Agent-Based User and Task Modelling for Seamless Information Access via Personal Digital Assistants

GOLHA SHARIFI, RALPH DETERS AND JULITA VASSILEVA
Computer Science Department, University of Saskatchewan,
1C101 Engineering Bldg, 57 Campus Drive, Saskatoon, S7N 5A9, CANADA
Phone: 1-306-966-2073, Fax: 1-306-966-4884
Email: {gos787, deters, jiv}@cs.usask.ca

Abstract

The paper describes a case study of how a combination of techniques, such as agent technologies, user, task and context modeling can be used to adapt a distributed application for mobile users with Personal Digital Assistants (PDAs) and support seamless connection and access to a centralized information system in areas of disconnection.

In a domain where users have typical tasks and information needs related to these tasks it is possible, by keeping track of the users' current context (time and location through the user schedule) and a user class stereotype task model, to predict the user tasks and their information needs and to adapt both the functionality and the presentation information provided by the application. Two types of adaptation are important in a domain where there is no reliable connection: hiding from the user the disconnection and ensuring consistency of the data, which is added, deleted or modified by the user on her device during periods of disconnection.

The case study's target domain is the Saskatoon District Health Division, where homecare workers provided with iPaq PDAs equipped with CDPD cards retrieve patient information from centralized information system, make notifications for schedule changes and add new data into the information system, e.g. file reports. The prototype has been evaluated in a simulated environment with the goal to see if it provides seamless connection and access to data in conditions of disconnection.

Keywords: user adaptation, context adaptation, mobile devices, PDA, user and task modelling, context modelling, mobile transaction management, agents

1. Introduction

There are a variety of problems that need to be tackled in the development of successful applications for Personal Digital Assistants (PDAs). Some of them are related to the limitations of the small device: short battery life, small storage and processing power, small screen which makes the interface design more challenging. To ensure scalability, performance and speed, it is necessary to develop tools and techniques to compensate for these limitations. Another group of problems is related to the fact that the device can be used in a variety of contexts, for example, in areas with poor or no wireless connection, in noisy environments or environments that are too dark or too bright, in situations where the user is occupied (e.g. driving), or under stress or high cognitive load, etc. A third major problem is related to ensuring security and privacy of data (customer, patient, etc.) both stored on the device and in transit. Data of this kind is likely to be sensitive and attractive target to criminals.

In this paper, we focus on coping with one of the major context-related problems when using a thin client – the varying quality of the wireless network connection, which can prevent the user from accessing the information server while on the move. Though connections remain fairly good in areas close to network stations, for instance, Cellular Digital Packet Data (CDPD) towers, it is not uncommon to have zones of low connectivity or even no connection due to physical structures, for example close to high concrete buildings blocking the signals. If an adaptive application is able to predict that the user is moving into a disconnected area, it can pre-fetch the information that the user is likely to need while being in the area. When the user tries to access the information, s/he will have a fairly fresh copy of the information pre-fetched from the server, i.e. will have the illusion of a *seamless connection* to the server. If the user attempts to send information back to the server while disconnected, the adaptive application can take care to save the information locally and send it as soon as it gets back in a connected area, without the need of the user doing anything, or even noticing that s/he was in a disconnected area, thus ensuring *seamless access* to the data.

The paper describes a case study of how a combination of techniques, such as agent technologies, distributed database transaction management, and user and task modeling can be combined to provide useful adaptation in a mobile context. The goal is to support seamless connection and access to a centralized information system in areas of disconnection in an application for mobile users with Personal Digital Assistants (PDAs). In a domain where users have typical (well defined) tasks and information needs related to these tasks it is possible, by keeping track of the users' current context (time and location through the user schedule) and a user class stereotype task model, to predict the user tasks and their information needs and to adapt both the functionality and the presentation information provided by the application. The domain of the case-study is supporting non-critical information needs for the homecare workers of the Saskatoon District Health Division (SDHD).

The paper is organized as follows: section 2 presents some related other work, section 3 describes briefly the domain of the case study, section 4 presents the combination of techniques characterising our approach, section 5 presents an evaluation using a simulation, and section 6 provides a discussion of the assumptions, generalizability and limitations of the approach. Section 7 concludes the paper.

2. Related Work

There is a great deal of research on adaptation of the application and web infrastructure for mobile devices. Most of this research, for example CRUMPET (Poslad et al., 2001a; Poslad et al., 2001b), MyCampus (Sadeh et al., 2002) and Hippie (Oppermann et al., 1999; Specht and Oppermann, 1999), focuses on content adaptation, adaptive information retrieval, interface and presentation adaptation. Furthermore, the focus of most of this research is on adaptation based on the user type and the level of experience. Projects, such as Broad-Car (Console et al., 2002), propose adaptation based on the context and use multimedia output, e.g. voice, video or text, for presenting the data to the user. In these studies, the parameters used for adaptation are bandwidth, user type, task, screen size, output channels of the device, and the location. However, there are no write transactions performed in any of these systems, and synchronization and transaction management are not needed.

Muñoz et al. (2003) propose an architecture using agents for context aware communication among hospital workers. Their application adapts to the location by allowing users to send messages and notifications specific to the location of a patient, to particular other professionals (roles) and at particular times. This work is closest to ours with respect to the application domain and the use of agents. However, this work like all other previous approaches assume continuous network coverage and do not deal with disconnection.

3. Domain

There has been an extensive search for applications for mobile computing devices in recent years. One potential application domain is providing mobile workers in healthcare departments, emergency services, and salespersons with access to existing information services through mobile devices such as personal digital assistants (PDAs), and advanced cell phones. These devices are light, highly portable, and affordable; they can be easily replaced if damaged or lost. They can ensure quick access to current information (for example, about patients, schedules, maps of buildings etc.) from a server when and where the user needs it.

