Activity-Driven Computing Infrastructure – Pervasive Computing in Healthcare

Jakob E. Bardram, Henrik Bærbak Christensen, and Anders K. Olsen

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,ako}@daimi.au.dk

Abstract. In many work settings, and especially in healthcare, work is distributed among many cooperating actors, who are constantly moving around and are frequently interrupted. In line with other researchers, we use the term *pervasive computing* to describe a computing infrastructure that supports work where users access a dynamic range of computing and software devices, where users can shift between devices, and where users move around while preserving their working environment. This paper describes our design of a pervasive *activity-driven computing infrastructure*. The main tenet in this approach is to preserve a user's computational working context enabling him to shift between different devices while on the move and enabling him to interrupt and return to work-activities fluently. Furthermore, the infrastructure contains agents that propose new activities based on the user's current context. This activity-driven computing infrastructure has been designed and developed in close cooperation with a large Danish hospital and is being evaluated in a hospital setting.

1 Introduction

Within research and industry R&D there is a growing interest in addressing the challenge of supporting people working together in environments that differ from the classical office user sitting at a desk with a Personal Computer or workstation. Interesting ethnographical studies of nomadic and mobile users has been made [20, 7] and several technical solutions for supporting this kind of work have been suggested (e.g. [17]). This kind of support has been termed 'Pervasive Computing' by enabling users to view and access data on server using mobile phones or PDAs [9] or by having services on mobile devices hook into a more general computing environment to perform tasks [2].

At the Centre for Pervasive Computing [12] in Denmark we are creating a computing infrastructure for pervasive computing. This work is partly grounded in ethnographic studies of hospital work and is carried out in close cooperation with medical staff [3]. Medical work within a modern hospital poses several challenges to a computing infrastructure. First of all, work is *cooperative*. The treatment of a patient involves many specialized professions, like physicians, nurses, radiologists, laboratory assistants, etc. Secondly, these persons are *distributed* across departments, physical spaces, different hospitals and even organizations. Thirdly, clinicians *move around* a lot; they seldom sit down and they do not have an office, a desk, or a computer. Most of their work is done while 'on-the-move' using cell phones, pagers, and dictaphones. Finally, clinicians are constantly *interrupted*, caused by the high degree of ad-hoc cooperation—physicians consult each other and specialists, nurses consult other nurses and doctors, patients contact nurses, etc. This means that clinicians constantly have to suspend and later resume many of their activities.

Our field studies have shown us that today's computing infrastructure for *electronic patient medical records* (EPR) do not support this kind of work very well. The classic client-server model with centralized servers and fixed workstations as clients fails in several ways. To name a few of the central problems, it makes no sense to have *personal* computers in a hospital environment: work cannot be taken to the computer—the computer must be available where the work is being done. Freely accessible computers, however, pose a data security problem and authentication must be enforced to grant access to the EPR system. This usually demands a tedious and time-consuming login procedure. Furthermore, the client-server model implicitly makes the assumption that only data is saved on the server but *not* the user interaction session on the client side such as layout of windows, what patient data is viewed, magnification factor, etc. Therefore clinicians waste time as they need to re-establish their working environment on each new client machine they access.

In this paper we describe a novel computing infrastructure for pervasive computing that we denote the *activity-driven computing infrastructure (ADCI)*. We argue that it addresses some of the problems and limitations mentioned and present initial evaluations. Our basic point of view is to provide an *activity-centred* perspective on computing in which the *individual work activity* is the basic unit of computational processing supported. Examples of activities in healthcare is giving medicine, enrolling a patient, prescribing medicine, conferring with a specialist on a certain treatment, viewing and examine lab results for a patient, etc. Physical activities like these are mirrored in the ADCI as "computational activities". The end users are directly supported by computational activities: activities can be initiated, interrupted and resumed in another point in time and/or space, stored so that the activity can be continued later, handed over to other persons, put on an individual's or group's "to-do" list, or shared among several persons. The activity embodies *all* relevant data: the underlying medical record data as well as all user interface specific data.

The paper first presents the empirical backdrop of this paper, our work of creating a pervasive computing infrastructure within a healthcare setting. We have termed this empirical case 'Pervasive Healthcare' [27]. We also present some of the central scenarios that drive the development of our infrastructure. Then our ADCI is described in detail. We then describe our experiments and evaluation of our infrastructure before we finally discuss related work and our future work.

2 Background—Pervasive Healthcare

One of the main application areas for pervasive computing in the Centre for Pervasive Computing (CfPC) in Denmark is *healthcare*. In Denmark there has been a very low adoption of electronic patient medical records (EPR) systems. We think this is caused by the problems of cooperation, distribution, mobility, and interruptions outlined in the introduction and therefore focus on supporting the clinicians' work. We think that pervasive computing technologies can improve their use of computer systems compared to the use of computers today.

