WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2004; **4**:595–607 (DOI: 10.1002/wcm.231)

Personal and service mobility in ubiquitous computing environments

K. El-Khatib*,†, Zhen E. Zhang, N. Hadibi and G. v. Bochmann

School of Information Technology and Engineering, University of Ottawa, 800 King Edwards, Ottawa, Ontario, Canada K1N 6N5, P.O. Box 450 Stn. A

Summary

Ubiquitous computing environment is defined by the shift of computing technology from the desktop to the background. One of its most notable attributes is its potential to extend the scope of service and personal mobility. This paper describes an agent-based architecture that brings personal and service mobility to the ubiquitous computing environment. A software agent, running on a portable device carried by the user, leverages the existing service discovery protocols to learn about all services available in the vicinity of the user. Short-range wireless technology such as Bluetooth can be used to build a personal area network connecting only devices that are close enough to the user. Acting on behalf of the user and based on a number of aspects, the software agent runs a quality of service (QoS) negotiation and selection algorithm to select the most appropriate available service(s) to be used for a given communication session. The software agent selects as well the configuration parameters for each service. The proposed architecture supports also service hand-off to recompense for service volatility during user movement. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: service mobility; personal mobility; WPAN; ubiquitous environment; software agent; profile; personalization; Quality of Service; multimedia

1. Introduction

Ubiquitous computing is a new trend in computation and communication; it is at the intersection of several technologies, including embedded systems, service discovery, wireless networking and personal computing technologies. It is best described by Mark Weiser, father of ubiquitous computing, as the world with 'invisible' machines; a world such that 'its highest ideal is to make a computer so imbedded, so fitting, so natural, that we use it without even thinking about it' [1]. In such an environment, computing devices are shifted to the background, and they are only visible through the services they provide; specific information about these devices such as location, address or configuration parameters are totally transparent to the user.

One of the major contributing factors to the big interest in ubiquitous computing is the advance in short-range radio frequency communication. This advance has created the notion of personal level communication infrastructure, referred to as wireless personal area networking (WPAN), of which Bluetooth [2] is an example. Devices connected over WPAN have the capability to locate, communicate and provide services for each other. This capability

*Correspondence to: K. El-Khatib, School of Information Technology and Engineering, University of Ottawa, 800 King Edwards, Ottawa, Ontario, Canada K1N 6N5, P.O. Box 450 Stn. A.
†E-mail: Elkhatib@site.uottawa.ca

Contract/grant sponsor: Communications and Information Technology Ontario (CITO).

allows these devices to collaboratively provide an ad hoc distributed computing environment and to deliver services that cannot be possibly delivered with only one device. For instance, a video display service may look for an audio playing service in its vicinity to play an audio and video recording. Also, an audio play-out service can use the service of a computer connected to the internet to download music from the internet before playing it out.

Given these trends in personal communication, there is a growing need to provide personal and service mobility for persons roaming in ubiquitous computing environments. Personal mobility [3] is defined as the ability of a user to get access to telecommunication services from any terminal (e.g. workstations, notebooks, personal digital assistants (PDA), cellular phones) at any time and from any place based on a unique identifier of the user, and the capability of the network to provide services in accordance with the user's service profile. Closely related to the subject of personal mobility is service or session mobility [4], which refers to the possibility of suspending a running service on a device and picking it up on another device at the same point where it was suspended. An example of service mobility is a call transfer from the mobile phone of the user to his office phone.

In this paper, we propose an architecture for supporting personal and service mobility in ubiquitous computing environment. The proposed architecture is an improvement on our personal mobility architecture [5] for classical non-ubiquitous environments. It leverages technologies in short-range wireless communication, such as Bluetooth, to construct the WPAN; it also leverages service discovery protocols, such as Jini, SDP, SLP and Salutation, to discover services available just in the WPAN of the roaming user. The architecture supports also optional service mobility, which allows, when possible, the service currently in use to follow the user as he moves from one location to the other. A major component of our architecture is a personal agent (PA) process running on a personal device carried by the user; the PA triggers the service discovery, service selection and controls service mobility. The PA also enforces the user's policies and preferences stated in the user profile.