Pinelle and Gutwin (2001, 2002) carried out a study of the information and collaboration needs of homecare workers in the Saskatoon District Health Division (SDHD). They found that it is difficult for home care workers to:

- *coordinate their schedules* so that unwanted conflicts are avoided, and so that desired meetings are possible,
- *disseminate information* to other members of the treatment team, .
- *obtain needed information* from other group members in a timely fashion.
- *arrange treatments* with each other so that the treatments are complimentary.
- formulate *shared treatment goals and care plans* for particular patients.

As a main solution, Pinelle and Gutwin suggest:

> *"the introduction of a point-of-care CIS (clinical information system) in home care that provides access to some form of shared patient record would in itself provide a significant improvement in collaboration,*

provided the CIS allows clinicians to make entries about their daily treatments. Daily entries that document treatments would give an indication of the actions carried out by a healthcare worker and the observations they have made. This allows collaboration implicitly—there is no specific support for direct collaboration between healthcare workers, but the documentation improves information access and fosters awareness of other treatment team members' activities and the outcomes of those activities. "

Accessing such a CIS from a mobile device requires dealing with problems of disconnections, synchronization of write operations and providing an appropriate interface. Our approach addresses these issues.

4. Overview of the Approach

Medical workers in hospitals have typical tasks with standard information needs (Vassileva, 1996). The tasks and rights to access to information are specific to the profession and rank of the user, e.g. doctor, nurse, etc. The homecare domain has also these features, since most of the homecare workers are medical workers and there is a growing trend towards standardization of procedures. By applying a user - task model, it is possible to (1) predict what kind of information will be needed by the homecare worker using her schedule, (2) pre-fetch and/or adapt information appropriate for the task, and (3) present it according to the user's preferences and the limitations of the mobile device on which the information is going to be viewed. Our approach relies on user, task and context models and is characterized by:

- *The use of agents* to give the users the impression of seamless connection. Agents hide the changes in the network bandwidth from the user by providing the needed information for the users before their information needs arise. For this they use task, user, and context models and adaptation techniques such as prefetching and mobile transaction management.
- *Providing only necessary functionalities and information:* When the context (time, location) and the current task of the user are known, the interface can provide only the functionalities needed by the user to perform the task in the specific context (e.g. appropriate menu options, presentation of information). In this way, the interface can be adapted to the constrained screen of the PDA.

The following assumptions are made in the design of the architecture:

- There is a centralized schedule for all homecare workers; patient and schedule-related data is stored in a centralized repository.
- The mobile devices are resource-rich, e.g., have at least 64 megabytes (MB) RAM, a wireless connection with a bandwidth of at least 19.2 kilobits per second (kbps), and the ability to run third party applications.
- The wireless network coverage in the area has known bandwidth and latency.
- The issues regarding the security of information and system are ignored since they are not the focus of this research.

The first assumption is needed to simplify the problem. Since our goal is not to develop a distributed scheduling system, we assume that there is an existing centralized schedule. We also assume that there is a centralized CIS on a server, which is currently not the case yet; however, the SDHD has taken steps along Pinelle & Gutwin's (2001, 2002) recommendation. The second assumption was made since the memory and CPU capabilities of mobile devices grow quickly and currently such devices have become standard. The third assumption is the strongest. While it is easy to obtain from the wireless network provider a wireless coverage map on a coarse level, it is much harder to obtain detailed coverage maps including individual buildings. Also due to weather, traffic, and other environmental factors, the bandwidth and latency can temporarily vary in an unpredictable way. We decide to ignore such variations and assume that the wireless network coverage is approximately known.

4.1. Adapting to the User: User- and Task- Models

In many domains, including the SDHD, the users can be classified based on their profession. Different professions perform different typical tasks. Each task has information needs that typically do not vary over time. In our case study domain, the homecare workers can be classified as nurses, physiotherapists, occupational therapists, social workers, home aide workers and dieticians (Pinelle & Gutwin, 2002). This classification makes a stereotype user modelling approach possible, where the stereotypes correspond to the professions of the users. Each stereotype contains a hierarchy of tasks typical for the profession.

The individual user model contains a reference to the particular user class (profession); in addition, it can contain features like personal information (name, address, and department), presentation preferences, representation for the rank and experience of the user. The latter features were not used in our case study, which focuses mainly on adapting to the network coverage, but they can be useful in other applications where user- and context-adapted presentation of content is more important.

There are many approaches to developing user-task models, for example through expert interviews (Tam, Maulsby, Puerta, 1998). We did not use such approach, since we had no direct access to homecare workers and were unable to perform task-analysis of their activities. Instead, we defined an example set of "typical" tasks for each homecare worker in an ad-hoc manner using the profession and rank as discriminators.

There are tasks that can be performed by users with different professions, with small differences in the subtasks, information needs or access rights to information. For example, when a nurse and physiotherapist perform the same task, i.e. retrieving a patient's record, the information that is provided to them is different. For example, in Figure 1, if the user is a nurse, s/he has the right to access the patient's medical record while other classes of users (e.g. social aide workers) do not have access to this information. In cases when there is a significant overlap, it makes sense to design a generic task model containing the tasks performed typically by all user classes and the information needed for these tasks including several alternative possible presentations of the information depending on individual user features (preferences, rank, experience) and context parameters (available bandwidth, and possibly other parameters like noise, light, privacy and user occupation level).

An example of a task from the general task model is shown in Figure 1. The task model is designed hierarchically, and each task is decomposed to subtasks. Some of the subtasks are restricted to certain user classes. For the task "Visiting patient" in Figure 1, if the user is a nurse, s/he needs to perform the following three tasks:

- Retrieve the patient's medical record
- Retrieve the patient's personal information
- Enter a report.

However, if the user is a home-health aide, s/he will perform only the latter two subtasks. The lower levels of the task model contain the detailed decompositions for each task and the information needed to perform each subtask. The description of each subtask includes information, such as the users' class and context in which the task can be performed. As can be seen from Figure 1, each successive level in the task model differentiates subtasks based on either user type or context parameters. The changes of task structures, the information needs for tasks and context-dependent presentations are regulated and can be changed only by the system administrator.

