Supporting Human Activities — Exploring Activity-Centered Computing

Henrik Bærbak Christensen and Jakob E. Bardram

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

Abstract. In this paper we explore an *activity-centered computing paradigm* that is aimed at supporting work processes that are radically different from the ones known from office work. Our main inspiration is healthcare work that is characterized by an extreme degree of mobility, many interruptions, ad-hoc collaboration based on shared material, and organized in terms of well-defined, recurring, work activities. We propose that this kind of work can be supported by a pervasive computing infrastructure together with domain-specific services, both designed from a perspective where work activities are first class objects. We also present an exploratory prototype design and first implementation and present some initial results from evaluations in a healthcare environment.

1 Introduction

The application-centered and document-centered computing paradigms have proved successful for programming in their respective domains: the application-centered paradigm fits large, centralized, business domains like banking, while the document-centered paradigm supports office-type work. It is not clear, however, that these paradigms are the proper ones for programming pervasive computing technologies.

In this paper we propose and explore an *activity-centered perspective* for modeling an important class of pervasive computing systems. Our main thesis is that the computing system must support handling human *work activities* directly; similar to how document-centered systems support handling documents directly. By "work activities" we mean (more or less) well-defined tasks or processes that a person has to carry out as part of his/her job, often using computers as part of the activity.

The background for this activity-centered perspective on computer support is a studies of healthcare practices and our theoretical work within CSCW and Activity Theory [3, 2,4,5]. There is a range of challenging properties of medical work, which makes it fundamentally different from typical office work: extreme mobility, ad-hoc collaboration, interruptions, high degree of communication, etc. This makes healthcare an interesting application area for the design of pervasive computing technology.

The paper will briefly present some of the key properties of healthcare work and will discuss their influence on the design of a pervasive computing infrastructure. We outline an early prototype implementation of an *activity-centered computing infrastructure*,

G. Borriello and L.E. Holmquist (Eds.): UbiComp 2002, LNCS 2498, pp. 107–116, 2002.

whose design is based on the outlined principles. Finally, we present some results from our design and evaluation workshops with healthcare staff, and ends the paper with a discussion of our work.

2 Healthcare Work and Pervasive Computing Technology

Our work has been carried out in the Center for Pervasive Computing (CfPC) [7] in Denmark, specifically in the research area “Pervasive Healthcare” [13]. Our work is based on studies of medical work at several large Danish hospitals [2,4,5] and close cooperation with clinicians in a Participatory Design process. A cornerstone in our development- and design validation effort is workshops in which clinicians perform role-playing games of future work situations using our prototypes.

At the hospital that we are studying in the present project the patient medical records are paper-based. This situation is going to change in the near future, however, as the hospital has contracted a major Danish software company for developing and deploying a system that is both an Electronic Patient Record (EPR) and an integration portal to a number of older computer based systems. The contracted EPR is rather “traditional” as it is based upon desktop and laptop computers and standard keyboard-based authentication procedures. It is organized as a number of applications that handle specific domains such as prescription, medicine schemas, X-rays, blood samples, etc. It provides standard “window and menu” navigation in a graphical user interface environment.

In this section we describe some key properties of healthcare along with our proposals and visions for support by a pervasive computing infrastructure.

2.1 Shared Material

Clinicians must share a lot of information stored in various artifacts. A prominent example is the medicine schema: all prescriptions are made in a patient’s medicine schema and it is therefore a resource, which is extensively shared among physicians responsible for the prescription and nurses responsible for giving the prescribed medicine to the patient. All the medicine schemas for the ward’s patients are collected in one red binder. Hence, this red binder becomes extremely central in most of the work concerning medicine at the ward, and we have experienced that clinicians spend a considerable amount of time looking for this red binder.

Of course, a key motivation for the contracted EPR is that it is supposed to solve the problems of finding and accessing shared material. It introduces a lot of new problems, though. Accessing medical data means accessing via a computer. Thus, valuable time is spent on frequent identification and authentication (keying in name and password) on computers in the many locations a clinician visits during his/her working hours. Secondly, valuable time is also spent on reestablishing the computational context for the task at hand: finding the right patient, open the proper applications, fetching data, etc.

2.2 Organizing Work in Activities

To the outsider, healthcare work may seem chaotic: nurses and doctors rush around and seldom sit down, they interrupt each other frequently, and their pagers or phones constantly distract their attention. However, patient treatment is organized and managed through a set of well-defined *tasks* or *activities* that must be carried out and are known and agreed upon by all clinicians. Activities may range from very simple ones, like fetching a glass of water to a patient, to very complex ones, like determining the treatment of a patient based on lab results, experience, and talking to the patient. Many activities are organized in time, e.g., medicine must be given at the proper time; others can be dealt with as time permits. Some activities have high priority, like immediate treatment of a patient suffering a heart attack. Many activities are planned in advance, like sending a patient to have X-rays taken, while others happen randomly, like fulfilling the request of a patient.

