Supporting Pervasive Collaboration in Healthcare — An Activity-Driven Computing Infrastructure

Jakob E. Bardram and Henrik Bærbak Christensen
Centre for Pervasive Computing
Department of Computer Science, University of Aarhus
Aabogade 34, 8200 Århus N, Denmark
Tel.: +45 89 42 32 00
{bardram,hbc}@daimi.au.dk

ABSTRACT

Clinical work in modern hospitals is characterized by a high degree of mobility, frequent interruptions, and much ad hoc collaboration between colleagues with different expertise. Electronic patient record systems are typically build upon the classic client-server architectural style that has evolved to support office work that do not have these properties. In this paper, we describe clinical work and compare it with office work. This analysis leads to the conclusion that the mobile, interrupted, and ad hoc collaborative nature of clinical work is poorly supported by traditional client-server architectures and we propose an *activity-driven computing infrastructure* as an alternative. We outline our prototype design; argue how it supports clinical work and present initial results from testing the architecture in workshops with clinicians from Aarhus county hospital.

KEYWORDS: Pervasive Computing, Pervasive Healthcare, Mobile CSCW, Distributed Collaboration, Software Architecture.

INTRODUCTION

The issue of 'mobility' is gradually attracting interest in the field of CSCW, both with regard to field studies of mobile collaboration as well as technologies supporting mobile work. This paper reports from research done in a hospital setting where mobility is fundamental to the way clinicians work. Clinicians are on the move most of the time and only few have their own desk or personal computer. Their work is highly collaborative and often involves clinicians from several departments and even hospitals to treat a patient. We have termed this kind of work *pervasive collaboration* because it is highly nomadic and take place anywhere and at almost any time. It is a big challenge to create computer support for such work because it is fundamentally in contradiction to the kind of work that originally was the target for computer systems: clerical office work performed at a desk in front of a personal desktop computer. We present our proposal for supporting pervasive collaboration, which we call an *activity-driven computing infrastructure*. The core idea is to directly support the work activities that people are engaged in, and make this support pervasive within a hospital enabling them to start, suspend, resume and finalize work activities anywhere and at any time.

MOBILITY, DISTRIBUTED COLLABORATION AND COMPUTER SUPPORT

The research field of CSCW has for years been investigating collaborative work in real world settings and recently a few studies have been looking explicitly at how mobility and distributed collaboration is central to certain types of work. Moreover, in many case studies of cooperative work, mobility has been a core component of work but not explicitly dealt with (e.g. [17]). As described by Bellotti and Bly [4] people working in an office PC environment with desktop computers need to move around in order to cooperate. They need to move to shared resources not located in their personal office space and they need to move in order to communicate. The interesting part of this study is that even though we are in a typical office setting with desks, chairs and desktop computers, the classic desktop personal computer design leaves much to desire when it comes to supporting their work. A substantial part of their work was done in different locations than their offices, but the computing infrastructure made it difficult to "get away from the desktop" while preserving one's computing environment.

Much work, however, does not take place in an office and studies of these kinds of work reveal new challenges for computer support for mobile use. For example, studies of outdoor mobile work like ecologists observing giraffes' behavior in Kenya [14], of distributed telephone company workers [17], and of wastewater treatment workers [5] have shown that there is considerable challenges associated with designing computer systems for mobile and collaborative computer systems. This is partly due to the fact that often in mobile situations, users do not only perform their work using the computer; they do their work and then use the computer. For example, wastewater staff or ecologists make observations, manipulate knobs, cameras, etc., and—as a small part of their job—they also use computers.

At the same time there are considerable technological challenges to the design of mobile computer systems supporting cooperative work. Computer systems supporting cooperative work normally rely heavily on reliable high-bandwidth local area networks, extensive processing power, large displays, keyboards, and a reliable power supply. Few of these assumptions hold in a mobile environment. Furthermore, there is a long list of new challenges to the software architecture of such systems, some of them arising as a result of the new hardware challenges. Commercial products like SUN's Jini framework [11] or IBM's WebSphere Everyplace Suite [22] are designed to tackle some of these challenges, but they do not explicitly look at the cooperative aspects of mobile computer support. In the research literature there are a few suggestions for software architectures for mobile computing system that can be broadly classified into two directions. In one research direction, the focus is at supporting mobile work with mobile devices. For example Fagrell et al. [9] describe a classic client-server architecture for mobile, handheld clients like cellular phones and PDAs that is designed to support mobile knowledge management. The ConNexus is similarly a system for providing awareness information for mobile users using PDAs [20]. The other research direction looks at supporting mobile workers by providing access to computational devices where people are. For example the 'Intelligent Hospital' is a location- and context-aware application that can locate and contact a person where she/he might be [13]. This is used primarily for audio/video conferencing by forwarding the conference call to the location of the receiver. The conference call then follows the physician as she/he moves around. Another example in this second group is the 'Notification Collage' that is a real- time collaborative and public electronic bulletin board [10]. This can be used for publishing notes, enabling communication, etc. and is typically located in public spaces like in a meeting room. Such public available displays enable people to work while moving around.

