Supporting Local Mobility in Healthcare by Application Roaming among Heterogeneous Devices

Jakob Bardram[1], Thomas A. K. Kjær[2], and Christina Nielsen[1]

[1] Center for Pervasive Healthcare
Department of Computer Science, University of Aarhus
Åbogade 34, DK-8200 Århus N.
{bardram,sorsha}@daimi.au.dk
[2] IBM Global Services, IBM Denmark
Dusager 4, DK-8200 Århus N.
thomas.kjaer@dk.ibm.com

Abstract. This paper presents results from a research project aiming at developing an architecture supporting local mobility within hospitals. The architecture is based on fieldwork and design workshops within a large Danish hospital and it has been implemented and evaluated after a pilot phase. Our fieldwork has emphasised the differences between *remote mobility*, where users travel over long distances, and *local mobility*, where users walk around within a fixed set of buildings and/or places. Based on our field studies and our design work, we conclude that local mobility puts up three requirements for computer support; (i) it should *integrate into the existing infrastructure*, (ii) it should support the use of various *heterogeneous devices*, and (iii) it should enable seamless *application roaming* between these devices. The paper describes how these requirements were realized in an architecture for local mobility, and how this architecture was implemented in the healthcare domain.

1 Introduction

Mobility and mobile computing is playing an increasing role in human-computer interaction research and design as a result of the ever-growing range of technology now supporting mobility. Mobile computers, laptops, tablet PCs, PDAs, cellular phones, and hybrids are all devices intended to support mobility of users, and the proliferation of wireless network access like WLAN, UMTS, GPRS, and GSM all support mobile computing. The increasing deployment and use of such mobile technology pose several challenges to the design of the user interaction, and we have already seen much interesting research on mobility and HCI related issues (e.g. [4, 8, 1, 7]). There is, however, a tendency to view mobility as a way to carry on working, while detached from your (physical) desk at work. For example, when users travel, attend meetings, drive a car, are in public places, etc. This is clearly the use case for laptops, but also for PDAs in many cases. The focus in that scenario is to design mobile computer support for users on a singular device, which can be used while away from 'the desk', where the 'real' work seems to happen. In this paper we want to draw the attention to another kind of work setting where people do not work at a desk, never move away from their work place, but yet are extremely mobile, namely work at a hospital. We found that medical work is highly mobile, but not in the sense of travelling long distances. Rather, the mobility in their work entails walking between the different sites within a hospital that a clinician needs to visit as part of her / his job. The work we have been observing thus corresponds to the local mobility, described in e.g. [3, 12, 1], where people move between buildings or rooms in a local environment. Bellotti and Bly [3] argue that we need to distinguish between local mobility and the more traditional notion of mobility which typically takes place between remotely distributed collaborating groups (remote mobility) because the needs for support vary greatly and are sometimes contradictory between the two modalities. This kind of local mobility poses new challenges for the design of computer support for mobility, and especially for the user interaction.

In our effort in designing computer support for local mobility we learned several things. First, when designing for local mobility, the *computational context* becomes relevant to consider in details. The mobile device is no longer isolated in the palm of a user sitting in an airport, or in a car driving in the streets, but is embedded within a complex infrastructure of existing computers, networks, and applications. For example, in a hospital a mobile solution needs to exist within the infrastructure set up by electronic patient records. Second, the mobile solution is now just an *option* within a range of computational devices at hand. The mobile device is no longer isolated in use either, but the user needs to be able to select from a range of devices to suit a specific task. For example, in the hospital a wide range of mobile devices (e.g. PDAs, laptops, mobile phones), as well as stationary devices (e.g. desktop PCs, projector-based PCs) that exist side by side. And third, the design for local mobility needs to recognise the *high pace in local mobility*. The mobile device is no longer something that is used for a whole journey or during a whole meeting, but can be picked up and used for maybe seconds. For example, when a nurse needs to register the measurement of blood-pressure this might take a few seconds on a PDA, after which she moves to a more comfortable desktop PC for to finish the report, using the keyboard.

The aim of this paper is twofold. On the one hand we want to draw the attention to the kind of mobility termed 'local mobility', which seems to be present in many work settings. On the other hand we want to present our design for local mobility. In the rest of this paper, we describe the hospital department we have studied and our methods. We report on our observations, discussing how mobility is key to their work and valuable for local collaboration. We then describe some of the design requirements for local mobility coming out of our studies. Then we present our design for local mobility, highlighting how we have addressed some of the thing found during our study. Finally, we conclude the paper with a discussion of related and future work.

2 The Project

Our design for local mobility takes its departure at a surgical department at a large metropolitan hospital in Denmark (see figure 1). We studied the work within the department as a whole and with a bed ward in particular. Our mobile solution for accessing medical data was subsequently put into field trial at this ward. We start by describing the background for the project, the department involved, and our research methods.