2.1 Research Methods

Within the CfPC we conduct research in an *experimental* and *multidisciplinary* manner with a strong participation of industrial partners. We have a project team consisting of computer scientists with various backgrounds (software architecture, distributed computing, HCI, and CSCW), an ethnographer, and clinicians from Aarhus county hospital. The industrial partner is one of the main suppliers of electronic patient records (EPR) for Danish hospitals.

We use this multitude of project participants as basis for an experimental system development effort with agreed upon clinical, commercial, and scientific goals for the project. The main idea is to apply, develop and evaluate new types of pervasive computer support in a healthcare setting.

Our research methods include ethnographic observations of clinical work and use of computer technology. Experimental systems development methods applied is scenarios-based design, future workshops, role-playing games, design workshops and evaluation experiments. Progress is made by working with different central clinical *themes*. In theme 1 the subject is *administration of medicine*, theme 2 deals with *prescription of medicine*, theme 3 is *clinical conference situations* and theme 4 concerns *the "pervasive" patient*. Presently, the first theme has been concluded, while the second is ongoing. 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 possible solutions through scenarios, role-plays, paper and cardboard mock-ups and discussions.
– *The Evaluative Workshop* tests one or more software-based prototypes based on experience from the design workshop.

3 Design Premises and Goals

The ADCI design is based upon a few premises that have been formulated in collaboration with the clinicians during the vision- and design workshops. In this section we describe the premises and their motivation and within this framework we define the design goals.

3.1 Premises

The primary premise of the ADCI has been to support clinicians in their everyday healthcare activities in the best possible way. We have primarily focused upon the interaction with the EPR system though other medical systems are also possible. The (important) issues of data consistency, security and availability are left for the EPR system to consider.

Our premises are:

- *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" of the infrastructure. For instance the physical activity of pouring medicine for patient Mr Hanson is supported directly by the `PourMedicinActivity` that embody patient data, medicine data, time of day, the identity of the tray that contains the medicine, how to present the medicine plan in the EPR system, etc.
- *Public computers:* We are using the term *public computer* computing devices that 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. 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 locations. This allows the infrastructure to make a person's 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.
- *Non-intrusive user interface:* Activities must be presented to users in a non-intrusive way on a public computer. There are several reasons for this. Firstly, the computer may already be in use; consider the case in which a nurse is viewing a medicine plan and a doctor happens to pass by—she does not want to be disturbed by activities of the doctor. Secondly, a person may not want to enact any activity, for instance if a nurse passes five computers on her way to the nurses' office she has no intention of using any of the computers. Thirdly, any person may have many activities pending and the infrastructure cannot decide which to enact, it must be the decision of the user himself.
- *Propositional:* Activities are *proposed* to their user—never enacted by the infrastructure itself. Enacting an activity may change vital medical record data and therefore the enactment itself must be done based on medical judgement. It is not the purpose of the ADCI to engage in medical decision-making. Activities are proposals that the clinicians may take advantage of or simply ignore.

3.2 Design Goals

Our design premises provides the foundation for pursuing the following design goals:

- Supporting *mobility and nomadic work* by keeping track of the user's activities and context, making users able to re-establish it anywhere and at any devices. Hence, a user can move seamlessly between different devices scattered around the hospital and his activity-specific set-up just "follows" him around.

– Supporting *interruptions* of a user's activities by making it 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 activity seamlessly.
– Supporting *collaboration* through sharing activities and contexts, having several people using the same computer at the same time, or participating in an activity distributed in time and/or space.
– Supporting *activity detection:* (proactive guidance) for users by letting the infrastructure "guess" activities based on a user's context and heuristics about recurring work processes. For example, if a nurse carries a medicine tray then the ADCI can propose the activity to review the medicine plan that details the contents of the tray for the patient associated with the tray.

So far, our project's focus has been on activities, mobility, interruptions, and discovery. Collaboration issues are dealt with in theme three.

4 Scenarios in Pervasive Healthcare

In this section we present some scenarios to illustrate how the ADCI supports clinicians in their daily healthcare activities.