Our main proposal, outlined in more detail in Section 3, is to model work activities as first class objects in the computing infrastructure thereby lessening the gap between the healthcare tasks and the work done using the computer. These *computational activities* help clinicians to do their job by maintaining the computational state of all applications used for the specific activity. This allows the clinicians to swiftly suspend or resume his or her pending activities and the associated computational state: patient record data, set-up of applications, windows, and user interface, communication links, etc.

2.3 Mobility

One of the most striking features of medical work is its nomadic nature—clinicians do not have a desk or a personal computer and they seldom sit down at all. The work of e.g. physicians is distributed across the whole hospital involving walking to and from different departments, wards, outpatient clinics, radiology departments, etc.

Many activities are linked to certain rooms or the presence of certain artifacts. An example is giving medicine to a patient, which is done at regular hours during the day and requires the medicine, the patient, and the nurse to be located together, typically at the bed of the patient. Another example is pouring medicine into a patient's personal medicine tray, which is only done in the medicine room.

Thus, clinicians' work is extremely mobile and cannot be hindered by carrying heavy equipment. This rules out desktop as well as laptop computers. Many activities involve viewing "bulky" data like X-rays, medical records, and graphs over lab results, etc. This rules out small devices like PDA's for many classes of activities except the simplest ones. Our conclusion is the need for computers with medium-sized to large screens available for clinicians "everywhere". We denote such computers *public computers* to stress that they are not personal but are available for anyone to use—even patients or their relatives. They range from wall-sized displays, through laptop-sized screens mounted in or near the beds, to PDA-sized computers that clinicians can grab and put into the pocket.

Utilizing public computers instead of personal computers means that a person's computational activities cannot be stored on the device. Thus, we must require that it is the infrastructure that *manages, stores, and distributes computational activities*.

If a person's work activity is associated with a computational activity, the infrastructure is required to be able to provide access to the person's activities swiftly on any public computer in his/her vicinity. This requirement therefore rules out authentication procedures that are not very fast, specifically the traditional, cumbersome, keyboard based login using username and password. We find that a system of *proximity-based* *authentication* is very interesting in this light. I.e. a user is authenticated to the infrastructure by proximity to a public computer. This can be achieved by the person wearing a location sensing device like active badges [6,16] or similar, or by presenting some artifact to the system: finger-print reading or a personal smart card. The challenge here is to devise such a proximity-based authentication mechanism that is easy to use, while at the same time meets the requirements for secure identification and authentication of the users in a healthcare setting.

As mentioned, many healthcare activities are recurring and linked to certain artifacts, places, and persons. This enables *proactive inference of activities* to be made by the infrastructure based upon heuristics about these recurring activities and the location of people and artifacts. This, of course, requires that the infrastructure has access to *real-time location information*. Tapping from such a source of location information has several advantages besides enabling inference of activities. For instance, it allows people to locate specific persons or clinical roles (like "nearest radiologist") or artifacts (like a lost medicine tray).

2.4 Interruptions

Collaboration means "interruptions" in clinical work. Coiera points out that clinicians preferentially turn to each other for information and decision support, even when computer-based decision support systems are available [10]. The point is that ad hoc conversations (i.e. interruptions) are not an evil but an interactive process of sharing and interpreting information.

This poses a requirement on the computing infrastructure to allow a user to be able to switch between his/her computational activities swiftly and seamlessly. This will allow a clinician to be interrupted in one activity, handle the interruption that may involve finding another patient's data, other applications and views, and then afterwards simply resume the previous activity thereby automatically reestablishes the computational context.

A concern is how the clinician accesses his/her list of pending activities—clearly we need a user interface element that provides this access. We denote this component the *activity bar*. The activity bar is partly inspired by the Windows task bar—it runs on every public computer and though it you have access to all your activities and may select one to be activated.

It also poses a requirement of *high availability* as switching swiftly and seamlessly between computational activities requires the data embodied in an activity to be available with minimal delay. In making critical decisions based on shared material like patient records it is vital that the material is up-to-date and different users access the same information. We thus find that this rules out unreliable networks between public computers and between centralized components of the infrastructure. In healthcare there is also often need to access large pieces of information, like X-rays, that require high bandwidth.