Throughout the paper, we will show how the presented architecture could be used during a communication session between Alice, a team manager on a business trip, and her team-members. The elaboration of the scenario is presented in Section 7. Before the meeting, Alice would like to have small chat with Bob, who is a team leader in her group. Using the multimedia workstation in the business office of the hotel, Alice sends an invitation for a multimedia conversation to Bob, 10 min before the meeting starts. Bob, sitting in his office, receives the invitation on his PDA. Since Bob has indicated in his profile that he is always willing to accept calls from Alice, his PDA tries to find a microphone, a speaker, a video display service and a camera to make for a full multimedia session. Assuming that such services exist in Bob's surrounding, the PDA discovers and reserves these services for the communication session with Alice. The PDA sends back information about all these services, and the videoconference is started between Alice and Bob.

When it is time for the meeting, Bob moves with his PDA into the conference room where all the team members are waiting. Bob's PDA detects that the services that Bob was using are not available anymore, and since he has already set the FOLLOW-ME option of the session to ON, his PDA tries to discover similar services to continue the session in the conference room. The PDA then detects and selects the big screen, the camera, the speaker as well as the microphone of the conference room; Alice's picture appears on the big screen and she is now ready to participate in the meeting.

The rest of the paper is organized as follows: Section 2 starts by presenting a literature review of a number of architectures for personal mobility with highlights to their limitations in ubiquitous environments. In Section 3, we discuss how important is content adaptation to supporting personal and service mobility in ubiquitous computing environments and what are the involved aspects. We then propose in Section 4 our architecture for supporting personal mobility in ubiquitous environment and its main components. In Section 5, we present the algorithm used for service and quality of service (QoS) parameter values selection. Section 6 shows how our architecture supports service mobility. Section 7 continues with more details about the usage scenario introduced in Section 1. Our prototype and performance measurements are presented in Section 8. We finally conclude in Section 9.

2. Architectures for Personal and Service Mobility

A number of architectures [5–9] have been proposed to solve the problem of personal mobility in the context of the internet. All these architectures share the same concept of a directory service (DS) that provides a user-customized mapping from a unique user identifier to the device that is best suitable for the user to use. The basic idea of these architectures is that the user keeps a profile in the DS containing static information about all the devices he/she has access to, and the logic or policy (user preferences) for when to use these devices. When the DS receives a session initiation request for a user, it executes the logic in the user's profile and handles the request according to the user's specified preferences.

While these architectures provide personal mobility for users with multiple telecommunication devices (telephone, PDA, laptop, cellular phone, pager), they all fail short to extend personal mobility to ubiquitous computing environments because:

- the information in the DS is static,
- there is no support for service discovery,
- they lack support for service mobility and finally
- they lack support for complex (combined) services.

A number of research works have addressed the problem of service discovery and selection in ubiquitous computing environments. The work in Reference [10] presented two approaches for selecting services based on the physical proximity and line-of-sight of the handheld device relative to the service. The authors in Reference [11] used a central gateway that makes the decision of delegating rendering tasks to devices in the environment of the user. A PDA carried by the user is responsible for detecting the user's nearby devices, and sending the list of available devices to the gateway. These two architectures [10,11] suffer from the drawback of using infrared communication for finding and/or selecting services. Because infrared communication requires aligning the devices before any communication is established, these architectures cannot be used in ubiquitous environments because they require user's awareness of the location of devices. Moreover, service mobility and QoS issues were not discussed in these works. In another work [12], the authors investigated the use of a browser running on a PDA to enable ubiquitous access to local resources as well as resources on the world wide web. The browser, called the Ubicompbrowser, detects devices and resources in the environment of the user, and delegates the rendering of the requested resources to the nearby devices in order to overcome the limitation of the PDA. A major drawback of the Ubicompbrowser is that it requires the user to know its current location to select the rendering devices. Additionally, the Ubicompbrowser does not deal with the issue of QoS negotiation, neither with the issue of service mobility.

In a recent project, researchers at the Smart Space Laboratory (SSLab) [13] have suggested the use of embedded computers as a substitute for the awkward interface of the portable devices. The researchers have suggested that collaboration between a portable device and embedded computers can help alleviate the problem of limited input and output capability on the portable device [14]. They have demonstrated their approach by implementing a mobile TV-phone prototype that uses a nearby high resolution-display instead of the small display on the mobile phone.

Our work is different from all these reviewed architecture in that we dynamically discover and update the list of services available in the WPAN of the user. Our architecture also uses a QoS negotiation and selection algorithm to select the services that best suit the context and preferences of the user. The architecture can also mix-and-match different services to fulfill all the requirements of the session. Additionally, our architecture incorporates additional components in order to support smooth and transparent service mobility.