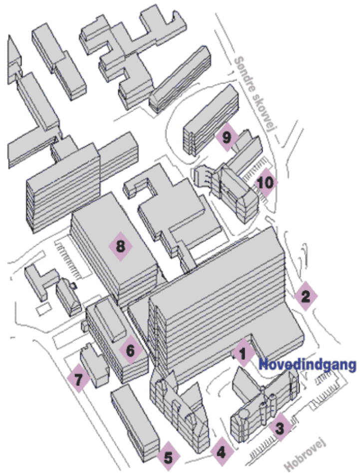

Fig. 1. Aalborg hospital. Important locations are the main building (1), the emergency room (2), and the administration building containing the offices for department T doctors and secretaries (3). Department T's operating rooms and wards are located in the main building (1) on two different floors.

2.1 Background - Mobile Support for Electronic Patient Records

Currently, there is an extensive focus on Electronic Patient Records (EPR) in Denmark. The current government has dictated that all hospitals in Denmark by 2005 should have total coverage for all patients in EPR systems. It is, however, up to the regional authorities (the counties) running the different hospitals to decide on the exact solution and vendor. Common for all EPR systems currently being implemented in the hospitals is that they all run on desktop PCs, and thus do not have any support for mobility. Taken the high degree of mobility within hospitals into account, there is a substantial motivation for both EPR vendors as well as the hospital administration to develop solutions for mobile access to the EPR systems. This project is made in cooperation with one large vendor of EPR systems to the Danish marked, focusing on the design of a mobile solution for clinical work building on top of an EPR system.

2.2 Department T

Department T specialises in surgical procedures relating to the heart, lungs and stomach – for example bypass operations and replacing heart valves. The department performs approximately 15 counts of heart surgery every week. Department T consists of one ward where the patients are initially admitted before surgery and transferred back to post-op treatment after having spent 24-48 hours at the intensive care unit immediately after surgery. The ward can carry 30 pre- and post-op patients, and department T treats approximately 1300 patients a year. The ward occupies the sixth floor in the main hospital building (no. 1 in figure 1), whereas the surgeons' and head nurse's offices are located on the second floor in the administration building (no. 3). Overall, the department employs roughly 20 surgeons, 50 nurses, 8 perfusionists and 6 secretaries.

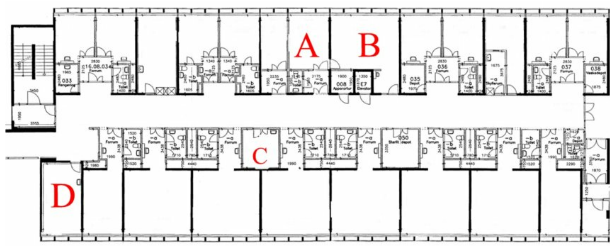

Fig. 2. Ground plan of the ward at department T. Important locations are: the ward office (A+B), the medicine room (C), and the conference room (D)

The ground plan for the ward is illustrated in figure 2. The number of doctors and nurses present at the ward changes depending on the time of day. In a day shift, 13-15 nurses are working at the ward while 8-10 surgeons do the morning round before proceeding to operating theatres, whereas during the night shift the ward is 'guarded' by 3-5 nurses with 2 doctors on call. Department T has been using the EPR system for 2 years and is one of the departments in Denmark with most experience in using EPR systems.

2.3 Research Methods

The goal of the project is to examine the possibilities for supporting work practice at the hospital with mobile technology. It was decided to pursue this goal by (i) conducting extensive field-studies of the work at department T, especially focusing on the use of the EPR systems; (ii) initiating an iterative design process with the clinicians, focusing on the design of mobile technology, which could extend the reach of the EPR; (iii) implementing a prototype and install it in a test environment; (iv) carrying out a pilot phase where clinicians should use the mobile equipment; and (v) evaluating the design.

Field studies. Two researchers made 80 man-hours of participant observations [14] of a mixed group of nurses and surgeons, covering different work tasks (e.g. preliminary patient examinations, different staff meetings, ward rounds, medicine dispensing and a by-pass operation), and different time slots (day, evening and night watch, week-days and week-ends).

Design Workshops. At the end of the field study, we conducted a future workshop [11] together with a group of nurses and physicians with the goal of prioritising and concretising a list of possible features in a mobile solution. During the implementation phase, three additional design workshops were held with this design group to support the collaborative design of particularly the user interface and navigation within the prototype.

Pilot Phase. The final version was installed in a test environment accessing the EPR system running at department T. The pilot phase ran for a total of 12 weeks, after which it was evaluated at an evaluation workshop and by video-recording a nurse as he was using the system for doing his daily tasks.