2.5 Ad-Hoc Collaboration

Another characteristic aspect of medical work is its collaborative nature. Studies show that a large fraction of clinicians' time is spent on discussions [11,14]. The work of treating and caring for patients involves a lot of different types of clinicians, like physicians, radiologists, anesthesiologists, nurses, secretaries, etc. Even though there is a formal division of work among different types of clinicians, this collaboration is highly ad hoc and “on demand”.

Hence, the computing infrastructure should support *collaborative computational activities* as a fundamental building block. This means that users should be able to share activities: by taking turns working on the activity, allowing them to hand over activities to each other, and enabling them to collaborate on an activity simultaneously. Furthermore, the infrastructure should support communication and support for collaboration across time and space.

3 Activity-Centered Computing

3.1 Vision

Healthcare has a long tradition of using computer-based systems, and a clinician is today faced with many different systems and even faced with a wide range of functionality within each one of them. Thus, carrying out a single activity typically involves a lot of different systems and a lot of specific functionality and data presentation within each system.

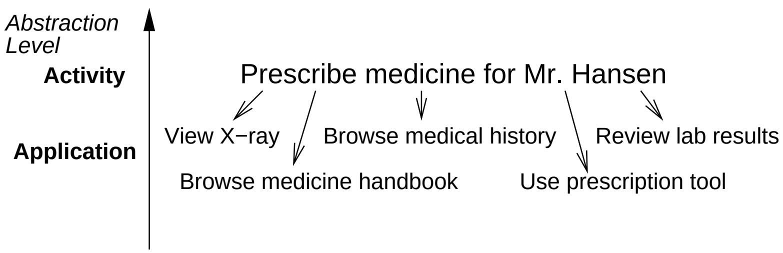

Fig. 1. A single activity involves many applications

This is illustrated in Figure 1. If you ask the doctor what he is doing, he would answer “I’m prescribing medicine for Mr. Hansen”. If you instead view it from the computational level, the doctor is actually handling several distinct applications: reviewing the medical history, looking over the medicine schema, studying X-ray images, etc. Thus, we can identify at least two levels of abstraction, namely the high level of human activities and the low level of computational services/applications manipulated. We denote these levels the *activity level* and *application level*, respectively.

Our key argument is that the computing system does not support the activity level, only the application level. Our aim is therefore to explore how to support the activity level directly in the computing system; explore what the concept of “activity” is in this context, and to evaluate how activities may help clinicians in their daily work.

3.2 Proposal

We formulate our key proposal as follows:

> *To support users with their physical work activities, the computing system must understand the concept of an activity and handle it like a first class object.*

We denote this object a "computational" activity (or simply an activity) that in a sense becomes the computational "granule" provided by the computing system.

We envision a computing system where *domain-oriented services* are implemented and executed on top of an *activity-centered computing infrastructure*. This layering is similar to the well-known middleware concept, like CORBA and J2EE, where applications/services adhere to rigorous rules defined by the middleware platform so it can draw upon high level functionality provided like remote method invocation, serialization, transaction security, etc.

The envisioned activity-centered infrastructure is deployed on every pervasive computing device. It provides standard middleware features related to user authentication, security, etc., but the main point is that it treats activities as first class objects, that is, it facilitates the management of activities like storing, retrieval, forwarding to relevant services, etc.

Domain-oriented services are a set of services/applications related to the particular domain; in healthcare this involves electronic patient record systems, X-ray viewers, laboratory test booking systems, etc.

3.3 Computational Activity Concept

A computational activity is the digital equivalent of a physical activity; for instance the activity of prescribing medicine for a given patient in healthcare can be mirrored by a "prescribe medicine activity" that embody all relevant computational state for that activity: identity of the patient, of the doctor, time and date, medical record data, lab results, etc., as well as used applications, views and user interface interaction state. It follows that computational activities can be *classified* in the same way that human work activities are.

A user is typically actively involved in one activity at a time while a set of other activities is pending. The user may at any time suspend an on-going computational activity to start a new one or resume one from his/her personal list of pending activities. Activities may 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.

Similar ideas have been explored in the Aura project [1]; Wang et al. [15] introduces the term *task-driven computing* defined as a computing environment where mobile users interact with the services and resources in terms of high level tasks and free them from low level configuration activities.