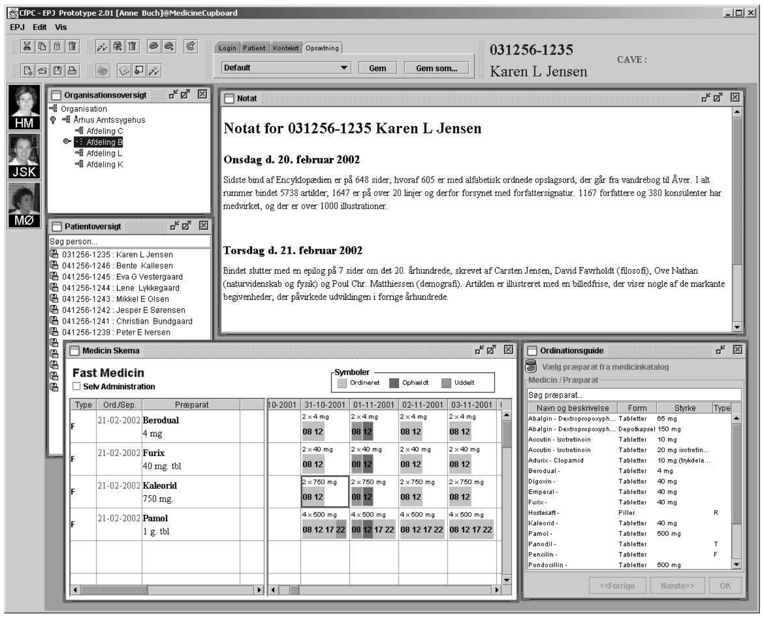
4.1 Scenario 1 – Interruption

Nurse Peterson needs to prepare medicine for patient Mr Hanson. She goes to the medicine room in which the medicine is located. When she approaches the computer in the medicine cupboard an icon showing her face appears on the computer's activity bar, similar to the Windows task-bar, and she is identified and granted access to the EPR system simply by touching this icon. Figure 1 is a snapshot of our prototype showing our present proposal for an activity bar integrated into the EPR system.

She picks up the medicine tray for Mr Hanson and as she places it on the table, a menu item appears next to her icon with the label "Prepare medicine for Mr Hanson". As she touches it, the computer shows the medicine schema for Mr Hanson, and she starts putting medicine in the tray. Suddenly nurse Berg enters the room with a tray in her hand. She asks Peterson if she can borrow the computer for a second to check the content of the tray. As Berg approaches an icon with her face appears on the activity bar. Touching her icon, a menu appears and she selects entry ”Show tray contents for Mr Carlson" and Carlsson's medicine schema is now displayed highlighting the content of the tray. Berg checks the schema and removes the tray again. As Berg leaves, the display returns to the state it was in before she interrupted Peterson. Peterson now finishes preparing medicine and moves out of the medicine room. As she does this, the medicine preparation activity is automatically suspended and the display blanks as she is logged out.

4.2 Scenario 2 – Mobility

Later in the day, nurse Peterson needs to show one of her colleagues the medicine that she has prepared for Mr Hanson. Approaching the computer in the nursing office, her



031256-1235
Karen L Jensen
CAVE :

031256-1235 : Karen L Jensen
041256-1246 : Bente Kallesen
041256-1245 : Eva G Vestergaard
041256-1244 : Lene Lykkegaard
041256-1243 : Mikkel E Olsen
041256-1242 : Jesper E Sørensen
041256-1241 : Christian Bundgaard
041256-1239 : Peter E Iversen

Notat for 031256-1235 Karen L Jensen

Onsdag d. 20. februar 2002

Sidste bind af Encyklopædien er på 648 sider, hvoraf 605 er med alfabetisk ordnede opslagsord, der går fra vandrebog til Åver. I alt rummer bindet 5738 artikler; 1647 er på over 20 linjer og derfor forsynet med forfattersignatur. 1167 forfattere og 380 konsulenter har medvirket, og der er over 1000 illustrationer.

Torsdag d. 21. februar 2002

Bindet slutter med en epilog på 7 sider om det 20. århundrede, skrevet af Carsten Jensen, David Favrholdt (filosofi), Ove Nathan (naturvidenskab og fysik) og Poul Chr. Matthiessen (demografi). Artiklen er illustreret med en billedfrise, der viser nogle af de markante begivenheder, der påvirkede udviklingen i forrige århundrede.

Fast Medicin

☐ Selv Administration

Symboler: ▨ Ordineret ■ Ophældt ▨ Uddelt

Type	Ord./Sep.	Præparat		10-2001	31-10-2001	01-11-2001	02-11-2001	03-11-2001
F	21-02-2002	**Berodual** 4 mg			2 × 4 mg 08 12	2 × 4 mg 08 12	2 × 4 mg 08 12	2 × 4 mg 08 12
F	21-02-2002	**Furix** 40 mg. tbl			2 × 40 mg 08 12	2 × 40 mg 08 12	2 × 40 mg 08 12	2 × 40 mg 08 12
F	21-02-2002	**Kaleorid** 750 mg.			2 × 750 mg 08 12	2 × 750 mg 08 12	2 × 750 mg 08 12	2 × 750 mg 08 12
F	21-02-2002	**Pamol** 1 g. tbl			4 × 500 mg 08 12 17 22	4 × 500 mg 08 12 17 22	4 × 500 mg 08 12 17 22	4 × 500 mg 08 12 17 22