2.4 Local Mobility in Medical Work

A fundamental characteristic of medical work in hospitals is that 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 surgeons and physicians is the hospital. Consider a typical day for a surgeon. 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 at the first floor of the main block (building 1), attending the radiology conference. Finally, he would take the lift to the ward (building 1) and start 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 visit their patients at the ward. The ward rounds typically start in the ward office (see figure 2). While seated the physician and the nurse(s) go through all the patients, read the electronic and the paper-based medical patient records, and look over results from lab tests, etc. Afterwards, they take various paper-based records along together with other relevant materials (medical handbooks, medicine schemas, and small medical instruments), and visit each of the patients at their bedside. Thus, the medical team moves around the wards carrying the paper-based material. 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 the ward, which are located on a different floor at the hospital. The physician on duty carries a pager and can be 'paged' by everybody at the department, in which case he often has to move around to consult patients and fellow colleagues. For the nurses, mobility is just as critical. They spend most of their working hours moving from place to place as an integral part of patient care. They constantly move between the nurses' station, the medicine room, the patients, the kitchen and various storage facilities, responding to tasks and needs as they occur as well as taking care of planned activities. Because so many needs occur ad-hoc, they have to update patient records when other, and more important, tasks allow for it.

3 Design for Local Mobility

Based on our field studies of local mobility and the use of the EPR system at department T, we have identified three central aspects of creating computer support for local mobility. First of all, mobile computer support needs to be a natural extension of the existing infrastructure already in place in the setting, where local mobility takes place. Second, from a user interaction perspective there is a need for supporting multiple devices, each device capable of supporting specific tasks and situations. Third, there is a need for application roaming in the sense that the alternation between multiple devices can be done ’on-the-fly’ by moving a task from one device to another in a fast pace. Let us consider these in turn.

3.1 Mobilising an Existing Infrastructure

Studying the use of the EPR at department T it became clear to us that there is a built-in tension between using the EPR and the mobile nature of medical work. The EPR at the ward is inherently tied to the desk because it is only running on desktop PCs. However, most clinical work takes place anywhere else but the ward office, where most PCs are located; simple things like handing out medicine to a patient is highly mobile work, including walking between the medicine room and a range of patients, carefully documenting every time medicine is poured, handed out and given to patients. In practice, the use of the EPR for this kind of documentation is impossible because it would require a nurse to walk between the medicine room, the patient’s bedside, and the PCs in the office constantly, having all the trouble of logging in, finding the patient and his medicine chart every single time. The consequence of this inherent tension is that the clinicians tried to mobilise the EPR and make it a tool while moving around. Mobilisation strategies especially involves printing out various parts of the records to be carried around. For example, it has become routine to print out each patient’s medicine chart on paper every morning. In this way it becomes mobile again and be carried around in the nurse’s pocket. However, this eliminates all the benefits of having an electronic medicine chart because it no longer works as a central coordination mechanism. Medication given to a patient is now (as it was done prior to introducing the EPR) documented on the paper-based medical chart, and the electronic version is no longer updated until late in the afternoon. Now, for a nurse to be sure of the medication of a patient s/he will not look up this information in the EPR but rather spend a lot of time trying to locate the nurse who carry the print-out in a pocket. Hence, it is important that support for local mobility is tightly integrated with the existing infrastructure. In our case, the design of mobile support for medication should be real-time integrated with the EPR medicine chart, and cannot rely on synchronisation, which is a common strategy for PDA usage.

3.2 Supporting Multiple Devices

Another distinctive aspect of the mobile work at department T is the constant alternation between tasks. This is due to both the fact that clinicians attend many patients simultaneously as well as because there are many interruptions, partly as a result of the highly ad hoc nature of much medical work at a surgical department. This multi-tasking has resulted in a strategy for having multiple artifacts that can support the same task as well as duplicating information in several places. For example, at department T measuring blood pressure and taking the pulse frequently is important for post-op patients. Hence, everywhere at the ward there are instruments available for this kind of measurement in large numbers. As for the EPR, it has been constructed in a way so that a user can leave the PC, lock the screen, go do something else and later return to the PC, unlock it and resume his / her work as s/he left it. This feature is considered highly useful at the ward. However, problems with this feature also occurs, because it often resulted in a situation where all 8 PCs are locked, leaving them unusable for others. This is often the situation where a nurse had gone in order to make a simple measurement of e.g. blood pressure for later to return and type it in. Another interesting observation is that the use of the EPR was abandoned in the medicine room, where all medicine is poured in small containers for each patient. One would imagine that a PC here would be useful, but the UI design and layout of the medicine chart has not been made to fit the tasks in the medicine room. Furthermore, there is no room for a PC in the small room where all the table space is needed for handling medicine. Our conclusion from these observations is, that there is a need for supporting multiple devices when supporting local mobility. It is important that the nurse who is measuring blood pressure can use a small PDA type of device to input simple data, whereas the nurse typing long notes in the record needs a full-fledged keyboard and mouse, and a comfortable working setup at a desk. Correspondingly, when moving to the medicine room, the nurses' requirements for the display of medicine charts change and the technology made available to them in this situation should reflect that.