This paper tries to look at mobile ad hoc collaboration at a hospital and presents a software architecture trying to support parts of mobile work. We are working in the line with the second research direction described above by deploying publicly available computer "all over" the hospital. A fundamental challenge not addressed by the systems described above is the preservation of users' working context while they are moving around. For instance, it is problematic if a physician needs to login, find the patient, and navigate to the medicine schema, etc. every time she moves from one public computer to another: as she is required to do this many times a day, a substantial amount of time is wasted simply on recreating her working context over and over again.

In the rest of the paper, we introduce the background for our project and our research methods. We summarize some of our main findings from our field studies of hospital work and from our workshops with hospital staff before we describe our *activity-driven computing infrastructure* (ADCI). Finally, we conclude with a description of our lessons learned so far and some of the future directions this research is planned to take.

BACKGROUND – PERVASIVE HEALTHCARE

One of the main application areas for pervasive computing in the Center for Pervasive Computing (CfPC) [6] in Denmark is *healthcare*. We have termed this research area "pervasive healthcare" [15]. We find that healthcare can be significantly improved by using pervasive computing technologies to create better patient treatment and to make better computer support for the work done by clinicians.

Research Methods

Within CfPC we conduct research in an experimental and multidisciplinary manner with a strong participation of industrial partners. In the projects concerning hospital work and electronic patient records we have a project team consisting of computer scientists with various backgrounds: CSCW, HCI, software architecture, and distributed computing; an ethnographer; and clinicians from Aarhus county hospital, including physicians and nurses. The industrial partner is one of the main suppliers of electronic patient records to the Danish hospital sector.

We use this multitude of project participants as basis for an experimental system development effort with agreed upon clinical, commercial, and scientific goals.

Our research methods include ethnographic observations of clinical work and use of computer technology. In our current project, an ethnographer has been making participant observations and interviews for months. The experimental systems development methods applied are scenarios-based design, future workshops, role-playing games, design workshops and evaluation experiments. Progress is made working through central clinical *themes*: (i) administration of medicine, (ii) prescription of medicine, (iii) clinical conference situations, and (iv) self-administration of medicine. Each theme goes through an iterative process consisting of three main workshops in which representatives for all partners participate:

- The *Vision Workshop* creates visions for how to support clinical work within a specific theme.
- The *Design Workshop* tests potential solutions through scenarios, role-plays, paper and cardboard mock-ups and discussions.
- The *Evaluation Workshop* tests one or more software-based prototypes based on experience from the design workshop.

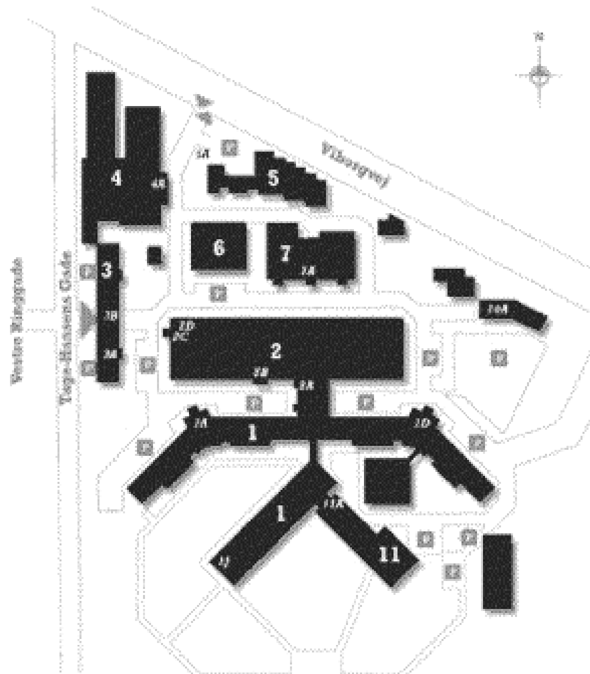

Figure 1: Map of the hospital of Aarhus county. Important buildings are: the main doctoral building (3), department B (1), the main treatment block (2), and (4) the canteen.

Department B at Aarhus County Hospital

This paper takes as its starting point our field studies at department B at the Aarhus county hospital (see figure 1). Department B is a hematology department for blood related diseases, typically leukemia. The department consists of three wards, an outpatient clinic, two laboratories, and a ward for incoming patients. The department can have 46 hospitalized patients and treats approximately 11.000 patients a year in the outpatient clinic. The main part of department B occupies all four floors in building 1. The department employs about 167 physicians, nurses, clinical assistances, laboratory workers, etc. The ground plan for a typical ward is illustrated in figure 2.

PERVASIVE COLLABORATION IN MODERN HOSPITALS

In this section we will outline some of our findings concerning the collaborative work taking place in a medical department at a modern Danish hospital. As stated in the introduction we have chosen to call it "pervasive collaboration" because it is characterized by taking place at any place inside and outside the hospital and at almost any time. Clinicians walk around the hospital premises and work closely together while moving. This section describes mobility in clinical work and some of the causes and consequences of mobility.

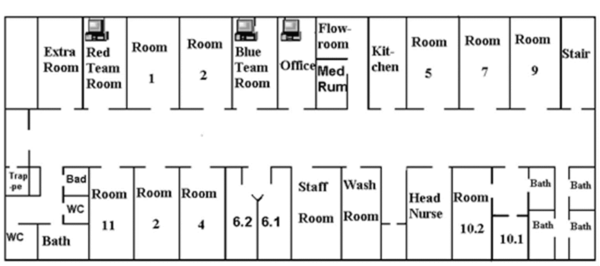

Figure 2: The ground plan of a typical ward at department B.