Navn og beskrivelse	Form	Styrke	Type
Abalgin - Dextropropoxyph...	Tabletter	65 mg	
Abalgin - Dextropropoxyph...	Depotkapsel	150 mg	
Accutin - Isotretinoin	Tabletter	10 mg	
Accutin - Isotretinoin	Tabletter	20 mg isotretin...	
Adurix - Clopamid	Tabletter	10 mg (trykdele...	
Berodual -	Tabletter	4 mg	
Digoxin -	Tabletter	40 mg	
Emperal -	Tabletter	40 mg	
Furix -	Tabletter	40 mg	
Hostesaft -	Piller		R
Kaleorid -	Tabletter	40 mg	
Pamol -	Tabletter	500 mg	
Panodil -	Tabletter		T
Pencilin -	Tabletter		F
Pondocillin -	Tabletter	500 mg	

Fig. 1. Display snapshot of our present prototype. The activity bar is on the very left showing images of three clinicians in the vicinity.

icon appears again, and she can reactivate the suspended “Prepare medicine for Mr Hanson” activity by selecting it from the menu of suspended activities. As she does this, the display restores to the exact state as when she left the medicine room.

4.3 Scenario 3 – Activity Discovery

In the afternoon, Berg has to give the five o’clock medicine to Mr Hanson. She goes to the medicine room, picks up his tray and takes it with her. At Mr Hanson’s bedside she can use the public computer embedded in the bed’s table. As she places the tray on the bed table, the infrastructure guesses that Mr Hanson is about to given his medicine and creates a “discovered activity” that is able to record this event in the EPR system. The discovered activity shows in the menu that appears as Berg touches her icon on the bed’s display. She activates the activity, Mr Hanson’s medicine schema is displayed and the content of the tray is highlighted. She touches the “sign button” in the EPR to sign that the medicine has been given to Mr Hanson. All in all, touching the display twice is all that is required to document the giving of the medicine.

5 Activity-Driven Computing Infrastructure

A logical view diagram of the ADCI is illustrated in Fig. 2. The infrastructure consists of four, loosely coupled, subsystems.

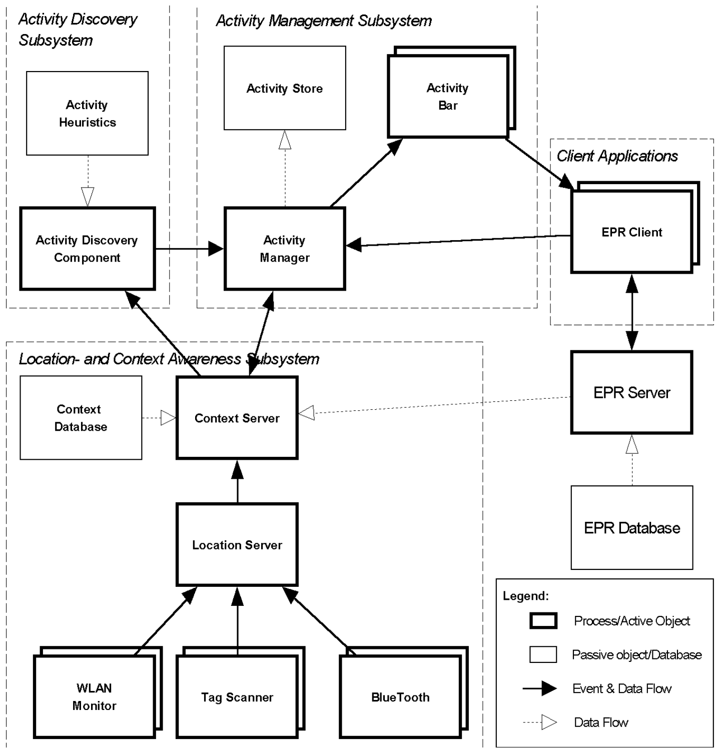


Fig. 2. Logical view of infrastructure, showing main runtime components and subsystems

- *Location- and Context Awareness Subsystem:* The responsibility of this subsystem is to manage context information for other subsystems. Location information is updated through tracking monitors like RFID tag-scanners, WLAN monitors, etc. that constantly feed the location server with data about location of entities. 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 persons moving.
- *Activity Management Subsystem:* The responsibility of this 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.
- *Activity Discovery Subsystem:* The responsibility of this subsystem is to proactively infer likely activities going on in the environment based on context 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 exemplified in the activity discovery scenario, section 4.3. It is an optional subsystem.

– *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.

The EPR clients access the EPR server that stores medical data in a traditional client-server architecture.