3.4 Initial Prototype

Our prototype architecture is sketched in Figure 2. Rectangles represent either processes/active objects (thick borders) or passive objects/databases (thin borders). Dashed lines represent data flow between objects while solid lines represent event- and data

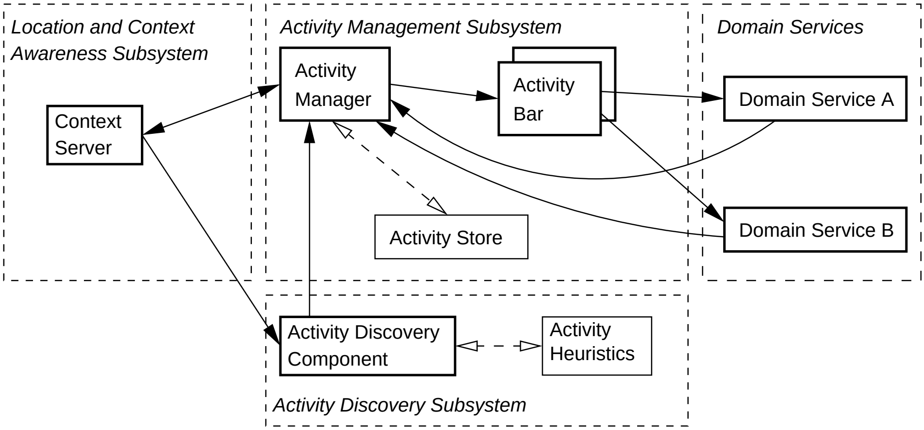

Fig. 2. Architecture of our prototype

flow. Subsystem boundaries are indicated by dashed rectangles. The location and context awareness subsystem contains several components dealing with location tracking and hardware handling which have been left out for the sake of overview.

The design is partitioned into four subsystems:

– *Location- and Context Awareness Subsystem* is responsible for 1) providing real time location- and context-data in a technology independent fashion and 2) to store and manage context-data. In our prototype, our location-monitoring set-up is based on ICode tag-scanners and passive radio frequency identity tags (RFID-tags) that are glued onto a medicine tray or worn on a clinician's coat. Low level tag scanning events are mapped by our context server into domain relevant events, like e.g. an event about a person leaving a room, or a medicine tray being put onto a bed's table.
– *Activity Management Subsystem* is responsible for 1) storing and managing activities and 2) distributing a user's activities to activity bars running on public computers in his/her vicinity. In our prototype, activities are modeled as serialized objects that are passed between a centralized activity manager and the activity bars running on the individual public computers. In Fig. 3 our present proposal for an activity bar 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—this brings up a hierarchical menu of his pending activities. Selecting an activity from the list reestablishes the selected computational activity on the public computer.
– *Domain-oriented Services* is responsible for 1) providing end users with domain related services; 2) providing comprehensive state objects to the infrastructure upon request and 3) reestablishing domain object and interaction state based upon a supplied state object. Given our focus on the healthcare domain, we have focused on services typically belonging to an Electronic Patient Record like medicine schema, patient lists, X-ray viewer, etc. These services run in a classic client-server set-up.
– *Activity Discovery Subsystem (ADC)* is responsible for autonomously inferring likely activities going on in the environment based upon location of people and artifacts, context information, and heuristics about recurring activities in healthcare. In our

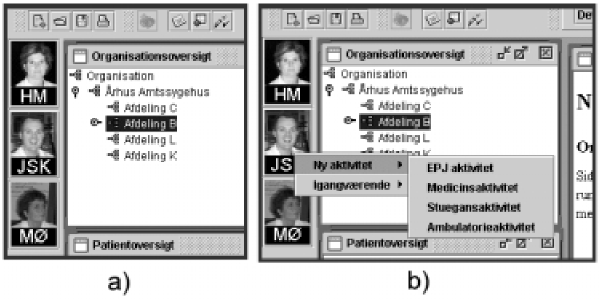

Fig. 3. A snapshot of present design of activity bar.

prototype, we have adopted an expert system [12] as it allows us to express heuristics declaratively and let the inference engine ensure that all possible combinations are handled. More detail can be found in [8,9].

At the moment the prototype only supports individual and non-composite activities. However, we are at the moment engaged in further development for supporting collaborative and shared activities used typically in clinical conference situations, as well as composite activities, where activities can be subordinate to other activities.

3.5 Initial Experiences

The basic idea of activity-centered computing and the functionality of the current prototype has been the subject of six intensive design and evaluation workshops. These workshops have encouraged us to carry on with the activity-centered design. The clinicians were particularly receptive to the proximity-based authentication function, the activity modeling of several services, the ability to suspend and resume activities in a distributed landscape of public computers, and the support for fluent interruption of each other.