3.3 Supporting Application Roaming

The way that a task is carried along is characteristic to local mobility. At the ward round, for example, the surgeon and the nurses carry the same set of tasks around for the duration of the round. It is basic to local mobility that a task is taken to different locations. Therefore, it seems important to be able to support that a task can 'follow' the user. This can clearly be achieved by using a single mobile device, which is also the common technological strategy in contemporary mobile computing. However, if we want to take the support for multiple devices seriously, we should also provide mechanisms for transferring an ongoing task from one device to another. For example, when the nurse has finished measuring the blood pressure using a PDA, she returns to the PC and transfers the 'blood-pressure-measurement' task to a desktop PC where she can use the keyboard to add a note about the general condition of the patient due to high blood pressure. We find this last requirement for application roaming among multiple devices a central contribution from our studies of medical work.

4 An Architecture Supporting Local Mobility

Based on the requirements above, we have designed a generic architecture for supporting local mobility and we have implemented a first prototype of this architecture in the healthcare domain. This section presents this architecture and its implementation in the hospital.

4.1 A Simple Healthcare Scenario

Ms. Hansen, a nurse at department T, picks up a PDA at the beginning of her working shift. She starts by logging in and then walks around to her patients saying good morning and measures their morning temperature and other central health data. She types in the data on the PDA, which is relayed immediately to the EPR for others to see. She returns to the ward office to meet with the surgeon for the morning ward round. In front of a PC she uses the bar-code reader in the PDA to scan the bar-code strip on the PC, thereby transferring her ongoing task to the PC. The PC displays the same page as the PDA, but now accommodates the much larger screen by containing much more details than the PDA. She shows the surgeon a particular critical reading, and uses the keyboard to key in a note. The surgeon and the nurse look up the patient to be visited first on the ward round, and find his medicine chart. When ready to leave the ward office, they hit the 'Resume Session' button on the PDA, which reloads the medicine chart, adapted to this smaller screen. They now visit the patient at his bed-side.

4.2 Architecture

The overall functional architecture is shown in figure 3. There are two important components in the architecture. The *Mobile Application Server* (MAS) is responsible for the support of heterogeneous devices and the application specific logic, including integration to the existing infrastructure. The *Application Roaming Server* (ARS) is together with the JMS Server and JMS Clients, responsible for application roaming among the different devices.

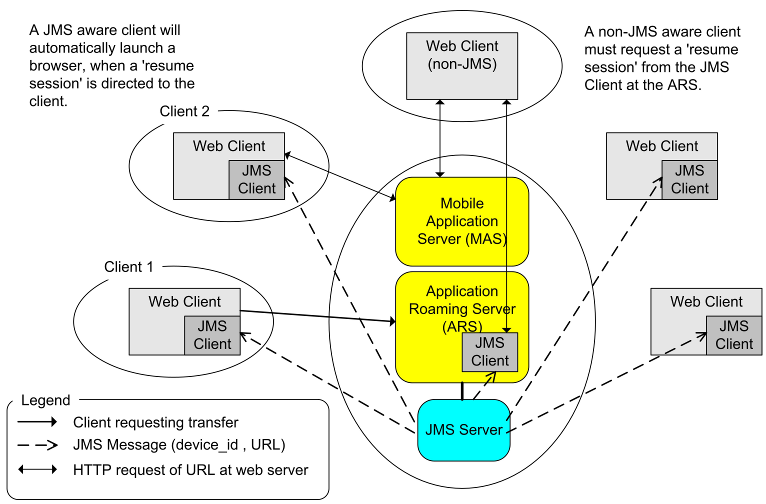

Fig. 3. Overall Functional Architecture

The architecture of the MAS components (detailed in figure 4) implements a web-based interface to an existing infrastructure using existing access paths to databases using e.g. JDBC, Enterprise Java Beans (EJB) or application programming interfaces (APIs). The architecture is a transformation engine based on the Model-View-Control design pattern [9]. The *Controller* component is responsible for interaction control with the client, including ensuring user authentication and identifying the requesting device type. The Controller is made up of a set of servlets handling the HTTP requests from the clients. The *Application Logic* component implements the actual logic of the application by using a command pattern [9] to specify the interface for the command beans. The *View Page Construction* component is a set of Java Server Pages (JSPs), which produce the response for a particular request. In contrast to traditional web server applications, the JSP pages in our architecture do not produce HTML to be sent back to the requesting client. Here, the view is constructed in XML, and then processed by a XSLT *Transcoder Service* component, which will apply an XSL style sheet to produce the HTML content appropriate for the requesting device. By following this approach, we are able to support different devices, each having different capabilities, like network bandwidth and display size and colour. But the response is based on the same content, described by the XML produced by the JSP view pages.