5.1 Location- and Context Awareness Subsystem

The main responsibility of this subsystem is to maintain context information about people, places and things and send notifications to other subsystems when contextual information changes.

We use the term 'context' in a broad and open-ended way. In the present prototype we are focused on (1) physical context, primarily the location of an entity, (2) computational context, primarily which applications a user is using, and (3) healthcare context, primarily associations between entities—like which patient a given medicine tray is used for, or the contents of a medicine tray. Of course, the present model can be extended with other types of context information. The subsystem can be viewed from two different perspectives.

From one perspective, the subsystem is a *pipe-line architecture* [6] in which low-level events are generated by the hardware-near components and sent to the location server. The location server processes the events and sends them to the context server. The context server processes the events even further, updates the context database and notifies all observers that have registered interest in context change notifications. As an example, the activity management subsystem is notified when a user is in near a public computer so it can forward his activities to it.

From another perspective, the subsystem is the server-side of a *client-server architecture* [6] in which other subsystems can query about context information. As an example, a subsystem can query the context server about which public computers are located in a given room.

The location server keeps track of low-level devices like RFID Tags, WLAN and Bluetooth devices, stationary computers and client applications running on the public computers.

The context server adds semantics to the data of the location server. It maintains a context database that describes relations between entities such as which RFID tag is worn by a person or glued to an artefact, the location of a given tag scanner, where public computers are located, etc. It combines this information with movement events to (a) update the context database and (b) to construct more semantically rich events. For instance if tag x is detected by tag-scanner A and the context database contains information that tag x is worn by nurse Berg and scanner A is located in the medicine room, it updates the location of Berg accordingly and notifies observers that Berg has moved to the medicine room.

5.2 Activity Management Subsystem

The responsibility of the activity management subsystem is to store, manage and present computational activities that mirror everyday healthcare activities.

In the logical view architecture diagram in Fig. 2 it is instances of *activities* that are sent between components. Before we describe the responsibilities of the individual components we outline our activity concept.

Activity Modelling. The activity concept is central to our infrastructure. We model activities in our system by defining the `Activity` interface as a generic template for known activities. An inheritance hierarchy of activity classes model concrete activities. The root class, `BaseActivity`, holds information about the creator, owner, participants and its state. In our design, a given activity instance can only exist in a single process at a time. This means that activity instances are basically value objects that are passed back and forth between client applications and the activity manager.

An activity is always in one of the following states:

– *Initialised.* The activity is newly created.
– *Resumed.* The user is actively engaged in the activity. Only in this state is the data embodied in the activity changed, like EPR data references, user interface layout information, etc.
– *Paused.* The activity has been interrupted and can be resumed/activated at a later time. Once an activity is suspended on a client it is moved to the activity manager that stores it for later retrieval and activation.
– *Terminated.* The activity has finished, and can no longer be activated but is still saved in the activity store for review.

It is only in the *initialised* and *resumed* states that the activity instance is present in the client applications. When it is in its paused and terminated state the instance is in the activity store. An activity is an instance of the *command pattern* [19] and modifies the state of the application is has been forwarded to when its `resume` method is invoked.

Other activity classes, like the `EPRActivity`, extends the `BaseActivity` by adding information and logic relevant to an EPR, like the patient involved, the user interface state of the EPR, and how to access the EPR Server. Likewise, the `DocumentMedicineGivenActivity`, add further information, like an identification of the tray in which the medicine is, and the identity of the clinician who poured the medicine. The basic idea is to allow for creation of new activities just by writing a new activity class extending the `BaseActivity`.

Components. The activity manager has several responsibilities. First of all it manages the activity store in which it stores and retrieves activities. Secondly, the manager takes care of sending activities to the right activity bars based on location information from the context server.

The responsibility of the activity bar is to provide users with non-intrusive access to their activities and to forward incoming activities to the right applications on the computer. In Fig. 3 our present design is shown. Part a) shows the activity bar when three persons are detected in the vicinity. Part b) shows a situation where JSK has clicked/touched his icon that brings up a hierarchical menu of his activities.

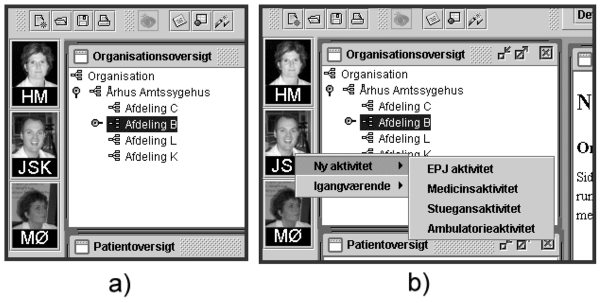


Fig. 3. A snapshot of present design of activity bar.

The activity manager runs on a server while activity bars run on the individual public computers as shown in Fig. 4 .