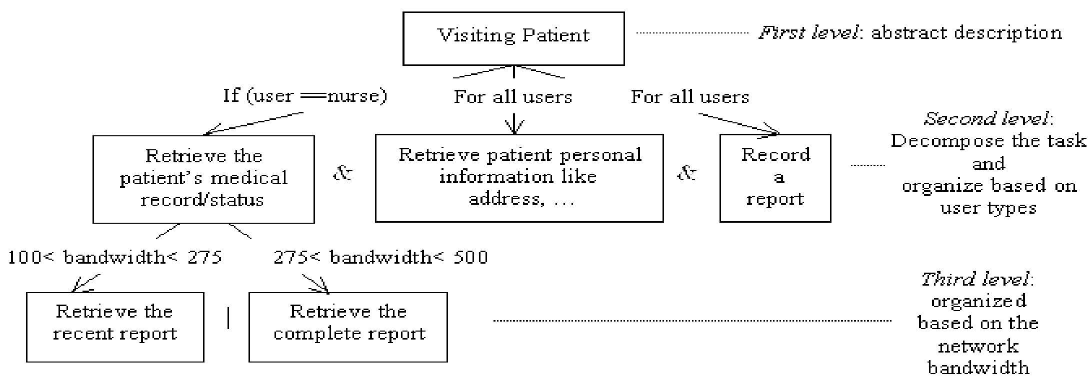


Figure 1: General Model of Task "Visiting Patient"

The user class stereotypes specify "views" over this general task model, i.e. subsets of tasks and information needs specific for the profession and rank defined by the user class (there can be overlaps in the tasks of different user classes). Figure 2 shows the task hierarchy for the user class "Nurse". Therefore if the current user is identified, her corresponding user class task model can be retrieved from the task model together with the possible variants of presenting this information needed for the tasks depending on the current context.

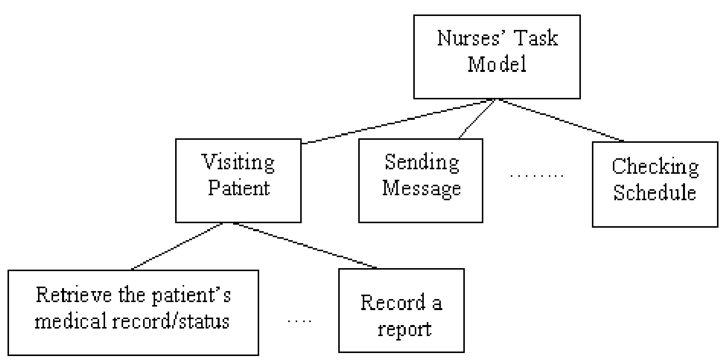


Figure 2: User-Task Model for User Class "Nurse"

The task hierarchy representation for each user class allows a consistent way to design the user interface around specific user tasks. Knowing the information needs of each user task and knowing which tasks the user typically performs allows optimizing the interface layout for the screen size limitations of the mobile device. Different interfaces are designed for the main user classes. To simplify the choice of options that the user sees, only the interface functions related to the homecare worker's current task are shown.

For example, in our domain, the only difference in the interface for a nurse and a physiotherapist is that the nurse has the right of access to the patient's medical record, while the physiotherapist does not have this right, which comes from the task models of the two respective user classes. Therefore, as can be seen in Figure 3 (a), the interfaces for the nurse and physiotherapist contain different sets of buttons.

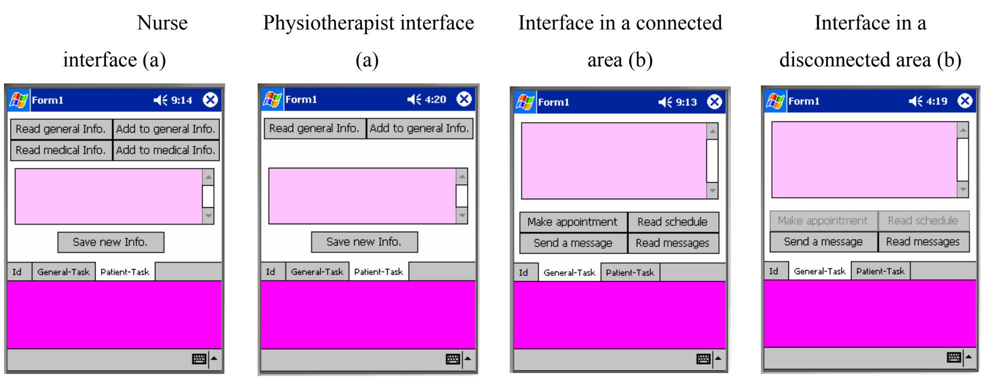


Figure 3: Task (a) and location (b) - adapted interface for user classes "nurse" and "physiotherapist".

The interface can also be adapted also to reflect the tasks that can or cannot be performed at the current time, because of context conditions (available bandwidth), as shown in Figure 3 (b). However, by adapting to the context, we are able to ensure seamless connection and access to the data and do not need to "disable" functionality. How this is achieved, is a topic of the next section.

4.2. Adapting to the Context: Connectivity Map, Location, Time

The context model includes three parts: *computing context*, in our case study, the network bandwidth; *user context*, in our case the user's location; and *time context*, in our case the time of day, week and month. Each part could in principle contain more variables. For example, the computing context could include also the particular type of mobile device used, the screen size, CPU power, memory or hard disk constraints, battery level, etc. The user context could include ambience variables (light, noise), privacy-level at the moment, or whether the user's hands and attention are occupied. Combining these three context parts, each characterised with several parameters can contribute to a good understanding of the user's current situation.

In our case study the computing context is restricted to a connectivity model represented by a map of the network connectivity and bandwidth of the real environment. Each significant location in the real environment (e.g. office building, patient homes, coffee shops where the homecare workers may have informal meetings) is represented as a point on the map. The available bandwidth of each location is stored in the context model and is referenced by the coordinates of the point.