3. Personalization Aspect of Personal and Service Mobility

Advances in computing technology have led to a wide variety of computing devices, and the miniaturization of these devices has widened the variety even further, and made interconnectivity very difficult. Added to this problem is the diversity of user preferences when it comes to multimedia communications. This diversity in devices and user preferences has made personal and service mobility a challenging task, and required content personalization in order to achieve satisfactory results to the user. Generally speaking, the flexibility of any system to provide content personalization depends mainly on the amount of information available on a number of aspects involved in the delivery of the content to the user. The more information about these aspects made available to the system, the more the content is delivered in a format that is highly satisfactory to the user. These relevant aspects are: user preferences, content profile, network profile, context profile, device profile and the profile of intermediaries along the path of data delivery. We will briefly describe each of these aspects; interested readers might refer to Reference [15] for more details.

3.1. User Profile

The user's profile captures the personal properties and preferences of the user, such as the preferred audio and video receiving/sending qualities (frame rate, resolution, audio quality . . .). Other preferences can also be related to the quality of each media types for communication with an individual person or group of persons. For instance, a customer service representative should be able to specify in his profile the preference to use high-resolution video and CD audio quality when talking to a client, and to use telephony quality audio and low-resolution video when communicating with a colleague at work. The user's profile may also hold the user's policies for application adaptations, such as the preference of the user to drop the audio quality of a sport-clip before degrading the video quality when resources are limited. Some other information in the user profile might include also the user's authorization, authentication and accounting information.

One of the most notable works on user profiles is the MPEG-21 standard [16], which describes attributes of the end user of multimedia content, including besides name and contact information, also content preferences, presentation preferences, accessibility and mobility preferences. These preferences are used for instance to provide effective and efficient access (search, filtering and browsing) to multimedia content.

3.2. Content Profile

Multimedia content might enclose different media types, such as audio, video and text, and each type can have different formats [17]. Each type and format has a number of characteristics and parameters that can be used to describe the media. Such information, referred to as meta-data information, is usually included in the content profile. Some of this meta-data about the content may include:

- information about the storage features of the content, such as the type of media (video, audio etc.), the transport protocol (RTP/UDP/IP, H.320 etc.), and the format (H.261 video, MPEG video etc.);
- information about available variants of the content, such as colored-and-black and white variants;
- information about the author and production of the content, such as the title, and date of creation;
- information related to the usage of the content, such as copyright, application adaptations and usage history.

The MPEG-7 standard [18], formally named 'Multimedia Content Description Interface', offers a comprehensive set of standardized description tools to describe multimedia content. These tools allow for a complete description of what is depicted in the content, the form (coding format and size), the condition for accessing the material, the classification, the context and the links to other relevant material. MPEG-7 provides also tools for describing variations of the content such as summaries and abstracts; scaled, compressed and low-resolution versions; and versions with different languages and modalities—audio, video, image, text and so forth. Using the content profile, a content adaptation system can decide what type of adaptations can be applied to the content.

3.3. Context Profile

The notion of context and its implications has been a research topic for a number of research groups [19–21] and is still attracting more interest. According to References [22,23], the context can be generally defined as: 'any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves'. Based on this definition, a context profile would include any dynamic information that is part of the context or current status of the user. Context information may include the physical (e.g. location, weather, temperature), social (e.g. sitting for dinner) or organizational information (e.g. acting senior manager). Some context information, such as the role or task of the user, can be manually keyed in by the user while other information, such as location, time of the day, weather condition, can be easily gathered using sensing devices. Some other information, such as the current status of the user, can be gathered from other sources such as the calendar of the user or from a meeting attendees list. The MPEG 21 standard includes tools for describing the natural environment characteristics of the user, including location and time, as well as the audio and illumination characteristics of the user's environment. Resource adaptation engines can use these elements to deliver the best experience to the user.

3.4. Device Profile

To ensure that a requested content is properly rendered on the user's device, it is essential to include the capabilities and characteristics of the device into the content personalization process. Information about the rendering device may include the hardware characteristics of the device, such as the device type, the display size, the input and output capabilities. The software characteristics such as the operating system (vendor and version), audio and video codes supported by the device should also be included in the device profile. The User Agent Profile (UAProf) created by the WAP Forum [24] and the MPEG 21 standard [16], both include description tools for describing device capabilities.