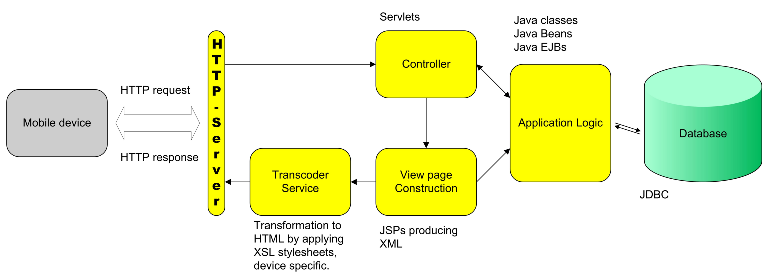

Fig. 4. relatively easy Functional Architecture of the Mobile Application Server (MAS) component

From a usability point of view, the mobile solution in many respects resembles the existing EPR systems. In figure 5 it is easy to see the resemblance between the 3 types of views on medicine; the same kind of information is listed in the same order and the same kind of colour coding is used. We have, relatively easy, been able to design, implement, and deploy new user interfaces for different situations. Figure 5 shows the two different design of the medicine chart for a PDA screen and a PC screen, respectively.

The functional architecture to support Application Roaming is shown in figure 3. We rely on Java Messaging Services (JMS) to handle the notification of application roaming events in our current implementation of the architecture. JMS is especially well-suited for this purpose, because the asynchronous communication in JMS is ideal for this kind of loosely coupled communication among devices. Asynchronous communication will ensure that neither the client nor the server will deadlock in an unsuccessful attempt to transfer the session from one device to another. However, we cannot always expect all mobile devices to support JMS natively. Therefore, the AR server also contains a 'pseudo' JMS client that manages messages to non-JMS clients (e.g. a Pocket PC device). The interaction diagram in figure 6 illustrates the sequence of events between the various components when one client (the *Producer*) tries to transfer its session to another client (the *Consumer*). The upper part of figure 6 depicts application roaming to a JMS-aware client (e.g. a PC) and the lower part depicts application roaming to a non JMS-aware client (e.g. a Pocket PC device). The basic steps for client 1 to transfer its session to a JMS-aware client 2 are: Client 1 sends a transfer request to the AR

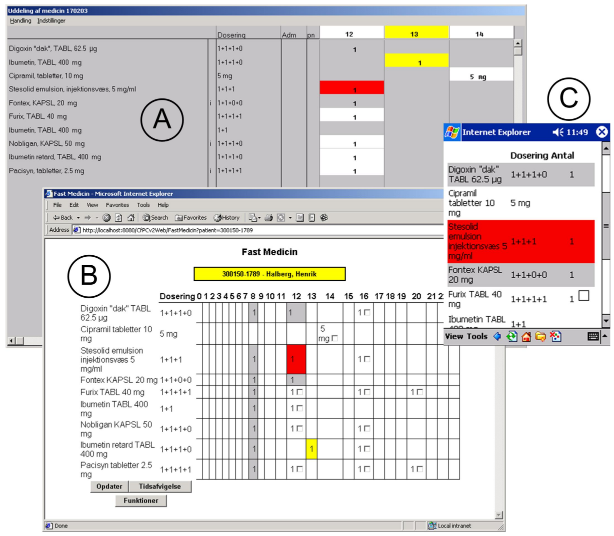

Fig. 5. Screen Shots showing the Medicine Chart. (a) from the normal EPJ system, (b) the web-based user-interface on a normal PC browser, and (c) the user-interface on a PDA

server via HTTP and the ARS returns a response to the client (A in figure 6). When the transfer controller in the ARS is activated it publishes a message, containing the destination client id and URL, on a JMS topic (B). Every JMS-aware client runs a listener, always listening for messages on this topic. On the consuming client (client 2), with the corresponding device id, the JMS listener launches a browser on the client (C), and asks the browser to request the URL within the message from the MAS web server (D). If client 2 is a non JMS-aware client (the lower part of figure 6), the ARS puts the message in a JMS client (E). Client 2 can now manually ask the ARS if there is any application roaming to be activated on it (F). If there is, the url is returned and this url can be fetched from the MAS. Finally, when a user has an on-going session with the MAS, the current url is always stored in the user's session object. When a user transfer a session from e.g. a PDA to a PC, the PDA reloads with a page with a 'Resume Sesion' button on it. This page maintains a link to the on-going session object, which is used on the PC. When activating the resume button, the current url (as shown on the PC) is reloaded in the PDA's browser.