Our workshops also highlighted a number of challenges. At the concrete level, the use of RFID tags and scanners had limitations, as the detection range is small, about 0.5 meters. Even small movements by the person meant that a tag worn on the coat was interpreted as leaving the public computer. The infrastructure responded by suspending the person's computational activity leading to a very frustrating user experience. Clearly, RFID techniques have to be combined with other location tracking techniques with a larger range.

There is also a substantial challenge in making the system provide value to the clinicians as they perform their tasks, and not interpose another level of concepts and user interfaces to attend to. A lot of questions were raised such as: how to represent activities in the interface, how to recognize activities, how and when to create new ones and to delete old ones, and whether sufficient rules for activity discovery can be defined for the ADC.

4 Discussion and Future Work

We have outlined our proposal for an activity-centered infrastructure for supporting nomadic, collaborative, intensive, and often interrupted work as we have seen it in healthcare work. The core idea is to let human activities be mirrored in computational activities that follow their owner to any pervasive and public computer devices in his or her vicinity. Modeling human activities in the computer sounds similar to the core idea of workflow systems where activity models are used to automate (office) work and to route work around to human 'resources'. We must stress, however, that our activity-centered design idea is not to be viewed as a workflow system (see also [2]). In our view, a human activity precedes the computational activity that mirrors it, whereas in a workflow system the computational activity precedes and dictates the human activity. Furthermore, we have no intention of modeling all activities within a hospital. Instead we want to make the activity concept available to be used when appropriate. Indeed, there is no need for modeling all activities, as nothing forbids the user to access systems and applications directly at the application level without going to the effort of defining or using a computational activity.

A lot of issues and work remain. One important issue has already been mentioned, namely to support collaborative and shared activities. We are currently working on this aspect. Our project benefits greatly from the collaboration with clinicians whose daily work is hectic, complex, and highly mobile; the feedback they provide in our workshop evaluations are extremely valuable. However, it also forces us to focus much on end user functionality and therefore many issues remain on the infrastructure side. Security is a major concern that needs addressing further; our present prototype actually allows one clinician to act on behalf of another that happens to pass by the public computer the first clinician is using. This is of course not acceptable. Scalability is another issue: our infrastructure design has inherently centralized components that may lead to poor scalability. For instance tracking every clinician and patient as well as all medicine trays, wheel chairs, beds, etc. at a large hospital on a single centralized server is not feasible. Also we have so far not addressed the important issue of reestablishing user interfaces on devices with varying properties like screen size. These issues are subject for further research at CfPC.

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. We like to thank the clinicians from Aarhus University Hospital for our enjoyable collaboration in the pervasive healthcare project.

References

1. Aura Project. http://www.cs.cmu.edu/~aura.
2. J. E. Bardram. Plans as Situated Action: An Activity Theory Approach to Workflow Systems. In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work*. Kluwer Academic Publishers, 1997.
3. J. E. Bardram. *Collaboration, Coordination, and Computer Support - An Activity Theoretical Approach to the Design of Computer Supported Cooperative Work*. PhD thesis, Department of Computer Science, Aarhus University, Aarhus, 1998.

4. 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.
5. C. Bossen. The parameters of Common Information Spaces: the Heterogeneity of cooperative Work at a Hospital Ward. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work CSCW2002*, New Orleans, Louisiana, USA, Nov. 2002. ACM.
6. A. L. Cambridge. The Active Badge System. http://www.uk.research.att.com/ab.html.
7. Center for Pervasive Computing. www.pervasive.dk.
8. H. B. Christensen. Using Logic Programming to Detect Activities in Pervasive Healthcare. In *International Conference on Logic Programming ICLP 2002*, Copenhagen, Denmark, Aug. 2002. Springer Verlag.
9. 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/.
10. E. Coiera. When Conversation Is Better Than Computation. *Journal of the American Medical Informatics Association*, 7(3):277–286, May 2000.
11. D. G. Covell, G. C. Uman, and P. R. Manning. Information needs in office practice: are they being met? *Annals of Internal Medicine*, 103(596–599), 1985.
12. E. Friedman-Hill. Jess, the Rule Engine for the Java Platform. http://herzberg.ca.sandia.gov/jess/.
13. Pervasive Healthcare. www.healthcare.pervasive.dk.
14. P. Tang, M. A. Jaworski, C. A. Felllencer, N. Kreider, M. P. LaRosa, and W. C. Marquardt. Clinical information activities in diverse ambulatory care practices. In *Proceedings of the American Medical Informatics Association Annual Fall Symposium*, pages 12–16, 1996.
15. Z. Wang and D. Garlan. Task-Driven Computing. Technical Report CMU-CS-00-154, School of Computer Science, Carnegie Mellon University, May 2000.
16. R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, Jan. 1992.