3.5. Network Profile

Streaming multimedia content over a network poses a number of technical challenges due to the strict QoS requirements of multimedia contents, such as low delay, low jitter and high throughput [25]. Failing to meet these requirements may lead to a bad experience of the user [26,27]. With a large variety of wired and wireless network connectivity, it is necessary to include the network characteristics into content personalization and to dynamically adapt the multimedia content to the fluctuation in network resources [28]. Achieving this requires collecting information about the available resources in the network, such as the maximum delay, error rate and available throughput on every link over the content delivery path. A description tool for network capabilities, including utilization, delay and error characteristics are included in the MPEG 21 standard.

3.6. Profile of Intermediaries

When the content is delivered to the user across the network, it usually travels over a number of intermediaries. These intermediaries (also referred to as proxies) have been traditionally used to apply some added-value services, including on-the-fly content adaptations services [29–32]. Using intermediaries for applying adaptations alleviates the problem of clients with limited-resources [33] and overloaded server [34]. Additionally, when the content travels through a series of intermediaries, a sequence of intermediaries can be chained together to perform successive transformations on the content. This proves to be useful when adaptations are computationally expensive [35,36] and the adaptation process is a combinatorial process [37]. For instance, trans-coding a 256-color-depth-jpeg image to a two-color-depth-gif image can be done in two stages, 256-color-depth image to a two-color-depth image in one stage, and jpeg image to gif image in another stage, with each stage carried out on a separate intermediary.

For the purpose of content adaptation, the profile of an intermediary would usually include a description of all the adaptation services that an intermediary can provide. These services can be described using any service description language such as JINI [38], SLP [39] or WSDL [40]. The profile would also include information about the available resources at the intermediary (such as CPU cycles, memory) to carry out the services. Note that the available bandwidth through an intermediary can also be included in the intermediary profile, but for clarity reasons, we have decided to include it in the network profile.

4. Proposed Architecture

Our work presented here is inspired by the Ubicomp-browser project, and is intended to support personal mobility in ubiquitous environments. Our architecture builds on our previous architecture for personal mobility [5] and includes additional functionalities to overcome its shortcomings in ubiquitous environments. The modified architecture uses the short-range Bluetooth wireless communication to construct the user's WPAN, and to restrict the domain of services available to the user just to the services running on devices that are within this WPAN. Our architecture differs also from the architecture in References [10,11] in that service selection is done automatically on behalf and according to the preferences of the user, and without requiring the user to point and select each service individually using infrared (since the user might not, and should not, be aware of the services and their locations). We also address the problem of service mobility by using periodical search for services similar[‡] to the services currently used by the user, in order to provide smooth hand-off for these services.

Our previous architecture for personal mobility [5] is based on the concept of a home directory (HD). The HD (same as DS presented in Section 2) has two functionalities: (a) a storage facility for the user's profile and (b) a forwarding service of incoming communication requests. As a storage facility, the HD is a personalized database of users profiles; each

[‡]We say that two services are similar if they serve the same purpose, for instance a TV and a wall projector, or a PC speakers and a mini-stereo.

profile holding the user's contact information, current access information, preferences for call handling, authentication, authorization and accounting information. With this logic stored with the data in the profile, end users can control the access permission to their devices according to their own preferences. During session initiation, the home directory agent (HDA) executes the logic in the user profile every time the user initiates or receives a call request.

In a typical ubiquitous computing environment, the set of available devices for the user may change continuously as the user changes his/her location. Updating manually the information about currently available services is not an option. Additionally, discovering the available services and sending update messages to the HDA is not a practical solution either, since the set of available services might change very often with the environment, which results in many update messages sent to the HDA. Moreover, if the update message incurs a certain delay, the information included in the message could be outdated when it gets to the HDA.

To overcome these limitations, we propose to run a modified version of the HDA on a hand-held device, such as a PDA, that is always carried by the user. We call this modified version of the HDA the PA of the user, and it is responsible for detecting devices in the vicinity of the user as well as managing the user's communication sessions. In order to retrieve the user profile and send/receive communication requests through the HDA, we require that the hand-held device, on which the PA runs, to have access to the internet (through a wireless modem or IEEE 802.11 [41] connection). The PDA is also supposed to be able to join a WPAN (such as Bluetooth WPAN) in order to be able to detect and communicate with other wireless devices just around the user. These requirements are readily available, for instance, in the new iPAQ Pocket PC models from Compaq. For the rest of the paper, we will assume that the PA is running on a PDA that satisfies these communication requirements.