4.3 Current Implementation of the Architecture

The server-side of the architecture is implemented on an IBM WebSphere Application Server (WAS) version 4.0.4. The 'Application Logic' component in the MAS (figure 4) is ibut have to fit in time in front of the computermplemented as Java Enterprise Beans accessing the EPR server using a JDBC connection accessing a remote IBM DB2 client connection to the EPR DB2 database server. This remote DB2 connection is protected

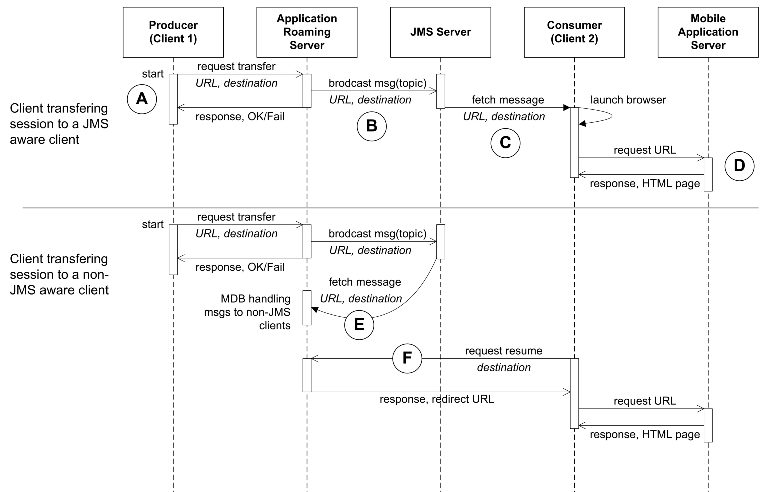

Fig. 6. Interaction Diagram showing application roaming between two clients

through a firewall, allowing only request from our DB2 client to access. The data accessed includes user authentication and access control lists (ACL) so that users can use their normal user names and passwords. The 'Controller' component is implemented as servlets, which each checks the user authorisation of the client trying to access the server. The 'View Page Component' is implemented using JSP pages producing XML for the 'Transcoding' component to transform to HTML using XSL style sheets. This uses the Apache Jakarta XSL Tag Library version 1.1. Identification of client devices is done by examining the client's HTTP request header. On the client side, the current prototype supports 3 types of clients. A Pocket PC based PDA with a built-in barcode scanner from Symbol Technologies, an EPOC based client from Psion, and a normal desktop PC. All of these use the built-in internet browser to display the pages. The network connection was made by Wireless LAN (IEEE 802.11b) using Symbol access points with no encryption. The application roaming (the ARS component) functionality is implemented via Java Messaging Services (JMS) using OpenJMS version 0.7.2 as the JMS provider. In our current implementation we have both JMS-aware and JMS-unaware clients (PCs and Pocket PCs, respectively), thus using both types of application roaming in our prototype. Currently, the PCs used at department T does not have barcode readers to be used in the application roaming.

The mechanism for trickering the application roaming process illustrated in figure 6 is currently implemented via the PDA's barcode reader. Using barcodes felt natural, because it is used for other things (e.g. reading barcodes on medicine bottles and on the armband of the patients). A small barcode reader application is running on the PDA and when it scans a barcode it checks whether this is another device (typically a desktop PC) and initiates the application roaming process. Other mechanisms can, however, be deployed in our achitecture, for example using IrDA. The application roaming mechanism is independent on its trickering mechanism on the clients. Application roaming using barcode currently only works from a PDA to a PC or laptop. The 'Resume Session' button is used to roam the session back to the PDA it came from.

A current limitation in our current implementation is that the state of a user's interaction with a web page is not saved to the server, and is hence not roamed between devices. For example, if a nurse is typing in the temperature in a text field on a web page on the PDA, and tries to roam this page to a nearby PC, then the figures s/he has already entered in the temperature field is not transfered. This is mainly because we do not submit HTML forms before doing application roaming. We are currently looking into methods for handling this issue.

4.4 Evaluation

The first version of the prototype has been running in a pilot test at department T lasting 12 weeks. The setup included one WLAN base station located in the centre of the ward, 2 PDAs, and 2 laptop PCs, all with WLAN access. During this period the two PDAs were used to support mobile work. From the log we can see that approx. 50 users logged on to the mobile EPR system, and that in total approx. 200 request were made to the server, of which approx. 50 requested medical notes and approx. 100 requested medicine information. We concluded the pilot phases with an evaluation workshop where the users could brainstorm on strengths and weakness of the mobile access to the EPR. The strengths were reported to be the mobile access to the EPR, enabling limited but central functionality to be available at e.g. the patient's bed side and in the medicine room. This increased the data quality in using and reporting medical data, like medicine prescriptions. The use of of the laptops were however quite limited. It seems like either the clinicians would use a full-fledged PC on a desk or they would use the PDA that can be carried in a white coat pocket. The in-between laptop devices were not used much. The weaknesses reported was concentrated around long response time, limited screen size and lack of keyboard on the PDAs. A limitation of the pilot study was clearly the limited number of devices (4 in total), and the limited range of just one base station (the latter was also the cause of the long response times reported). It was judged that the number of PDAs should approximately match the number of clinicians in a day shift. The prototype is currently being developed into an official application supported by the vendor and is being marketed and sold to other hospitals. The next step in our project is to implement a better application roaming mechanism and to incorporate context-awareness into the prototype (see 6). This second version of the prototype will be installed at department T and a larger number of PDAs and basestation will be deployed. This new setup will accordingly be evaluated and followed during a new pilot phase. We hope in this way to learn more about local mobility as supported by technology.