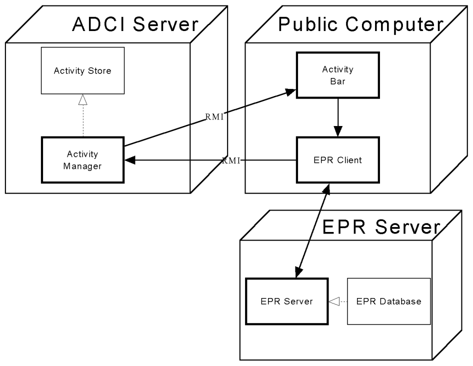


Fig. 4. Deployment diagram for activity management subsystem

The manager does not forward activity instances, rather it forwards *handles* to the activities to a given activity bar (recall that the activities are still in their suspended state and thus resides in the activity store). This is because the same activity can be relevant to send to several different public computers. Only when an activity is activated is the activity instance sent to the activity bar. Hence, it cannot be accessed from other computers or applications anymore. When an activity arrives it is forwarded to the proper client application that then invokes the `resume()` method responsible for setting the application's state appropriately.

5.3 Activity Discovery Subsystem

The responsibility of the activity discovery subsystem is to monitor the location of people and things and tries to infer likely activities in the healthcare environment based on rules and heuristics. This subsystem is thus an autonomous agent that proposes activities on behalf of users.

The subsystem consists of two components: a database of heuristics about recurring work processes and the *activity discovery component (ADC)* (see Fig. 2). The ADC surveys the information about people, places, and things that is available in the context server, combine this with heuristics to infer likely activities. A good example was outlined in the activity discovery scenario in section 4.3.

Discovered activities are proposals that clinicians may activate to significantly speed up recurring working processes—or safely ignore in case the ADC has not guessed the right activity.

The ADC is described in more detail elsewhere [13, 14].

5.4 Client Applications

The main client application in this pervasive healthcare project setting is of course the *electronic patient record* (EPR) system that was illustrated in Fig. 1.

It is not the purpose of this paper to describe the EPR, but basically the EPR is a framework for different data displays of data in a centralized server. This classic client-server architecture has been chosen to resemble the EPRs used in Danish hospitals. Thereby we can illustrate that our ADCI can be added to existing EPRs and thus enhance the existing systems without requiring fundamental re-implementation.

In Fig. 1 different views are shown, like an organizational overview, a list of patients, the medical record view, the medicine schema, and an ordination guide. This is a typical set-up for a physician when doing his ward round. The ADCI helps him to move this set-up around to all the different computers that he access, and to quickly shift between this one and another complicated set-ups.

6 Evaluation and Experimentation

As described in section 2.1 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 partly theme two. At the time of writing we have conducted five workshops. Thus the work is still in progress.

Our design ideas has been evaluated and further developed, and our prototype has been subject for experimentation. 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 project uses scenario-based design methods [10, 4]. These scenarios are used as a basis for role-playing in the experimentation workshop and the clinical personnel are asked to carry out everyday tasks using the new technology as described in the scenarios.

Below we discuss the technical set-up for the evaluation workshops and provide the lessons learned during the workshops.

6.1 Technical Set-up

For the evaluation workshops we have developed a prototype implementation of the ADCI. The prototype is mainly written in Java Standard Edition 1.3. The activity discovery component partly relies on the Jess [18] expert system that integrates seamlessly with Java.

Our 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.

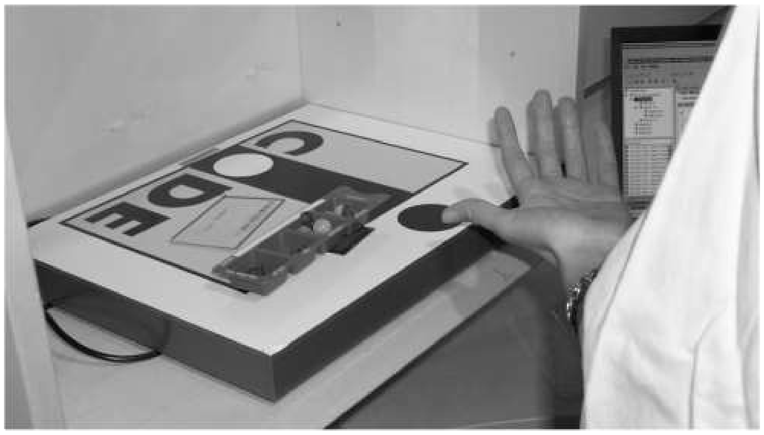


Fig. 5. Snapshot from the workshop showing our prototype RFID based set-up.