At any one time, either the HDA or the PA is responsible for providing personal mobility service to the user. When the PDA is switched ON, the PA contacts the HDA to retrieve the user profile. From that point on until the PDA is switched OFF, the PA is responsible for executing the logic in the user profile, and the HDA would switch into passive mode and act only as a proxy for incoming call requests. To ensure that the HDA is aware of the status of the PA, we decided to send all replies to communication requests through the HDA. The HDA can detect when the PA is not running or the PDA is currently out of reach if the HDA does not see a reply to a forwarded call after a certain time-out period. The HDA would then switch into active mode, and handle the communication request according to the rules specified in its local copy of the user profile.

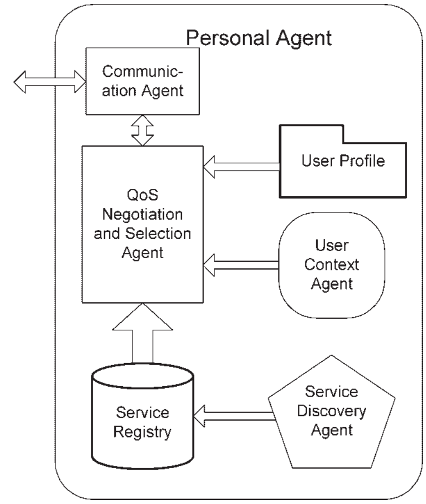

Fig. 1. Components of the personal agent (PA).

To fully account for personal and service mobility, we have designed an architecture for the PA with five major components: a Communication Agent (CA), a Service Discovery Agent (SDA), a User Context Agent (UCA), a QoS Selection and Negotiation Agent (QSNA) and a Service Registry (SR). Figure 1 shows the architecture of the PA with its components. We will present here a detailed description of each of these components. We will also show how the information in all the profiles is gathered and processed in the architecture.

- *Communication Agent:* The CA is responsible for the exchange of communication requests/replies with other parties' communication agents. Communication requests/responses carry usually the content profile, and can also be used to put together the network profile as well as the profiles of all intermediaries along the data path. Each intermediate node that forwards the request/response message adds information about its local available services as well as the network characteristics to which the intermediate node is connected.
- *Service Discovery Agent and Service Registry:* The function of the SDA is to search for all services in the WPAN of the user. Because devices in ubiquitous computing environment are more likely to be single-service devices, the line between a device

and a service becomes blurred. For instance, a display device can be identified by a display service with the same capabilities as the physical device that could be discovered using service discovery protocols. We will refer hereafter to the device that provides a visible service to the user as an end-service, and the rest of the services as intermediary services. Example of an end-service is a video display screen or an audio play-out service.

The SDA provides the QSNA (discussed below) with the list of currently available services (including end-services). Since different services might be using different service discovery protocols (JINI [38], SDP [42] or SLP [39]), the SDA shall act as a service discovery client in multiple service discovery protocols. The SDA periodically searches for all available services, and stores this information in SR. This information allows for fast session setup and smooth service mobility, as we will discuss in Section 6.

- *User Context Agent:* The UCA is responsible for collecting and providing the user's context profile to the QSNA. Context information includes, as we mentioned earlier, the location of the user, whether the user is by herself or surrounded by other people [43,44], and any other context information. The UCA assists the QSNA during the service selection phase by providing up-to-the-minute information about the user's context.
- *QoS Selection and Negotiation Agent:* The function of the QSNA is to select the best services that satisfy the session requirements, and comply with the preferences of the user. This selection is based on the session requirements[§], the information in the user profile, the list of available services found by the SDA, the network and intermediaries profiles, and the user's context provided by the UCA and the QSNA. The QSNA implements the service and QoS parameter values selection algorithm presented in the next section. The QSNA might also mix-and-match several currently available services to satisfy the requirements of the session.

5. Service and QoS Parameter Values Selection Algorithm

To provide personal mobility with personalized content in a ubiquitous computing environment, the system should make the decision regarding (a) the end-service(s) (camera, speakers, microphone ...) to use among the discovered end-services in the vicinity of the user and which meet the requirements of the communication session, and (b) what QoS parameter values (frame rate, frame resolution, audio quality ...) should be used for each end-service, and (c) which intermediate services to use along the data path of the content. These decisions are affected by all the elements we have introduced in Section 4, namely: the user profile, content profile, context profile, end-service profile, intermediaries and network profile. We have devised a selection algorithm that makes these decisions in four steps.

