone.

Mandatory requirement for devices targeted for hospital usage is of course long battery life. To achieve this we had to balance the between computational power required by the software, audio quality perceived by the users, which is close related to the bandwidth required, and number of features introduced by the first version of the prototype of CMS SoftPhone. This was achieved by using OpenSL ES and G.711 audio codec to implement the audio recorder, which we optimized for low CPU usage on an acceptable low recording latency. Another reason choosing this was the compatibility of the Ascom system from [9]. By choosing this solution it seems like a Samsung Galaxy SIII with extended battery is able to last at least one normal communication intensive shift.

To count up for the unpredictability and variability of the network conditions, we used the most common solution and implemented an adaptive jitter buffer on the phone side of the system, and thereby keeping the buffering delay as short as possible and minimizing the number of packets that arrives too late. The implementation also takes in consideration the number of packet losses as well, and in combination this really shorten the delay between the caller and the called.

For low bandwidth networks like 3G and 2G, the G.711 codec is not suitable. This codec requires too much bandwidth, and was perceived as not suitable due to; either we had to deal with an increasing delay, or scattered sound losing a lot of audio packages. The solution was to use the Speex codec on the 2G and 3G networks in case of emergency communication, and to keep the G.711 codec on other wireless networks. Since the Speex codec is not supported by the Ascom system, we had to implement a solution where we were performing a transcoding on the media gateway in real time, when a communication channel is set up between the CMS SoftPhone and a DECT phone.

Another problem we had to face was the echo. When calling or receiving a call, we experienced a lot of echo, which was annoying and made the conversation difficult. We tried different approached, but discovered after the Android 4.0 was released that this version included a well working echo cancellation feature, and we concluded that the CMS SoftPhone has to rely on devices running minimum the 4.0 version of the Android operating system.

The tracking of the CMS SopftPhone was done by using Bluetooth adapters as sensors. This was not an optimal solution due to battery drainage, and unreliable tracking, and therefore we need to find a better solution. The solution that seams most reliable and accurate are an ultrasound solution, which requires an ultrasound tag on the phone and a microphone inside of each area we want to track the phones. This is planned tested in the next version of CMS.

Most of the Android phones we tested do not perform roaming within WiFi networks in a seamless way. This is a serious problem, and every device that should be used within CMS, have to be tested and approved able to roam between different WiFi antennas, in real time, without losing the connection or ending the call. This has been a known problem on earlier Android based devices, but after testing new devices, hi-end devices from Samsung and HTC, we found out that this problem is on its way to be solved, and the roaming is working very well on the Samsung Galaxy SIII.

Since both the echo cancellation and roaming within WiFi networks is solved, and that our implementation of the softphone is working just as well as on the DECT system, and since the smartphones has a wider area of usage, for instance to include patient information, medical reference work, etc., we conclude that the first version of CallMeSmart SoftPhone is ready to be tested in real life within health care settings. This also opens up for future work on including more features into CMS.

ACKNOWLEDGMENT

This research is supported by the Research Council of Norway, grant no. 176852/S10. We would like to thank Ascom AB all help so far in the project, and for loaning us the equipment for our Context lab.

REFERENCES

[1] E. Coiera and V. Tombs, "Communication behaviours in a hospital setting: an observational study," *BMJ,* vol. 316, pp. 673-676, February 28, 1998 1998.

[2] K. J. Ruskin, "Communication devices in the operating room," *Curr Opin Anaesthesiol,* vol. 19, pp. 655-9, Dec 2006.

[3] T. Solvoll, J. Scholl, and G. Hartvigsen, "Physicians interrupted by mobile devices in hospitals – understanding the interaction between devices, roles and duties," *unpublished*.

[4] M. A. Munoz, M. Rodriguez, J. Favela, A. I. Martinez-Garcia, and V. M. Gonzalez, "Context-aware mobile communication in hospitals," *Computer*, vol. 36, pp. 38-+, 2003.

[5] J. Scholl, P. Hasvold, E. Henriksen, and G. Ellingsen, "Managing Communication Availability and Interruptions: A Study of Mobile Communication in an Oncology Department," in *Pervasive Computing*, ed, 2007, pp. 234-250.

[6] R. G. Soto, L. F. Chu, J. M. Goldman, I. J. Rampil, and K. J. Ruskin, "Communication in critical care environments: mobile telephones improve patient care," *Anesth Analg*, vol. 102, pp. 535-41, Feb 2006.

[7] P. A. Spurck, M. L. Mohr, A. M. Seroka, and M. Stoner, "The impact of a wireless telecommunication system on time efficiency," *J Nurs Adm*, vol. 25(6), pp. 21-26, Jun 1995.

[8] T. Solvoll and J. Scholl, "Strategies to reduce interruptions from mobile communication systems in surgical wards," *Journal of Telemedicine and Telecare*, vol. 14, pp. 389-392, 2008.

[9] L. Gironi, "A prototype system for context sensitive communication in hospitals based on an Ascom/trixbox experimental platform" Master thesis; University of Tromsø, June 2011

[10] T. Solvoll, J. Scholl, and G. Hartvigsen, "Physicians interrupted by mobile devices – relations between devices, roles and duties," *Studies in Health Technology and Informatics*, vol. 160, p. 1365, 2010.

[11] D. Schuler and A. Namioka, *Participatory design: principles and practices*: L. Erlbaum Associates, 1993.

[12] J. Nielsen and R. L. Mack, *Usability inspection methods*. New York: John Wiley, 1994.

[13] M. Jones and G. Marsden, *Mobile interaction design*. Chichester: Wiley, 2006.

[14] T. Solvoll, "Mobile Communication in Hospitals: What is the Problem?," in *Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences*, ed: IGI Global, 2013, pp. 287-301.

[15] B. Boehm, "A spiral model of software development and enhancement," *SIGSOFT Softw. Eng. Notes*, vol. 11, pp. 14-24, 1986.

[16] T. Gilb, *Principles of software engineering management*: Addison-Wesley Longman Publishing Co., Inc., 1988.

[17] (Oct. 19). *MjSip*. Available: http://www.mjsip.org/

[18] (Oct. 19). *OpenSL ES*. Available: http://www.khronos.org/opensles/

[19] (Oct. 19). *OpenSL ES for Android*. Available: http://mobilepearls.com/labs/native-android-api/opensles/index.html

[20] *Android - An Open Handset Alliance Project*. Available: http://code.google.com/p/android/issues/detail?id=3434

[21] (Oct. 19). *VOICE_COMMUNICATION*. Available: http://developer.android.com/reference/android/media/MediaRecorder.AudioSource.html#VOICE_COMMUNICATION

[22] (Oct. 19). *Speex*. Available: http://www.speex.org/

[23] (Oct. 19). *iLBC*. Available: http://www.ilbcfreeware.org/

[24] J. E. Bardram and T. R. Hansen, "The AWARE architecture: supporting context-mediated social awareness in mobile cooperation," presented at the Proceedings of the 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois, USA, 2004.

[25] J. E. Bardram, T. R. Hansen, and M. Soegaard, "AwareMedia: a shared interactive display supporting social, temporal, and spatial awareness in surgery," presented at the Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, Banff, Alberta, Canada, 2006.