Public computers are simulated by a number of laptop computers. A snapshot from our evaluation workshop is seen in Fig. 5. To the left you see the ICode tag scanner and two tags on top of it. The upper one is taped onto cardboard and is the nurse's personal tag. Below is a medicine tray with a tag glued onto the bottom of it. On the right, partially covered by the nurse's back, is a laptop computer that displays activity bar and the EPR system.

6.2 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-centred, 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. The need to support clinical personnel moving around all the time is obvious. Our mobility support is based on the use of public computers and a set of computational activities that "follows the clinician" to any public computer in his vicinity. The clinicians liked this idea and found it quite similar to their present, paper based, practice whose main benefit is indeed mobility.

Interruptions. (Note for reviewers: the evaluation workshop in which this kind of support will be demonstrated is scheduled right after the submission deadline. If the paper is accepted we will describe the lessons learned.)

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.

Limitations in use of RFID-tags. Passive RFID tags are only detected within a 0.5-meter range of a tag-scanner. This range is too short to detect movement in general, for instance in the aisles. Therefore the location- and context awareness subsystem must utilise a range of different techniques and hardware devices in order to provide location information with sufficient detail. Our workshops showed, however, that passive RFID tags are fine for authentication. Tags could be worn in wristbands and a nurse could simply hold her hand near a scanner next to a public computer to access her activities.

7 Related Work

Our perception of context is inspired by CoolTown's notion of *people, places, and things* [11, 15]. Essential in healthcare is maintaining *relations* between these phenomena: nurses have responsiblity for a set of patients, patients are assigned to specific beds in wards, securing that a patient gets the right medicine tray is extremely important, etc. Thus many activities in healthcare can be described in terms of relations between people, places, and things.

Location- and context awareness has received a lot of attention in the research community. Context aware computing has been used for a wide range of domains such as guided tours [25], location- and context aware reminders [16, 22], and to facilitate automatic composition of dynamic devices [26]. In our approach location- and context awareness is primarily used for two purposes, namely to forward a person's activities to the computing devices in his vicinity, and to infer likely activities based upon proximity of people and things in combination with heuristics. Thus our ADCI combines several contributed ideas.

Similarities also exist to research in ubiquitous environments in meeting rooms and homes like for instance EasyLiving [8], iCrafter [28], and many others. Our project draws upon the experience in these projects—one of the main assets of our project is that the collaboration with hospitals provides a complex and realistic environment for testing various ideas. This changes the focus somewhat. Whereas most other projects focus on technical issues and evaluate them in artificial, and often rather simple contexts like the meeting room, our focus is more towards end user experiences and providing infrastructure to support human activities in a realistic environment with complex work processes and interactions.

An obvious comparison is to the SmartCard and SunRay technology from Sun Microsystems [24]. This client-server technology consists of thin clients and a SunRay server in which multiple users have their *sessions*. A session completely describes the computational state of the user's open applications right down to where the cursor is and is uniquely identified by the user's SmartCard. Thus a user can move to any SunRay client, insert his SmartCard, and instantly continue work on his session. Our ADCI differs in a number of respects. Our activity concept is akin the session and while a SmartCard only maintains a single (though complex) session our infrastructure presents a range of activities from which the user may choose one. To start a SunRay session you need to either use the SmartCard or perform a traditional login; in contrast our infrastructure forwards activities to a a device based on location awareness no matter how the location information has been obtained. Also our activities may be stored on to-do lists, triggered as reminders, or handed over to another person. Finally, the ADCI may autonomously create new activities for a person.

Our ADCI as described within this paper is based on a set of centralised services. Much research in pervasive computing aims at ending the rule of centralised, server-based, architectures and explores peer-to-peer computing and ad-hoc- and proximity based communication [21, 23]. It seems feasible that a more localized architecture will scale better in a large hospital setting like Aarhus county hospital. However, the present focus in our project is investing the benefits and liabilities of the activity-driven concept from an end user perspective. We hope to look further into the scalability issue later.

8 Conclusions

We have reported on work in progress on the ADCI where human activities are the granules that define the computational support offered. Our research has been within the domain of healthcare and our objective has been to support everyday activities in healthcare to augment patient record data quality and, in particular, to ease and speed up the use of EPR systems by clinicians. Based on location- and context awareness and pervasive computing hardware the infrastructure is able to offer personalized, computational, activities to clinicians anywhere and anytime. Activities can be defined by clinicians and suspended and resumed at will at any public computer in the hospital. The ADCI further proactively discovers and presents possible activities to users. Our design has been evaluated in workshops where prototype implementations are used by clinicians from the Aarhus county hospital.

Pervasive and ubiquitous computing is associated with “anywhere and anytime computing”. Bringing computing to us in our everyday endeavours will change the way we perceive computers. The shift from mainframes to desktop computers changed the view from an *application-centred* to a *document-centred* perspective. We think that pervasive computing will once again change the perspective to an *human activity centred* perspective where our activities decide what information is relevant, how to present it, and what combination of equipment to use to manipulate it. We therefore find that the presented ADCI has wider applications than just pervasive healthcare.