The default value of the bandwidth in the connectivity model is calculated based on the distance of the homecare worker from the network towers. When a homecare worker is close to the network tower the connection is stronger and the bandwidth value is larger. Similarly, when a homecare worker is far from the network tower the connection is weaker and the bandwidth value is smaller. However, there are many exceptions to the default. When a homecare worker is in a disconnected area (such as a concrete building), he/she doesn't have a connection regardless of his/her distance from the tower. Such disconnected areas are represented in the model assuming that they are known in advance.

The context model serves to predict the bandwidth available at the location where the user is going to be in the near future. If the location has good bandwidth according to the model, no adaptation is needed and the user can retrieve the information from the server as s/he needs it for the completion of the task. However, if the bandwidth at the location is low or not existing, the information needed for the completion of the task needs to be pre-fetched. The user-task model is used to find the information that needs to be pre-fetched.

There are two policies for pre-fetching information that can be set as preferences of the user in the individual user model. The first policy is to pre-fetch all the information that is needed in the beginning of the day when the user is in connected area, and later during the day just to send updates. The second policy is to pre-fetch the information as close as possible to the time when it will be needed. By using the first policy the user is guaranteed to have the information when it is needed, but if an unpredicted disconnection occurs, it will not be the newest information. This is acceptable when the homecare worker will be out of connection for a very long time. The second policy will give the newest available information, but in case of unpredicted disconnection, the homecare worker will have no access to the information at all.

4.3. Agents Utilize the User-Task Model and the Context Model to Carry Out the Adaptation

User and proxy agents are used on both the client and the server side to prefetch information and to synchronize access to the CIS when needed. They use data from the user schedule and follow a simple five steps algorithm to prefetch information. Mobile transaction management is used by the agents to synchronize the writing access to the information system when user tasks require it. The next sections describe how the agents put together user-task and context model.

4.3.1 The Schedule

A centralized schedule is created from the individual schedules of the user. It represents the tasks that each user will perform in certain times and locations. Figure 4 shows an example of how the schedule is organized. "Visiting Patient" indicates the task that the two homecare workers will be performing, A and B indicate the respective patients, the location coordinates of their homes on the map are represented, as well as the IDs of the homecare workers, HCW1 and HCW2 respectively.

Date: Aug 10, 2004
Time: 9:00-10:00 -> Visiting Patient A, (location Ax, Ay), HCW1
Time: 9:30-11:30 -> Visiting Patient B, (location Bx, By), HCW2

Figure 4: The centralized schedule

From the information in the schedule – the current patient, the representation of the task's information needs (from the user-task model) and the bandwidth at the location (from the context model) – it can be decided what information needs to be pre-fetched.

4.3.2. Prefetching Algorithm

The process of pre-fetching follows five steps:

(step 1) The current time is used to find the next appointment (next location, task to be performed, patient to be treated) from the user's schedule.

(step 2) The bandwidth at the next location is retrieved from the connectivity model.

(step 3) From the task to be performed and the patient to be treated, the necessary information for the completion of the task is retrieved from the task model.

(step 4) The individual user model is checked for any specific preferences.

(step 5) The necessary information is prefetched / hoarded.

It is clear, that if a user performs tasks that are not included in her/his schedule, the system will not guarantee the availability of information in disconnected condition.

4.3.3. Middleware: Agents, Proxies, Mobile Transaction Management

The application has a client-server architecture represented in Figure 5. The client is a Mobile Device (MD) of a user and the server runs on a desktop computer. To give the users the impression of a seamless connection, agents are used as a middleware layer to hide changes in the network bandwidth from the user by automatically pre-fetching/hoarding the needed information from/to the server ahead of time.

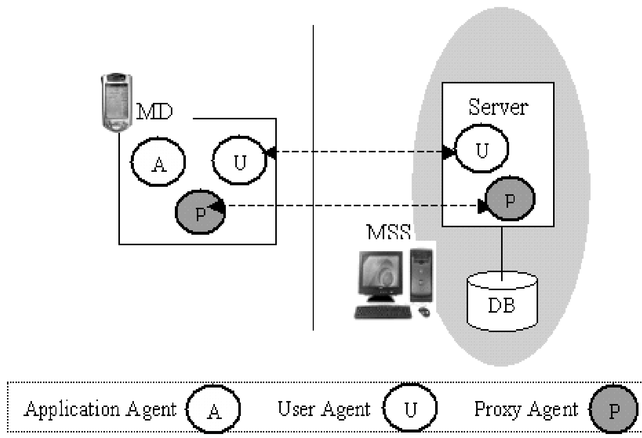


Figure 5: The Application Architecture

A proxy agent and a user agent represent each client (i.e. each user) on the server side. They are implemented in Java using the JADE (2004) platform. The schedule, the user class model, the task model, the context model are kept on the server side and are also implemented in Java.

The client side contains a proxy agent, a user agent, an application agent and the individual user model. Since the client side is implemented in C#, ACL messages (over TCP Sockets) are used to enable the C# to Java communication.

To ensure that the read and write operations are transparent, we use two sets of agents (user agent and proxy agent) and two copies of data, one on the MD and the other on the server (data on the MD is a copy of the data on server). The user agent on the server side looks for any changes and updates in information related to the user's forthcoming tasks (from the schedule) and will prefetch the information according to the policy preferred by the user (from the individual user model).

During disconnection the proxy agent on the server side plays the role of a client for the user agent on the server, while the proxy agent on the MD plays the server role for the user agent on the MD. When the MD is in a disconnected area and there are changes on the server in the information related to the user's forthcoming tasks, the user agent on the server creates a prioritized queue of changes to be sent to the client and sends them to the proxy agent on the server, which plays the role of a client. The proxy agent on the server receives the changes from the user agent on the server, waits until there is connection, and then forwards the messages to the client. The proxy agent performs automatic hoarding to the proxy agent on the client, and the user agent on the client performs automatic reintegration.

If the user executes a write operation while in a disconnected area, the data is queued on the MD by the user agent on the client and sent to the proxy agent on the client, which plays the role of a server in this case. When connection is established, the information will be sent to the proxy agent on the server, which will update the information system on server.