Figure 3: The mobile, analogue patient record.

Medical Work is Mobile and Nomadic

A fundamental characteristic of medical work in hospitals is that clinicians seldom sit down. Clinicians of all kind are constantly moving around within their "action-range". The action-range of nurses is typical the ward or the outpatient clinic, and the action-range of the physicians and other doctors is the hospital.

Consider a typical day for a physician, for example. He would start by attending the morning conference at the department's conference room located in the main doctoral building (building 3 in figure 1). This is the place for general conferences on issues related to the department as a whole. Then he would move across the parking lot and into the radiology department in the basement of the main block (building 2), attending the radiology conference. Then he would walk the staircase to the ward (building 1) that he is responsible for and starts the ward round.

The ward round is another fine example of mobility in medical work. Every morning a team of one physician and one or two nurses visits their patients at the ward. The ward rounds typically start in the team room (see figure 2). While seated the physician and the nurse(s) go through all the patients, read the paper-based medical patient records, and look over results from lab tests, etc. Afterwards, they take the records along together with other relevant materials (medical handbooks, medicine schemas, and small medical instruments) on a trolley (see figure 3), and visit each of the patients at their bedside. Thus, the medical team moves around the wards using the trolley to carry all the paper-based material.

Another central scenario of mobility on a ward is the nurse's task of giving medicine. When a patient has been prescribed medicine, it is the duty of the nurse to ensure that the medicine is given to the right patient at the right time, is given in the proper way, and that the process is properly documented. To do this she must first locate the medicine schema in which the prescription is written. This schema can be almost anywhere at the ward because it is a shared resource among many clinicians. When the schema is located the nurse takes it to the medicine room and starts preparing the medicine. The medicine room in many sections actually consists of several rooms, specialized for different types of medicine, like normal pills (the 'Med Room' in figure 2) and intravenous medicine (the 'Flow Room'). When done preparing the medicine she walks to the patient's bedside and hands out the medicine (the different 'Rooms'). If she carries the medicine schema with her she can documents the process right away. Otherwise she must make a note of it and later locate the schema and document it then. This task is repeated for practically all patients four times a day. On a typical day the nurses would repeat this scenario about 40 times.

A third central mobility scenario concerns the physician on duty. The physician on duty is often responsible for a whole department, including the outpatient clinic and various wards that are scattered around the hospital area. Hence, when he is called he must move around to patients and fellow colleagues to consult them. Sometimes the physician on duty is at home and has to drive to the hospital in order to see a patient.

Cooperation and Specialization in Medical Work

This need for mobility is partly caused by the way a modern hospital is organized. Medical work within a hospital is highly cooperative as the treatment of a patient involves various clinicians in departments distributed throughout the hospital or even at other hospitals [1]. This is caused by the high degree of specialization within clinical work [21]. Therefore, clinical work relies heavily on communication between colleagues. Telephones, both mobile and stationary, and pagers are widely used to support communication, but they are clearly not always sufficient, as decisions often cannot be made without consulting the patient, viewing X-rays, or similar. Hence, clinicians need to move around to get in touch with each other and to visit the different departments they are cooperating with.

Sharing of Resources and Material

A fundamental way of coordinating the cooperative treatment of patients is by sharing material—most notable documents like records, charts, and medicine schemas, but also other shared resources, like medicine, the patient or a senior physician. Thus, clinicians need to move around to get access to shared resources. Bellotti and Bly [4] also observed mobility caused by shared resources. .

A critical shared document in medical treatment is the patient record. All hospital staff—nurses, physicians, secretaries, etc.—spends a considerable amount of time getting access to the patient record, which means finding and fetching it. As one might guess from the description above, another particular critical shared resource on a ward is the medicine schema. All the nurses involved in the medication of patients need to access the medicine schema and all physicians prescribing and discussion medication need to use the medicine schema as well. To complicate things the medicine schemas for all the patients at the ward are collected in a single red binder. Thus everybody at the ward who needs to consult a medicine schema is forced to locate this red binder.

Another good example of a shared material used for coordination purposes is different kind of whiteboards containing status overviews of some kind. For example, in the conference room at the ward there is a whiteboard inside the door containing a list of all patients. This board is used as a list of the tasks that are to be done during the day, as a status overview, and as a message board (see also [1] and [23] of the extensive use of whiteboards in hospitals). As whiteboard contents are not distributed, people need to walk up to them in order to view and use them. Persons are also shared critical resources—the patient as the core example. Clinical staff visits a hospitalized patient at his bedside and hence need to walk up to him. Similarly, clinicians need to go and visit other clinicians like the radiologist.

Medical Work is Ad Hoc and is Often Interrupted

The collaborative work among clinicians is most of the time done in an ad hoc manner. There are a few scheduled medical conferences: the morning- and the radiology conference; but the most predominant way of working is to establish ad hoc collaboration as the situations call for it. A typical situation is a patient calling the nurse because of pains, who then calls the physician on duty in order to have the physician prescribe some painkillers.

An unavoidable side effect of the distributed and close collaboration among clinicians is that they often interrupt each other. In order for a clinician to establish a collaboration or communication he must often interrupt the other person in whatever she or he is doing. In the example above, the nurse interrupts the physician who needs to suspend what he was doing when the phone rang, and start a dialog with the nurse about the patient and his condition. Often such a situation would require the physician to actually go and visit the patient, distracting him even more from his original work activity.