5 Discussion and Related Work

Providing a range of devices (stationary, mobile, wearable) for supporting a work practice builds on the understanding of the fact that different people prefer different tools for solving similar tasks, and that a selection of tools makes nurses and surgeons better equipped to deal with the ad hoc demands they face several times a day. Carrying the context with you across devices poses challenges to the user interface design as well as the technical integration between the devices. Rist [15] proposes a technical solution to accessing a virtual meeting place through highly heterogeneous devices based on the development of device-specific user interface proxies not unlike the approach we have chosen here. Roman et al. [16] also explore the challenges of integrating a PDA in a distributed environment. They argue the importance of using PDAs as *enabling bridges* to services rather than treating the PDAs as isolated entities. Their approach to integration is technical and the consistency in their system is supported by contents alone.

Fagrell et al. [7] propose 'FieldWise' as an architecture for Mobile Knowledge Management. Like our architecture, the FieldWise architecture puts emphasis on adapting the response to a client according to its network connection and user-interface capabilities. Furthermore, the FieldWise architecture implements many other features, like support for task overviews with notification mechanisms, overview over records, and suggestions for available expertise. Our architecture does not per se support these latter features. Most of these features are a part of the EPR system and as such can be made available on the mobile devices also. There are, however, also some major differences between the FieldWise architecture and ours. These differences are based on the kind of mobility that has be the target for the two architectures. The FieldWise architecture takes its outset in studies of mobile journalists, who move around whole cities and countries. Hence, there is a major difference between this kind of remote mobility and the kind of local mobility, which we have described and designed for. As we have argued, in local mobility within the premises of e.g. a hospital, it becomes essential to make mobile support that blend seamlessly into the existing infrastructure, including the kind of application roaming we have described. Application roaming is hence not a part of the FieldWise architecture. As for application roaming among heterogeneous devices using web-based access to legacy systems we are not familiar with any related work. However, the work on Activity Based Computing [5] aims at supporting mobile work by enabling users to transfer the state of their work activities between different heterogeneous devices. Similarly, the task management architecture Prism in the Aura project [10] supports mobility by migrating application among heterogeneous computing environments. These two approaches to application roaming, however, involves the migration of whole applications on the client devices. In our work, the 'migration' only takes place on the server side, and is accessed by standard web browsers.

6 Conclusion and Future Work

In this paper we have done two things. First we have analysed mobile medical work within a large Danish hospital. Second, we have designed, implemented and evaluated an architecture supporting this kind of mobile work. Based on our field studies we have argued that there are some fundamental difference between 'remote mobility' and 'local mobility'. The former refers to the mobility of users moving *between* different working sites, e.g. between the office and home, or travelling to visit customers. Much of the literature and the technical solution for mobility concentrate on this kind of mobility. Local mobility, on the other hand, refers to users moving around – often on foot – *within* the same site. Based on our field studies and a range of design workshops with clinicians, we have argued that local mobility put forth 3 requirements, which are distinct to local mobility. First, the technological support for mobility has to *integrate seamlessly with the existing computing infrastructure* at the site. For example, in the hospital setting, the mobile support had to be tightly integrated with the existing EPR system – in function as well as in usage. In a remote mobility situation, this is less important. Here the linkage to more general-purpose infrastructures like IP over UMTS or GSM seems sufficient. Second, the technological support for mobility should support a *variety of devices*. For example, in the hospital environment, users would like to alternate continuously between using small hand-held devices and large desktop or wall-based displays. In the remote mobility situation, this is seldom a core requirement, because there is often only one device available – a laptop, a PDA, or a cell phone. Third, these multiple devices should support seamless *application roaming* among these devices. For example, if a nurse is using a PDA for documenting medicine she would be able to transfer this task, including its present state, to a PC and continue the task of documentation there. In the remote mobility scenario, there is no need for this kind of functionality, because the pace of multi-tasking is less prevalent. We have suggested an architecture that demonstrates how these three requirements can be met and a prototype for mobile access to an EPR systems has been implemented using this architecture. This prototype has been evaluated in pilot studies at the hospital and the outcome of this evaluation is currently being incorporated in a production ready version of the prototype. The architecture and the prototype were developed for mobile work within a hospital. We want to argue, however, that the architecture in general might be suitable for supporting other kinds of local mobility environments. If we look at other projects that we have been involved in (e.g supporting mobile workers at a waste water plant [13] and area managers at the county's Building and Energy Office [2]), mobile work in these settings might also be supported by our proposed general architecture. In the future we plan to take the architecture to other work domains, and apply and develop it accordingly. Another item on our list to do in the future is to make the architecture context-aware [6] in the sense that the Mobile Application Server not only adapts its response to the type of requesting device, but also according to the context of the device. For example, when the medicine chart is requested in the medicine room, the reponse is tailored to show a list of medicine for several patient to be poured now. And when the medicine chart is requested beside a patient's bed, the medicine chart shows only medicine for this patient, in a historical perspective.