There can be two different kinds of read and write operations, some which require synchronization, e.g. deletion, or entering items in a strict order, and others that do not, e.g. when information is only added. The first kind of operations needs a lock on the centralized information system and the others do not need a lock. When a user reads/writes to the schedule or the agent pre-fetches/updates the schedule, the server side locks the schedule to avoid any inconsistencies, such as the loss of information updates. If the client gets disconnected while reading the schedule, the schedule lock will be broken after a certain amount of time. If the client stays connected or only has a brief disconnection, the schedule stays locked until the client finishes the transaction. For example, when users make changes in the schedule, the schedule is locked and will not be updated until the transaction is committed, since these changes may affect other users. In case of an abort, the client on the MD will be notified. The user agent on the MD always displays the status of the network bandwidth, so the user knows the accuracy of the data that she is using.

An example of read/write operations that do not require lock is the access to patient records, since information cannot be deleted or modified; the users can only add new information to the patient record. In conditions of disconnection, changes originating on the client are queued and forwarded. When several clients modify simultaneously a patient record in conditions of disconnection, at the time of reconnection the proxy agents will negotiate and enter the changes ordered by the time the changes were made on the clients.

On the client side, the user agent and the application agent adapt the interface functionality with respect to the current context and user. The application agent commands the interface layout. The client-side user agent receives the next type of information access to be performed by the homecare worker from the user agent on the server who uses the task model (known from the user class), the connection model, the schedule and a library of cases. Because of the screen size limitation of the PDA, the user agent provides only the suitable functionality for the current task and the homecare worker's user class. When the client-user agent selects the needed functionality, it communicates it to the application agent, and the application agent displays the needed functions on the MD's interface.

5. Evaluation

The main goal of the evaluation is to test two claims about the approach, namely, that it ensures nomadic workers with:

- Seamless connection and
- Seamless access to information

Another goal is to see how dependent is the approach on the availability of correct context information (map of the bandwidth in the area) and what pre-fetching policy is better when there is and when there is no correct context information.

To test the first hypothesis, we evaluate the availability of data needed to perform different tasks at locations with different bandwidth (quality of connection). To test the second hypothesis, we evaluate the staleness of information on the server and MD at the time of performing different tasks at locations with different bandwidth. To test the dependency on correct bandwidth map, we test two setups, one with correct map and one with a faulty map using the two prefetching policies in each setup.

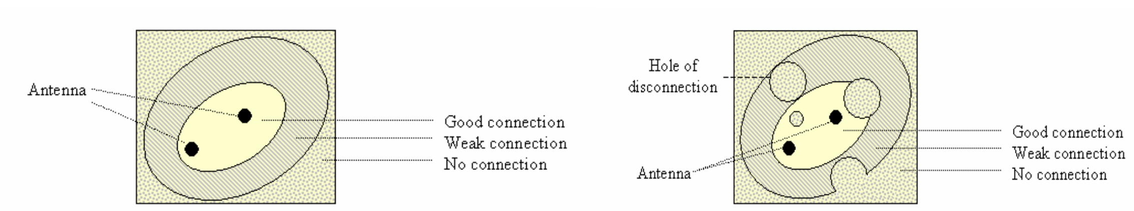
Unfortunately, we were not able to perform an experiment in a real setting, since the system is only a case study. The evaluation is carried out in a simulated environment. The simulation allows to evaluate exactly the features we are interested and to isolate the influence of external random factors.

5.1. Design of the Simulation

A simulated environment has been developed for a hypothetical area with given locations of the homecare offices and patient homes. The values of bandwidth at each point in the environment are simulated and static. The homecare workers start moving to the next location 15 minutes before the next appointment and their movement is modeled as a straight line between the current location and the destination. These simplifications seem reasonable approximations of the real environment.

Four different network maps are used in the simulation. Each network map illustrates the possible range of bandwidth and disconnection areas. The value of the bandwidth is relative with respect to the location of the network tower. If the homecare worker is closer to the network tower she/he has a better connection, unless there are predefined areas of disconnection in the area (e.g., a concrete building close to the tower has no connection). The ranges of bandwidth are modeled as circles with center – the location of a network tower. The areas of disconnection are also modeled as circles around points with no connection (e.g. concrete buildings).

The first two maps shown in Figure 6 differ with respect to the level of precision of the information known about the connection in the area. Map-1 does not contain information about disconnected areas within the generally connected area, i.e. assumes that there are no "holes". Map-2 represents more detailed knowledge about the connectivity, i.e. a better context model. By comparing the adaptation in the same scenario with these two maps we want to see how the quality of the context model influences the adaptation.



a) Map-1: Simple, with two towers

b) Map-2: More realistic, with two towers

Figure 6: Two maps of network coverage with different level of detail.

The locations of the homes of the twelve patients are distributed in the environment in such a way that there are locations in all areas of connection and disconnection. The diamond and square shapes in Figure 7-a) shows the distribution of patients' homes in Map-1 and their available bandwidth. Figure 7-b) shows the distribution of patients' homes in network map from Map-2 and their available bandwidth. It can be seen that two of the patients' home locations which have weak connection according to Map-1 have no connection according to Map-2 since they are in fact in disconnected areas.

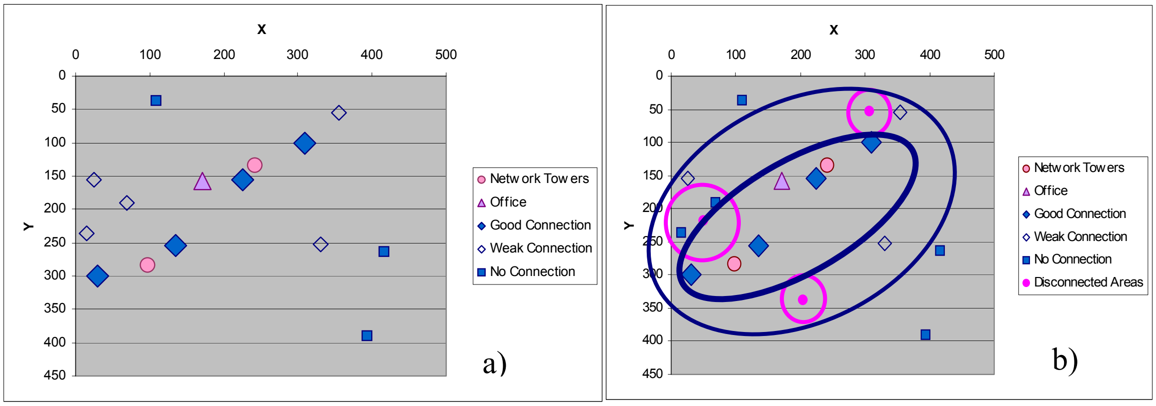


Figure 7: Distribution of Patient Homes in Map-1 and in Map-2.