Another central example of interruption happens among nurses working in the medicine room. They frequently cooperate by asking each other question and by handing over the red binder with medicine schemas. Hence, there is a high degree of interruptions, but this is considered fruitful because it is a central part of a close cooperation. This kind of micro-mobility was also noted and described by Heath and Luff [12].

SUPPORTING PERVASIVE COLLABORATION

Computer systems designed to support cooperation in a hospital setting are typically build around the classic *client-server architecture* [3]. This architecture has proved to be a robust way of building distributed computer systems supporting distributed collaboration centered around a set of data objects that represent the objects central in the domain: patient, medicine, disease, etc. The characteristics of this architectural style is that the server is the component responsible of persistence, of modification and concurrent access of data, and of distributing data to clients.

In an electronic patient record (EPR) system, for example, objects model patients, deceases, staff, medicine, clinical observations, the patient record, etc. The strength of the client-server model is the distribution and thereby sharing of critical data necessary for doing a job. For example, a physician can access records, medicine information etc. on a patient from any computer thereby avoid the annoying job of walking around to find the medical record and/or the red binder. However, in a classic client-server EPR, walking up to a computer to consult the medicine schema for a certain patient, is still tedious: the nurse must provide user name and password to login (potentially in several applications) and then navigate the user interface to fetch data for the right patient, the right type of data like medicine schema, and right day and time. The login and navigational effort is required because the data inherent the user interaction is not part of the data stored on the server and must therefore be reestablished manually every time the nurse accesses a computer. It is left as the client's responsibility to maintain state information about the interaction with the user, and to enable the manipulation of data objects on the server through different user sessions residing purely on the client. When the client session is terminated, information about this user interaction session is seldom made persistent or distributed.

The problem is that the "work" and the "activities" that are carried out are *not* equivalent to just the objects of the domain. For instance the EPR knows about a given patient and about his/her associated medicine, but the EPR does not know about a nurse's activities like "to pour medicine" or "to given and document medicine".

This classic interpretation of the client-server architecture has evolved in a typical office environment where users have personal computers and desks. In this setting, there is little need for distribution of the client-side user interaction session as there is typically one machine - one man. In a hospital, however, this is not the case. As described above, clinicians work while walking around, they are constantly interrupted and they need to hand over tasks and share them with other colleagues.

Of course, things can be made smarter and some technologies exist, like SunRays [19], to help overcome some of the problems described above. In the SunRay system, a user's applications are running on a central server and manipulated through thin clients, the SunRays. A user gets access to his/her applications by inserting a smart card into the Sunray's smart card reader. Thus the smart card both authenticates the user to the system (login) as well as reestablishes his/her session of open applications instantly, right down to the position of the mouse cursor. However, only one session exists per user at any time.

However, our main point is that there is a fundamental contradiction between the kind of nomadic, ad hoc, interrupted collaboration taking place in a work setting like the hospital and the classic and prevalent client-server architecture, made to support office workers having personal workstations and a desk. And we therefore need to consider, and design for, this contradiction.

DESIGN GOALS

Based on the above discussion we argue that the classic interpretation of the client-server architecture, where the server handles domain objects but not the processes that modifies them, contradicts the way clinicians work at a hospital. Hence, our design has been guided by the following basic premise:

> *Pervasive collaboration can only be supported by a computing infrastructure if and only if both the data and the work processes of handling data are available pervasively.*

We denote these work processes "activities", hence the name of the proposed computing infrastructure: *activity driven computing infrastructure*.

Based on our field studies and our workshops with the clinicians we have elaborated on this premise to formulate the following design goals for a computing infrastructure that supports clinical work. The goals can be seen as different aspects of the basic premise stated above: to support the clinicians' work processes anywhere and anytime.

Activity-Driven

The physical healthcare activity is what we want to support and we do that by mirroring it in a computational activity that becomes the computational "granule" offered by the infrastructure. For instance the clinical activity of "pouring medicine for patient Mr. Hanson" is supported directly by an equivalent computational activity that embody patient data, medicine data, time of day, the identity of the tray that contains the medicine, the identity of the nurse, how the nurse has arranged the medicine plan windows in the EPR system, etc. At any time a clinician may suspend an on-going computational activity to start a new one or resume one from his/her personal list of pending activities. Activities can be planned ahead to be initiated at a later time, they can be handed over to another person, or they can be shared to enable collaboration.

Public Computers

We are using the term *public computers* to denote the kind of computing devices that we envision must pervade the hospital. Devices range from handheld devices over laptop- and desktop computers to wall-sized computer screens. Devices are connected in a reliable high bandwidth network including Wireless LAN (IEEE 804.11b). Devices are public in the sense that any person, even patients and their relatives, may use any device not already in use.

Location- and Context Awareness

People and selected things wear sensors that pinpoint their location. This allows the infrastructure to make a person's work activities available at any public computer that is in his vicinity. Persons and things are also associated with contextual information besides location. For instance, the infrastructure keeps track of the location of a medicine tray but also the identity of the patient the tray is associated with, what kind of medicine it contains, and whether the medicine has been given to the patient.