A lot of issues remain to be dealt with. We will not detail these here but just mention some key problems. Activities embody user interface state but this poses the problem of how to migrate it between devices with radically different characteristics: like suspending an activity on a wall screen and reactivating it on a PDA. We like the idea of virtual, composite, devices [26] and a hospital seems like an ideal setting for experiments; initial work has already begun in this area [5]. Collaboration is a key work process in clinical work and the issue of theme three in our project. Finally, an important issue is privacy as our infrastructure makes it possible to generate very detailed records of the staffs’ movements.

Acknowledgments

The Danish Centre of Information Technology (CIT) funded this research. Claus Bossen was part of the team designing the proposals and designs presented within this paper 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. G. Banavar, editor. *Advanced Topic Workshop—Middleware for Mobile Computing*, Heidelberg, Germany, Nov. 2001. http://www.cs.arizona.edu/mmc/Program.html.
2. G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman, and D. Zukowski. Challenges: An Application Model for Pervasive Computing. In *Proceeding of MOBICOM 2000*. ACM Press, 2000.
3. J. E. Bardram. Designing for the Dynamics of Cooperative Work Activities. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, Seattle, Washington, USA, 1998. ACM Press.
4. J. E. Bardram. Scenario-based Design of Cooperative Systems: Re-designing an Hospital Information System in Denmark. In *Group Decision and Negotiation*, volume 9, pages 237–250. Kluwer Academic Publishers, 2000.
5. J. E. Bardram and H. B. Christensen. Middleware for Pervasive Healthcare - A White Paper. In Banavar [1].

6. L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley, 1998.
7. 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.
8. B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. EasyLiving: Technologies for Intelligent Environments. In Thomas and Gellersen [29], pages 12–29.
9. J. Burkhardt, H. Henn, S. Hepper, K. Rintdorff, and T. Schäck. *Pervasive Computing - Technology and Architecture of Mobile Internet Applications*. Addison Wesley, 2002.
10. J. M. Caroll. *Scenario Based Design: Envisioning work and technology in system development*. John Wiley & Sons, Inc, 1995.
11. D. Caswell and P. Debaty. Creating Web Representations for Places. In Thomas and Gellersen [29], pages 114–126.
12. Center for Pervasive Computing. www.pervasive.dk, 2002.
13. H. B. Christensen. Using Logic Programming to Detect Activities in Pervasive Healthcare. Submitted to “International Conference on Logic Programming ICLP 2002”.
14. 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/.
15. Cooltown. http://www.cooltown.hp.com/.
16. A. K. Dey and G. D. Abowd. CybreMinder: A Context-Aware System for Supporting Reminders. In Thomas and Gellersen [29], pages 172–186.
17. H. Fagrell, K. Forsberg, and J. Sanneblad. FieldWise: A Mobile Knowlegde Management Architecture. In *Proceedings of CSCW 2000*. ACM Press, 2000.
18. E. Friedman-Hill. Jess, the Rule Engine for the Java Platform. http://herzberg.ca.sandia.gov/jess/.
19. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reuseable Object-Oriented Software*. Addison-Wesley, 1994.
20. P. L. . C. Heath. Mobility in Collaboration. In *Proceding of CSCW 1998*. ACM Press, 1998.
21. G. Kortuem. Proem: A Peer-to-Peer Computing Platform for Mobile Ad-hoc Networks. In Banavar [1].
22. N. Marmasse and C. Schmandt. Location-Aware Information Delivery with ComMotion. In Thomas and Gellersen [29], pages 157–171.
23. R. Meier, M.-O. Killijian, R. Cunningham, and V. Cahill. Towards Proximity Group Communication. In Banavar [1].
24. S. MicroSystems. Using Smart Cards With the Sun Ray 1 Enterprise Appliance. Whitepaper, available from www.sun.com, sep 1999.
25. R. Oppermann and M. Specht. A Context-Sensitive Nomadic Exhibition Guide. In Thomas and Gellersen [29], pages 128–142.
26. T.-L. Pham, G. Schneider, and S. Goose. Exploiting Location-Based Composite Devices to Support and Facilitate Situated Ubiquitous Computing. In Thomas and Gellersen [29], pages 143–156.
27. Pervasive Healthcare. www.healthcare.pervasive.dk.
28. 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.
29. P. Thomas and H. W. Gellersen, editors. *Proceedings of Handheld and Ubiquitous Computing*, volume 1927 of *Lecture Notes in Computer Science*, Bristol, UK, sep 2000. Springer Verlag.