To study the effect of wrong information in the context model, we introduce also two faulty network maps. Both of them are based on Map-2 (the more detailed map), however, in FaultyMap-1 the client is disconnected while the server believes it is connected. In FaultyMap-2 the client is connected while server believes that the client is disconnected.

Detailed example task models for two user classes – nurse and physiotherapist – have been developed. Our simulation scenario consists of four users (two nurses and two physiotherapists) and twelve patients. One schedule for each user was created manually, so that it contains appointments at each hour from 9:00 until 16:00 in areas with different connection and in entirely disconnected areas. Each appointment involves different tasks from the task model of the corresponding user class.

The simulated users – one nurse and one physiotherapist – move in the simulated environment over the duration of a day according to their schedules and perform various tasks that require access to information and modifying information on the server. The following data is collected on the client side: pre-fetched information items with timestamp and the available bandwidth; old and new values of information items before and after updates from the server with timestamp and the available bandwidth; IDs of tasks performed with timestamp of initiation and the available bandwidth; the status of tasks after they are done (Success/Fail) with timestamp of task completion and the available bandwidth; the result of a task (if it is a Read transaction, the data read by the client, if it is a Write transaction, the data added by the client); the old and new values of local data before and after updates by the client with timestamp.

On the server side the following information is collected: the old and new values of data before and after updates (by the client) with timestamp and the available bandwidth; the data and updates sent to the client with timestamp and the available bandwidth.

To evaluate the seamless connection ensured by the system, we measure the availability of information when the simulated user is at a disconnected location by counting the successfully completed tasks requiring information access during disconnections. The seamless access to information is measured by the staleness in the data (defined as the number of inconsistencies) that the simulated user receives at the moment when the data is needed for the task the user is performing. Inconsistency is defined as a point in time when information is requested and the server and client have different versions. For example, if there is an update on the server in the record of a patient, while the client is disconnected, the client has an older version of the patient's record, and each time such old version of the information is required by a task performed by the user, it is counted as inconsistency. By analysing the data logged on the client side, it is possible to determine the number of failures and successful Read/Write operations. It can be also determined how stale is the information/data provided by comparing the data logged on the client and the server sides.

We measure the overall performance of the system as a combination of the measures for seamless connection and seamless access to information.

$$Performance = \frac{[a-(b+c)]}{a}$$

Where:

a = total number of transactions on the client during the disconnection
b = number of failed transactions due to unavailability of data
c = number of inconsistencies in the data on server and on client

The *Performance* can vary between 1 and –K, where K is the number of all changes that can be made in the data on the CIS during a working day by all the users. If the Performance is equal to 1, there have been no failed transactions and no inconsistencies, so seamless access has been ensured. A positive number for Performance shows that the approach brings some benefit. If all transactions during disconnection have failed due to unavailability of data (i.e. a = b), and there is no data available on the MD left from previous tasks, there cannot be any inconsistencies (c = 0) and as a result *Performance* = 0, which means that the approach has not ensured seamless connection. The *Performance* can also be negative, if there is data left on the MD by chance from previous transactions, which is not accurate anymore. Since the information that is needed for the task at hand, is available, the transaction doesn't fail (i.e. b=0), but since the piece of data is not consistent with the current state on the server, c>0. During a long period of disconnection can be potentially many inconsistencies, more than the number of transactions a. In this case, the Performance is negative. Having a negative number for the Performance, shows that there are updates needed on both client and server side, but due to disconnection of the client the updates are waiting to be performed. This doesn't mean that the approach has failed, since such inconsistencies will be unavoidable in case of prolonged disconnection.

We test the performance for both user classes using all four maps and using each of the two policies for pre-fetching information mentioned in the end of section 4.2.

P1: Pre-fetch all the data at the start of the day. If there is a connection and there are updates, the client pulls the update information continuously.

P2: Pre-fetch the data as close as possible to the time when the data will be used.

5.2. Results

The results of the simulation runs were logged and analysed. Four experiments were made, one with each network map. Each experiment involved two simulated users - a nurse and a physiotherapist following their respective schedules and compared policies 1 and 2 to prefetch information.

Table 1: Simulation results for each experiment

Homecare Worker	System Performance	Number of Transactions During Disconnection	Number of Inconsistencies	Number of Failed Transactions Due to Lack of Data	Policy	Map
Nurse	0.66	3	1	0	P1	Map-1
Physiotherapist	0	2	2	0	P1	Map-1
Nurse	*0.66*	*3*	*1*	*0*	*P2*	*Map-1*
Physiotherapist	*0*	*2*	*2*	*0*	*P2*	*Map-1*
Nurse	0.60	10	4	0	P1	Map-2
Physiotherapist	0.25	4	3	0	P1	Map-2
Nurse	*0.63*	*11*	*4*	*0*	*P2*	*Map-2*
Physiotherapist	*0.25*	*4*	*3*	*0*	*P2*	*Map-2*
Nurse	0.66	12	4	0	P1	Faulty Map-1
Physiotherapist	0.25	4	3	0	P1	Faulty Map-1
Nurse	*0.27*	*11*	*6*	*2*	*P2*	*Faulty Map-1*
Physiotherapist	*-0.33*	*6*	*6*	*2*	*P2*	*Faulty Map-1*
Nurse	0	3	3	0	P1	Faulty Map-2
Physiotherapist	-0.5	2	3	0	P1	Faulty Map-2
Nurse	*0*	*3*	*3*	*0*	*P2*	*Faulty Map-2*
Physiotherapist	*-0.5*	*2*	*3*	*0*	*P2*	*Faulty Map-2*

The results show that the approach generally brings benefit since the performance is mostly positive. There are four cases of Performance = 0, and they are all due to inconsistent data, not to tasks failed because of lack of data. The negative performance in two of the cases is due to the faulty map and the usage of policy 2, which waits to prefetch information in the last moment.