Supporting Mobility and Nomadic Work

The user must be able to re-establish his/her work activities including their corresponding user-interaction sessions anywhere and at any devices. Hence, a user can move seamlessly between different devices scattered around the hospital and the user's activities just "follow" along.

Supporting Interruptions

The infrastructure must support that any user's activity can be interrupted at any time. When interrupted, an activity is simply suspended until it is resumed later, keeping all its state intact.

Thereby it is easy to take turns in using a device: a device may serve one user's activity, be momentarily handed over for use by another person where after the first person can resume his/her activity seamlessly.

Supporting Collaboration

The distributed ad hoc collaboration among clinicians is to be supported in several ways. First through *sharing* activities: several people must be able to participate in a single activity. Second through encouraging *co-located collaboration* by making it easy to have several people use the same computer at the same time by alternating between their activities. Third to allow people to *send* activities to each other and support that two users look at the same thing even though they are distributed in time and/or space.

Non-intrusive User Interface

Activities follow the person owing them and is available on request. The "on request" aspect is important and taken very seriously in our project—this became evident at one of the workshops where we proposed simple automatic execution of activities which caused many objections. Thus, we have designed a small user interface from which the user can activate his/her activities. The user interface gives access to activities for all persons in the computers vicinity but in a way that does not disturb in case a person is already using the computer. Our present proposal, an *activity-bar*, is shown in figure 5 and will be discussed below.

AN ACTIVITY-DRIVEN INFRASTRUCTURE

In this section we sketch our activity-driven computing infrastructure. A more detailed account is provided elsewhere [8, 2].

A logical view diagram of the ADCI is illustrated in figure 4. The infrastructure consists of four, loosely coupled, subsystems.

Location- and Context Awareness Subsystem

Our ADCI is based on its ability to track the location of people and things. People and selected things must wear identity sensors or tags that can be identified by tracking sensors deployed at important locations: the medicine cupboard, at bedsides, in the team room, etc.

The responsibility of this subsystem is to retrieve location information and manage context information for other subsystems. Location information is fed into the subsystem by tracking monitors like RFID (radio frequency identity) tag-scanners, WLAN monitors, etc. Thus the location server provides location data in a technology independent fashion. Still, the abstraction level of events is low in the location server—typically in terms of hardware IDs. The context server serves two purposes. It adds semantics to the low level events of the location server based on context data; for instance the context database describes what RF tag any given person wears and the room RF tag scanners are located in. The context server can therefore map an event about a tag scanned at a scanner into an event about a person moving to a new location. Secondly, the context server also draws on other sources of contextual information, notably the EPR server. The subsystem both serves as a database of context information to query by other subsystems as well as actively notifies the activity management subsystem about events in the environment like e.g. persons moving.

Activity Management Subsystem

Central to our proposal is the activity concept. In the ADCI activity objects mirror actual work activities in health care that involve access to computational objects, like patient records, medicine handbook, work plans, etc. Computational activities embody references to viewed objects in the EPR- and other medical computer systems, as well the state of the user session: graphical user interface appearance, input focus state, etc.

Activities are basically either *resumed* or *paused*. When resumed, the user is actively engaged in the activity—like for instance nurse Berg pouring medicine for patient Mr. Hanson. In this situation the activity object will embody showing the medicine schema for Mr. Hanson and potentially other relevant information. When paused the activity is simply

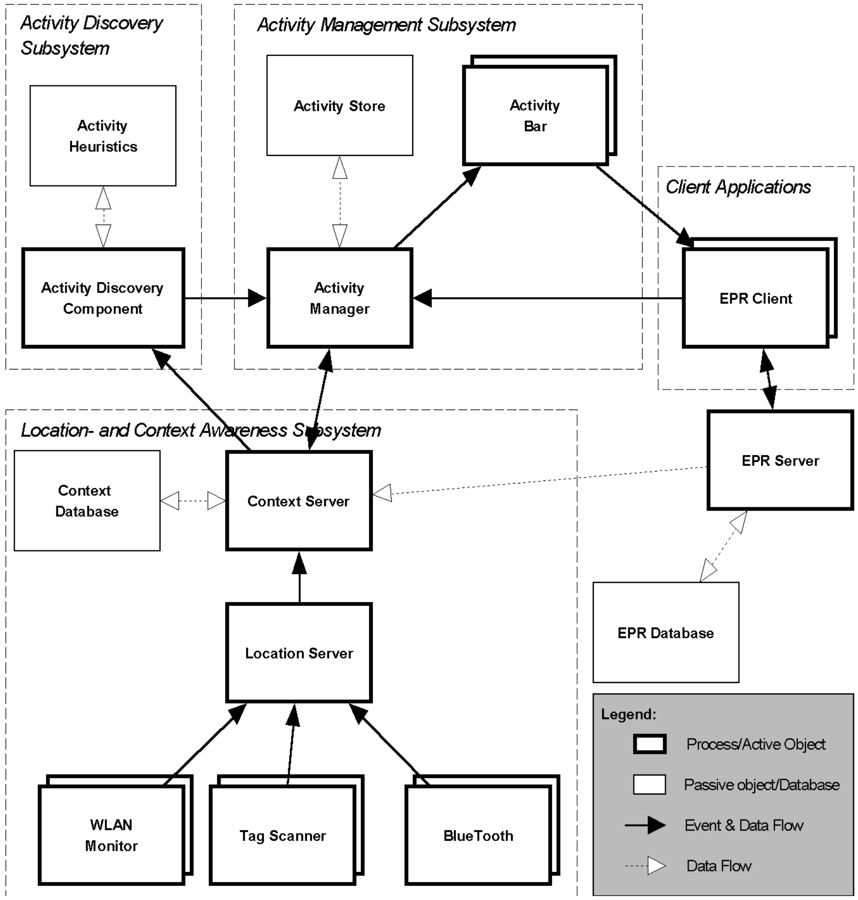

Figure 4: Logical view of architecture.

stored in the activity store for later resuming.