7 Acknowledgements

This research has been funded by the Danish Center for Information Technology (CIT) and IBM. Thanks to the clinicians at department T and the employees at the IT department at Aalborg Hospital for their valuable participation in this project.

References

1. J. E. Bardram and C. Bossen. Moving to get aHead: Local Mobility and Collaborative Work. In P. Dourish, G. Fitzpatrick, and K. Schmidtk, editors, *Proceedings of the Fifth*

European Conference on Computer Supported Cooperative Work, pages xx–xx, Helsinki, Finland, Sept. 2003. Kluwer Academic Publishers.
2. S. Bødker, J. Friis Kristensen, C. Nielsen, and W. Sperschneider. Technology for boundaries. Submitted to the 8^{th} European Conference on Computer Supported Cooperative Work, 2003.
3. V. Bellotti and S. Bly. Walking away from the desktop computer: Distributed collaboration and mobility in a product design team. In K. Ehrlich and C. Schmandt, editors, *Proceedings of ACM 1996 Conference on Computer Supported Cooperative Work.*, pages 209–218. ACM, ACM Press, 1996.
4. O. Bertelsen and C. Nielsen. Dynamics in wastewater treatment: A framework for understanding formal constructs in complex techincal settings. In S. Bødker, M. Kyng, and K. Schmidt, editors, *Proceedings of the 6^{th} European Conference on Computer Supported Cooperative Work.*, pages 277–290, Copenhagen, Sept. 1999. Kluwer Academic Publisheres, Dordrecht.
5. H. B. Christensen and J. E. Bardram. Supporting human activities – exploring activity-centered computing. In G. Borriello and L. E. Holmquist, editors, *Proceedings of Ubicomp 2002: Ubiquitous Computing*, volume 2498 of *Lecture Notes in Computer Science*, pages 107–116, Göteborg, Sweden, Sept. 2002. Springer Verlag.
6. A. Dey, G. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction.*, 16:97–166, 2001. Lawrence Erlbaum Associates, Inc.
7. H. Fagrell, K. Forsberg, and J. Sanneblad. Fieldwise: A mobile knowlegde management architecture. In *Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work.*, pages 211–220. ACM, ACM Press, 2000.
8. H. Fagrell, S. Kristoffersen, and F. Ljungberg. Exploring support for knowledge management in mobile work. In S. Bødker, M. Kyng, and K. Schmidt, editors, *Proceedings of the 6^{th} European Conference on Computer Supported Cooperative Work.*, pages 277–290, Copenhagen, Sept. 1999. Kluwer Academic Publisheres, Dordrecht.
9. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reuseable Object-Oriented Software*. Addison-Wesley, 1994.
10. D. Garlan, D. P. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing.*, 1(2):22–31, 2002.
11. F. Kensing and K. Halskov Madsen. Generating visions: Future workshops and metaphorical design. In J. Greenbaum and M. Kyng, editors, *Design at Work: Cooperative Design of Computer Systems.*, pages 155–168. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.
12. P. Luff and C. Heath. Mobility in collaboration. In S. Poltrock and J. Grudin, editors, *Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work*, pages 305–314. ACM Press, 1998.
13. C. Nielsen and A. Søndergaard. Designing for mobility: an integration approach supporting multiple technologies. In *Proceedings of the 1^{st} Nordic Conference on Human-Computer Interaction (CD-ROM)*, 2000.
14. M. Q. Patton. *Qualitative Evaluation and Research Methods*. Sage Publications, London, second edition, 1990.
15. T. Rist. Using mobile communication devices to access virtual meeting places. In *Proceedings of the 2^{nd} Workshop on Human Computer Interaction with Mobile Devices*, pages 81–86, Edinburgh, Scotland, 1999.
16. M. e. a. Román. Integrating PDAs into distributed systems:2k and PalmORB. In H. Gellersen, editor, *Handheld and Ubiquitous Computing. Proceedings of First International Symposium*, pages 137–149. Springer-Verlag, 1999.