When the server has the correct information regarding the client's location and its connectivity (the first eight rows representing Map-1 and Map-2), the percentage of inconsistencies in data under both policies are the same for each user type. If the information regarding the location and network connectivity is correct there is no difference between the pre-fetching policies. Of course, the generated traffic with the first policy will be much higher. The percentage of failed transactions due to lack of data is the same (zero) under both policies, since the information needed is pre-fetched and available. The difference between the results between the nurse and physiotherapist are due to the different schedule (the physiotherapist happened to have tasks that required write operations during time of disconnection which lead to unavoidable inconsistencies).

When using the FaultyMap-1 (the server-user agent thinks that the client is connected but actually it is not), we observe a difference between the two policies. P2 performs poorly in comparison to P1 because in P2 the information is pre-fetched as close as possible to the time of use. This leads often to unavailability of data because the server believes that the client is still connected, while it is not.

In the case of the FaultyMap-2 (the server-user agent thinks that the client is disconnected while the client has a connection), the performance of the two policies is the same. Because the server-user agent thinks that client is disconnected, it will pre-fetch the information earlier than necessary so there is no failed transaction due to lack of data in this setting.

It is not surprising that in case of unpredicted disconnection P1 performs better, because it guarantees that the information is pre-fetched ahead of time. There is always information available, but it is not necessarily the newest information (if changes happen on the server during the disconnection time, nothing can be done). The trade-off is the increased traffic due to the frequent updates during the time when the client is connected. Depending on the costs of the connection and the precision of the context model the user can choose which policy she prefers her agent to use.

6. Discussion

Our approach ensures adaptation to the context (e.g. current time, user task, and connection availability). Unlike previous approaches, we address the problem of disconnection. Agents are used to pre-fetch information relevant to the user task and context and thus to hide the disconnection from the user and synchronize access to the data.

We do not propose a new approach for user or task modelling, or for context modelling, but rather combine a set of existing techniques. The task, user class and context models are simple, static and defined in advance. There is currently no diagnosis of the context or of the user's features, current task etc. What we believe is interesting in this approach is the combination of the context modelling and user / task modelling through the schedule and the use of agents to adapt the application using the information in these models and methods from mobile transaction management.

Our approach can be easily generalized. Even if bandwidth limitations were not an issue, the same approach could be used to realize context- and user-adapted presentation of content. For example, according to the user study made for context aware mobile communications in hospitals (Muñoz et al., 2003),

> *"Where hospital staff members are at a particular time determines in part the type of information they require. For example, access to a patient's medical records is most relevant when the doctor or nurse is with the patient, so the patient's bed is the best place to display detailed information. A nurse doesn't need to know the appropriate dose of medication until she must give it to the patient. ... Thus, a system that accounts for staff location as part of its design protects against information overload because it ensures that a staff member receives only information that is useful and relevant for that location."*

Using a map of the locations of patients, the agents can easily display the user- and task-related patient information only when the user is at the location of that patient. The interface must be adapted as well to signal to the user which information she can obtain at the given time as it was shown in Figure 3.

Our approach can also be extended to ensure a seamless communication among the homecare workers, which is one of the main problems, identified in the user study of the SDHD by Pinelle and Gutwin (2001, 2002). We currently see three different aspects of achieving such seamless communication:

- Buffering a message send from one user to another; alerting the recipient and displaying it when she is free to read it. Since the personal agent knows the current location and task of the user, as well as her preferences, it can decide whether it would be possible for the user to attend to the message (e.g. the user can be busy, her hands can be greasy, she can be driving, or the privacy level of the message may be inappropriate for viewing at the patient's location). This will free homecare workers from the need to repeatedly page each other and from missing important communications, which is currently the case according to Pinelle and Gutwin's (2002) study.
- Routing context-based messages to appropriate recipient(s). Sometimes, homecare workers need to send warnings about the state of a particular patient to their colleagues who are going to see the patient after them hospitals (Muñoz et al., 2003). They may not know which homecare workers are scheduled to see the patient during the day. By creating a special "location-related messages" part of the information system and adding this information as a task-related information need, the personal agents of users who are scheduled to go to a particular location (or patient) will be able to retrieve these messages and present them to their users, even if they were not sent directly to them.
- Routing role-based messages (or user-class related messages). Homecare workers often need to send warnings and/or messages to other workers of a particular class (e.g. nurse, home-aid etc.) without knowing who exactly is scheduled to visit the patient. Again, agents can help to find the individual user from this class scheduled to visit a particular patient and route the message directly to her.

Our approach for achieving seamless connection and access to data uses two non-critical assumptions:

- the value of bandwidth is static, but in the real world it is prone to fluctuations.
- the speed of the movement of the homecare workers is considered to be constant, but in the real world it varies depending on the time of the day, location, and the heavy traffic.

Knowing the value of bandwidth coverage of the area is needed to create the context model used in our approach. A detailed bandwidth coverage map can usually be obtained from the wireless network provider. Our current context model relies on a static map. However, this is not a serious limitation, since at least in principle changes in the context model can be learned automatically. If the MDs are equipped with GPS and bandwidth monitor applications, they can record times and locations of encountered disconnections and upload this data on a daily basis to a server. The collected data can be used for training a machine-learning algorithm to generate a detailed updated bandwidth map of the area on a daily basis. Even though there will be fluctuations in the bandwidth on a micro-level due to various factors, an approximate coverage map can be created with a fairly good precision. A harder problem still remains - disconnections due to temporary or random factors, such as weather, traffic, building activities at a certain place etc. For example, if the user is in a car parked next to a truck loaded with iron pipes, there may be disconnection, which is due to circumstances. Our approach cannot deal with this kind of disconnections, and the user will have no seamless connection in such cases. However, due to their random and temporary character, we believe that such disconnections will not cause a serious impediment in the users' access to information or prevent them from completing their tasks.