The responsibility of the activity management subsystem is to store and manage activities, forward them to activity bars on relevant public computers, and present them in a non-intrusive way to end users. It draws upon context data in order to determine which public computers any given activity should be forwarded to. Thus if nurse Berg moves to the bed of Mr. Hanson our activity management subsystem will forward all Berg's activities, including the activity concerning Hanson medicine schema, to the bed computer. The activity manager and the activity store run on a server computer while the activity bar applications run on the individual public computers.

Our current proposal for a non-intrusive user interface on which to present activities is an activity bar as shown in figure 5. The figure shows two snapshots. In the left pane a) the activity bar shows the presence of three clinicians in the vicinity of this computer. In the right pane b) clinician JSK has touched his image icon to bring up a menu of new and paused activities.

In our present design, activities are value objects that are forwarded to the client applications once they are resumed. In their paused state they always reside in the activity store. Thus, when a clinician approaches a computer, the activity manager only forwards 'handles' to his/her activities. Once an activity is selected for resuming, the activity is moved to the client application. This ensures that once an activity has been resumed it can be continued on the client even though the connection to the activity manager is lost.

Activity Discovery Subsystem

The responsibility of this subsystem is to proactively infer likely activities going on in the environment based on con-

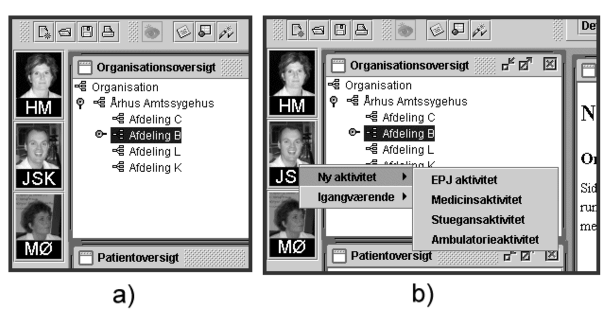

Figure 5: Present activity bar proposal showing icons for three clinicians in the vicinity.

text information and heuristics on recurring work processes in healthcare. Plainly speaking, it tries to guess what is going on and present the guesses to the clinicians. As an example if a nurse places a medicine tray next to the bed of the patient that the tray is assigned for, then in all likelihood she is about to give medicine to the patient and document it. The activity discovery subsystem infers this and autonomously creates an activity reflecting this event embodying the proper data for patient identity, nurse identity, feasible view of medicine plan, etc. The nurse can access these "discovered" activities from the activity-bar and save time. It is an optional subsystem that can be disabled. It is described in more detail in [7].

Client Applications

This subsystem covers third-party healthcare applications. In our project we have focused upon the EPR system where EPR clients may be run on the public computers. Client applications are responsible for storing on-going activities in the activity management subsystem, either when the user explicitly requests it to do so, or simply when the user leaves the public computer.

EVALUATION AND EXPERIMENTS

As described in the section on research methods we iterate between three kinds of workshops for each of the four themes in our project. The infrastructure described above is a result of iterations over theme one and theme two. At the time of writing we have conducted six workshops. The typical participants at the workshops are five clinicians (three doctors and two nurses), the development team (approximately four computer scientists), and an ethnographer responsible for the fieldwork in the hospital during our project.

Our present prototype does not support all design goals listed earlier. Our main focus has been on the feasibility of the activity concept and on supporting context-awareness, mobility, and interruptions. Currently, the prototype does not support collaboration in the sense of sharing and sending activities. Based on our design discussions with clinicians we do not, however, plan to support synchronous collaboration of the 'shared desktop' kind of style. The need for collaboration as expressed by the clinicians is to make it easy to shift between different work activities, where you work together with different people. Hence, the infrastructure should support the work of the individual, but in a way that you can collaborate with others simultaneously.

Prototype Specifications

For the evaluation workshops we have developed a prototype implementation of the ADCI. The prototype is mainly written in Java Standard Edition 1.3. Our prototype location-monitoring set-up consists of ICode tag-scanners and passive radio frequency identity tags (RFID-tags). These tags are cheap, weigh a few grams, are paper thin, and are easily glued onto a medicine tray or worn on a clinician's coat. Each RFID-tag has its own unique 64-bit identity. A *tag scanner* is able to detect the 64-bit identity of a tag whenever it enters the scanners detection area (about 0.5 meters) and also whenever it leaves the detection area again. Public computers are simulated by a number of laptop computers and a Smartboard connected in a LAN.

Lessons Learned

One major problem our research group faced was that presently the clinicians are using paper based medical records, as the EPR system has not been fielded yet. The prime advantage of the ADCI compared to a document-centered, client-server based, EPR system based on desktop computers is exactly that the work processes are very similar to their present processes. Thus the almost impossible challenge was to assess the feasibility of the ADCI compared to a EPR system that the clinicians had no experience with.