Though it is very hard to predict the exact speed of movement of users, it is possible to determine their exact location at any point of time if the MDs have a GPS. If the destination is assumed to be known from the schedule, the mode of transportation is known and there is a detailed map of the area including streets etc., it would be possible to use scheduling techniques to predict the trajectory and the time when the user will enter a disconnected area, or area with weaker connection. This may be computationally expensive, but it will be done on a server, and can be done off-line (i.e. overnight) with a resulting schedule of prefetching generated for each day.

Our approach also has limitations. In the real world, the users may unexpectedly make changes in their schedules, which are not reflected in the central schedule, for example, take an opportunity to check upon a patient scheduled for the next day, since his/her home is on the way etc. Our approach cannot support adaptation (seamless connection or adaptive content presentation) in such a situation. Also in the real world, a home-care worker may decide to view information that is not supposed to be viewed for the particular task according to the task model. Again, our approach cannot support adaptation, if the user requires information that is not provided in the task model. However, the worst thing that can happen is that the user will have to deal with the disconnection him/herself, i.e. prefetch the data in advance, or ensure synchronization of the device after getting back in connected area. Maintaining always a reminder in the interface that the user is currently in disconnected / connected area and that the information used is pre-fetched and may not be up-to-date will help users know what to expect from the functionality of the system.

Generally, since the adaptation is based on a user, task, and context model, as soon as the user deviates from the expected behaviour, reflected in these models, she can not rely anymore on the adaptation. We do not see a problem with this, as long as the user is aware.

7. Conclusions

The paper presented a combination of techniques from user, task and context modelling, agents and mobile transaction management, to ensure adaptation to one aspect of context - the network connection. A case study of this approach in the domain of homecare work is presented and a simulation is used to evaluate the performance of the system. The results show that this approach provides seamless connection and information access by prefetching information before entering disconnected areas and synchronizing data modified on the mobile device during disconnection.

Even though pre-fetching may not seem as a spectacular idea (if it works perfectly, the effect is the same as that of having continuous network coverage), it is important in practice, since some of the most important obstacles for the usage of MDs is the unreliable network connection.

The approach can be generalized to adapt the content and functionality to the context and user and to create a seamless communication among users. We believe that this approach is appropriate for domains where mobile users have fairly well-specified tasks and information needs.

Acknowledgement. This work has been supported by a Graduate Scholarship from TR-Labs for the first author and by the NSERC Discovery Grants of the other two authors.

References

CDPD (Cellular Digital Packet Data), available on line at (accessed March 28, 2004): http://www2.picante.com/~gtaylor/cdpd.html

Compaq iPaq, available on line at (accessed March 28, 2004): http://h18000.www1.hp.com/products/quickspecs/10977_na/10977_na.HTML

Console, L., Gioria, S., Lombardi, I., Surano, V., Torre, I., 2002. Adaptation and Personalization on Board Cars: A Framework and Its Application to Tourist Services. *Proceedings 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, AH 2002*, 112-121.

JADE, available on line at (accessed March 28, 2004): http://sharon.cselt.it/projects/jade/

Muñoz M., M. Rodríguez, J. Favela, A. Martinez-Garcia, V. González, 2003 Context-Aware Mobile Communication in Hospitals. IEEE Computer 36 (9): 38-46.

Oppermann, R., Specht, M., Jaceniak, I., 1999. Hippie: A Nomadic Information System. Gellersen. W. (Ed.): *Proceedings of the First International Symposium Handheld and Ubiquitous Computing (HUC'99)*, Karlsruhe, September 27 - 29, 330 - 333.

Pinelle, D. and Gutwin, C., 2002. Supporting Collaboration in Multidisciplinary Home Care Teams. *Proceedings of American Medical Informatics Association (AMIA)*

Annual Symposium 2002, San Antonio. Available on line at (accessed March 28, 2004): http://hci.usask.ca/publications/2002/home_care_amia02/index.xml

Pinelle, D. and Gutwin, C., 2001. Collaboration Requirements for Home Care. *Research Report, University of Saskatchewan,* available on line at (accessed March 28, 2004): http://hci.usask.ca/publications/index.xml

Poslad S., Laamanen H., Malaka R., Nick A., Buckle P. and Zipf, A., 2001a. CRUMPET: Creation of User-friendly Mobile Services Personalised for Tourism. *Proceedings of 3G 2001 - Second International Conference on 3G Mobile Communication Technologies.* London, UK, 26-29.

Poslad S., Charlton P. In Marik V., Stepankova O., 2001b. Standardizing agent interoperability: the FIPA approach. *Proceedings of the Multi-Agent Systems & Applications Conference, ACAI-01* (invited paper).

Sadeh, N., Chan, E., Shimazaki, Y., Van, 2002. MyCampus: An Agent-Based Environment for Context-Aware Mobile Services. *In AAMAS02 Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Devices*, Bologna, Italy.

Specht, M., and Oppermann, R., 1999. User Modeling and Adaptivity in Nomadic Information Systems. Proceedings of the 7. *GI-Workshop Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen (ABIS99)*, Universität Magdeburg, September 27 - October 1, 325 -328.

Tam, R.C., Maulsby, D., and Puerta, A.R, 1998. U-TEL: A Tool for Eliciting User Task Models from Domain Experts. *IUI98: International Conference on Intelligent User Interfaces*, San Francisco, January 1998, pp. 77-80. available on line at: http://smi-web.stanford.edu/projects/mecano/publicat.htm (accessed April 1, 2004)

Vassileva, J., 1996. A Task-Centered Approach for User Modeling in a Hypermedia Office Documentation System. *User Modeling and User Adapted Interaction*, 6, 2-3, 185-223.