Easy and fast EPR access. The ADCI's activity forwarding that enables fast EPR access just by approaching a computer was highly appreciated. The project team has calculated that a nurse or a doctor will typically have to access EPR at least 20–30 times a day in many different places in the hospital thus manual authentication will be very time-consuming.

Mobility. In our evaluation workshops we showed how activities "follow the clinicians" to any public computer in their vicinity. The clinicians liked this idea and found it quite similar to their present, paper based, practice whose main benefit is indeed mobility. However, it of course offered the advantage that there was no need to run around in search for the red binder to get access to medicine plans: any public computer provided instant access.

Interruptions. During the workshop we played out several 'interruption' scenarios, where users where interrupted by other users, the telephone or just needed to do something else. The ability to hand over the public computer to another person and just resuming ones work later (sometimes another place) was clearly appreciated. In the scenario where the users are interrupted by themselves (or by the telephone) it was not always obvious how one would create a new activity for the interruption, in order to be able to return to the old one later. Instead of create a new activity when interrupted, the user would continue in the same activity. Hence, the user had no way of restoring the 'old' activity and get back to where things were when he was interrupted.

Activity bar. The idea of having an activity bar was judged to be essential and a good way of providing assistance without imposing things on the users that may not be relevant in their current use context. It also became clear, however, that our present activity bar proposal poses a severe security problem as it is easy for one clinician to get access to another clinician's activities as he/she walks by as they appear on the activity bar immediately.

DISCUSSION

The evaluation workshops have given us confidence that our ideas, centered on supporting activities pervasively on public computers, are a step in the right direction.

Our proposal is deeply grounded in observations of current work practices and processes on medical hospital departments. The present paper-based medical record practice has obvious advantages compared to traditional client-server based EPR system when it comes to usability and mobility. Traditional EPR systems have their advantage in the distribution of data and hence easy data access. In our opinion, our proposal tries to combine the best of both worlds.

Our ADCI is build upon the premise to understand what activities are, and aid in managing, storing, and distributing activities for the benefit of mobility, usability and easy data access for the clinicians. Furthermore, the set of pending activities serves as kind of a “to do” list for the clinicians.

By giving clinicians easy and fast access to their personal list of pending activities on any public computer we achieve the *mobility and nomadic quality*. As soon as any clinician is in the vicinity of a computer, his/her icon appears which provides access to the activity list. This way cumbersome authentication sequences and user interface navigation to fetch the right patient’s data, moving windows to get a proper view, etc. is avoided.

The *interruption support* quality is achieved as the ADCI supports interrupting any activity at any time and later resuming it in the exact same state as when left. The infrastructure supports having several activities pending for later activation and therefore aids the clinicians to remember unfinished activities during long interruptions.

Of course, a lot of important issues remain open at this early stage of our project. Some have already been touched upon. *Security* in data- and activity access is important and our present proposal relying only on nearness to public computers is obviously too naive: a nurse using a public computer may gain higher security level data access if a doctor passes by and he/she pushes the doctor’s icon.

Another point that needs clarification is the exact *semantics of an activity*: presently we use a inheritance hierarchy for activities that couple a given activity closely to a certain work situation, like an activity modeling “pouring of medicine” that embody patient data, tray identity, medicine schema view state, etc. This activity is next followed by the activity of giving the medicine to the patient and documenting the fact. Thus, the original activity is the obvious ‘template’ for the following activity as it is almost the same data and views that are needed. This leads to the conclusion that a more flexible and dynamic way of classifying activities is necessary.

Another aspect is how to determine when a user changes focus from one on-going activity to a new one? Must the user indicate this explicitly; or can the infrastructure infer this from context-changes in the usage behavior like e.g. change of applications or moving to “unrelated” data? While the latter may enhance usability it also comes with the danger of many false guesses that actually lower usability and may become a nuisance.

A difficult topic that we have not dealt with is how to migrate an activity’s embedded user interface state between devices with greatly varying screen characteristics. It is an obvious point for a separate research initiative that can draw upon existing work, like e.g. the iCrafter framework [16].

CONCLUSION

In this paper we have presented some characteristics of collaboration in clinical work in a modern Danish hospital. Based on these insights we presented our design of a computing infrastructure, which we argued would support clinical work. During our evaluation workshops we were encouraged in this perception.

Our studies of clinical work in a hospital made us describe it as ‘pervasive collaboration’, i.e. as cooperative work taking place at any time and any place—within the hospital and outside of it. The work is highly mobile and nomadic because of the specialized nature of modern medical work and because of the need to get access to shared material. This collaboration often takes place in an ad hoc manner, which leads to frequent interruptions of people.

There are substantial design challenges associated with designing computer systems for such an environment and up till now there is no record of successful clinical applications, in Denmark at least. We have argued that the prevalent way that information systems are created, based on a client-server model, actually contradicts the way work is achieved in such a hospital setting.

Based on the two insights above we have suggested the *activity-driven computing infrastructure* as a brick in the puzzle of supporting such heterogeneous work practices. The basic ideas is to have user sessions in terms of *activities* that moves along with the user and enables him/her to resume work on any public available workstation. This simple design seems to capture many of the requirements for supporting ‘pervasive collaboration’ in a hospital environment.

The activity-driven computing infrastructure is still work in progress and currently we are looking at improving it, addressing some of the issues discussed above, like security and usability. We would however argue that our design and implementation of the ADCI seems to be applicable for other computer systems besides clinical ones. The basic premises for the computing infrastructure, having work characterized as highly cooperative, ad hoc, mobile and nomadic, and associated with a high degree of unprioritized interruptions, is a description that also fits other kind of real world work-settings.

ACKNOWLEDGMENTS

The Danish National Centre for IT Research (CIT) funded this research. Claus Bossen and Anders K. Olsen were part of the team designing the proposals and designs presented within this paper and Anders K. Olsen and Hans-Henrik Olesen was involved in the implementation. We like to thank the staff from Aarhus county hospital for our enjoyable collaboration in the pervasive healthcare project.

REFERENCES

1. J. E. Bardram. Temporal Coordination. *Computer Supported Cooperative Work*, 9:157–187, 2000.

2. J. E. Bardram, H. B. Christensen, and A. K. Olsen. Activity-Driven Computing Infrastructure — Pervasive Computing in Healthcare. Submitted to "Pervasive 2002".

3. L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley, 1998.

4. V. Bellotti and S. Bly. Walking Away from the Desktop Computer: Distributed Collaboration and Mobility in a Product Design Team. In *Proceedings of CSCW 1996*. ACM Press, 1996.

5. O. W. Bertelsen and C. Nielsen. Augmented Reality as a Design Tool for Mobile Interfaces. In *Symposium on Designing Interactive Systems*, pages 185–192. ACM, 2000.

6. Center for Pervasive Computing. www.pervasive.dk.

7. H. B. Christensen. Using Logic Programming to Detect Activities in Pervasive Healthcare. Submitted to "International Conference on Logic Programming ICLP 2002".

8. H. B. Christensen, J. Bardram, and S. Dittmer. Theme One: Administration and Documentation of Medicine. Report and Evaluation. Technical Report TR-3, Center for Pervasive Computing, Aarhus, Denmark, 2001. Available at http://www.healthcare.pervasive.dk/.

9. H. Fagrell, K. Forsberg, and J. Sanneblad. FieldWise: A Mobile Knowlegde Management Architecture. In *Proceedings of CSCW 2000*. ACM Press, 2000.

10. S. Greenberg and M. Rounding. The Notification Collage: Posting Information to Public and Personal Displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems* [18], pages 514–521.

11. Jini network technology. www.sun.com/jini.

12. P. Luff and C. Heath. Mobility in Collaboration. In *Proceding of CSCW 1998*. ACM Press, 1998.

13. S. Mitchell, M. D. Spiteri, J. Bates, and C. Coulouris. Context-Aware Multimedia Computing in the Intelli. In *9th ACM SIGOPS European Workshop*. http://www.diku.dk/users/eric/ew2000/, 2000.

14. J. Pascoe, N. Ryan, and D. Morse. Using While Moving: HCI Issues in Fieldwork Environments. *ACM Transactions on Computer-Human Interaction*, 7(3):417–437, sep 2000.

15. Pervasive Healthcare. www.healthcare.pervasive.dk.

16. S. R. Ponnekanti, B. Lee, A. Fox, P. Hanrahan, and T. Winograd. iCrafter: A Service Framework for Ubiquitous Computing Environments. In G. D. Abowd, B. Brumitt, and S. Shafer, editors, *Proceedings of Ubicomp 2001: Ubiquitous Computing*, volume 2201 of *Lecture Notes in Computer Science*, pages 49–56, Atlanta, Georgia, USA, sep 2001. Springer Verlag.

17. P. Sachs. Transforming Work: Collaboration, Learning and Design. *Communications of the ACM*, 38(9):36–44, 1995.

18. SIGCHI. *Proceedings of the SIGCHI conference on Human factors in computing systems*, Seattle, USA, mar 2001. ACM.

19. Using Smart Cards With the Sun Ray 1 Enterprise Appliance. Whitepaper from Sun MicroSystems, available from www.sun.com, sep 1999.

20. J. C. Tang, N. Yankelovich, J. B. Begole, M. V. Kleek, F. Li, and J. Bhalodia. ConNexus to Awarenex: Extending awareness to mobile users. In *Proceedings of the SIGCHI conference on Human factors in computing systems* [18].

21. S. Vallgårda. *Sygehuse og sygehuspolitik i Danmark: Et bidrag til det specialiserede sygehusvæsens historie 1930–1987 [Hospitals and hospital Politics in Denmark: A Contribution to the history of the specialised hospital section 1930–1987]*. Jurist- og Økonomforbundets Forlag, Copenhagen, 1992.

22. IBM WebShpere Everyplace Suite. http://www-3.ibm.com/pvc/products/wes/index.shtml, 2000.

23. Y. Xiao, C. Lasome, J. Moss, C. F. Mackenzie, and S. Faraj. Cognitive Properties of a Whiteboard: A Case Study in a Trauma Centre. In W. Prinz, M. Jarke, Y. Rogers, K. Schmidt, and V. Wulf, editors, *Proceedings fo the Seventh European Conference on Computer Supported Cooperative Work*, pages 259–278. Kluwer Academic Publishers, 2001.