- **Step 1:** The purpose of the first step is to decide on which content is it possible and satisfactory enough for the user to send and receive. This decision is based only on the preferences from the user profile, the user's current context, and within the availability and capability of the end-services in the vicinity of the user. For instance, a user might have to settle for only black and white video output since this is the only available display service close-by. The algorithm merges the user preferences, context information and capabilities of end-services together to form a big set of constraints on what makes an acceptable and satisfactory content.
- **Step 2:** The second step of the selection process represents finding what possible variants of the content could be delivered to the user's end-service. Some content might be initially generated with different variants to meet different requirements, or might be dynamically transformed using a transcoding service on any of the intermediaries. Starting with each variant of the content, the algorithm tries to find what are the possible variants of the content that we can derive, using all possible combinations of adaptation services available at the intermediaries along the path of the data.
- **Step 3:** The third step of the selection process is to merge together the results from the two previous steps: merging what could be delivered to the user, and what the user prefers to receive within the limitation of his context and the end-services. During this step, the algorithm selects the QoS parameter values for each required service, such as the audio quality for an audio service and the video frame rate and resolution for a video service.

  To select the appropriate value for each QoS parameter, we have presented in Reference [8] an extension to the work introduced in Reference [45],

[§]Session requirements may be described using the Session Description Protocol (SDP) carried in the SIP INVITE message.

wherein, the QoS parameters for each service are selected based on a user satisfaction function. The selection algorithm selects the QoS parameters for each service is based on the concept of maximizing the user's satisfaction. In his/her profile, the user indicates a satisfaction function that maps a QoS parameter value to a satisfaction value in the [0 . . . 1] range. The user also assigns a weight value for each media type. The total satisfaction value of the user for a combination of services is based on the weighted combination of his/her satisfaction with the media type and the individual QoS parameter value of the service. Using all possible combinations of QoS parameters of all available services, we select the combination of QoS parameter values that generates the maximum satisfaction (within the restrictions of the devices where the services are running) and the preferences of the user. A detailed description of the work can be found in Reference [5].

In case that the algorithm does not find an appropriate configuration satisfying the user's preferences, the algorithm should either ask the user to release some restrictions on the preferred qualities, or it could present the user with the best configuration found, and the user is asked whether to accept the configuration or just abort the request. A similar approach was introduced in Reference [46].

- **Step 4:** The last step of the selection algorithm is to make sure that the transport requirements of the selected configuration and QoS parameters values do not exceed the currently available network resources described in the network profile. For instance, it might not be possible to deliver a high-resolution video content to a display service connected to a network with currently limited resources. The selection algorithm sorts all possible combinations from Step 3 according to the user's satisfaction, and selects the combination with the highest satisfaction value, and with network resources less than or equal to the currently available network resources.

6. Support for Service Mobility

During a communication session, a nomadic user in a ubiquitous environment might move away from one device and get closer to another device that provides a service similar to the one used on the first device. To continue with the communication session, the user has to re-initiate the session again with the new services. This could be a problem for the user, especially if the user continues moving from one place to the other during the session. Our architecture solves this problem by supporting service mobility during the communication session.

Service mobility is required since the life span of the communication session might be longer than the time during which the currently used device is available in the user's WPAN. To solve this problem, the PA should switch the service from one device to another providing a similar service when the device that is currently used becomes unavailable or it should inform the user about the disappearance of the service. For instance, a user moving away from his computer and entering the conference room should have, if desired, the multimedia session transferred from his computer to the TV and stereo system in the conference room. If the conference room does not have a TV set, the user should be warned that the video display service would be discontinued if he/she stays in the conference room.

Our architecture supports service mobility through service hand-off, transparently to the user. A smooth transparent service handoff requires continuous discovery and update of the SR with the information of the currently available services in order to provide smooth service transfer. When a connection to a service is fading, the SDA informs the QSNA about the possible replacement service. The QSNA passes the information about the new service to CA, which in turn, sends an update message CA of other party.

7. Usage Scenario

In this section, we will elaborate more on the scenario presented in Section 1. We will assume that the SIP [47] signaling protocol is used to establish and maintain the communication session. The scenario of Alice trying to reach Bob, who is in the lounge area, is divided into five phases (Figure 2), with the first phase executed only once, when Bob switches ON his PDA. We will assume that Bob has enabled the service mobility option with his PA. Due to the space limitation, we will only give a short description of each phase.

- *Startup Phase:* The PA retrieves the user's profile from HDA (Messages 1–2).
- *Session Initiation Phase:* Bob's HDA forwards the request to PA (Messages 3–5). The SDA uses the service discovery protocol to discover the available services for the session and update the SR (Messages 6–8). The QSNA selects from the SR the

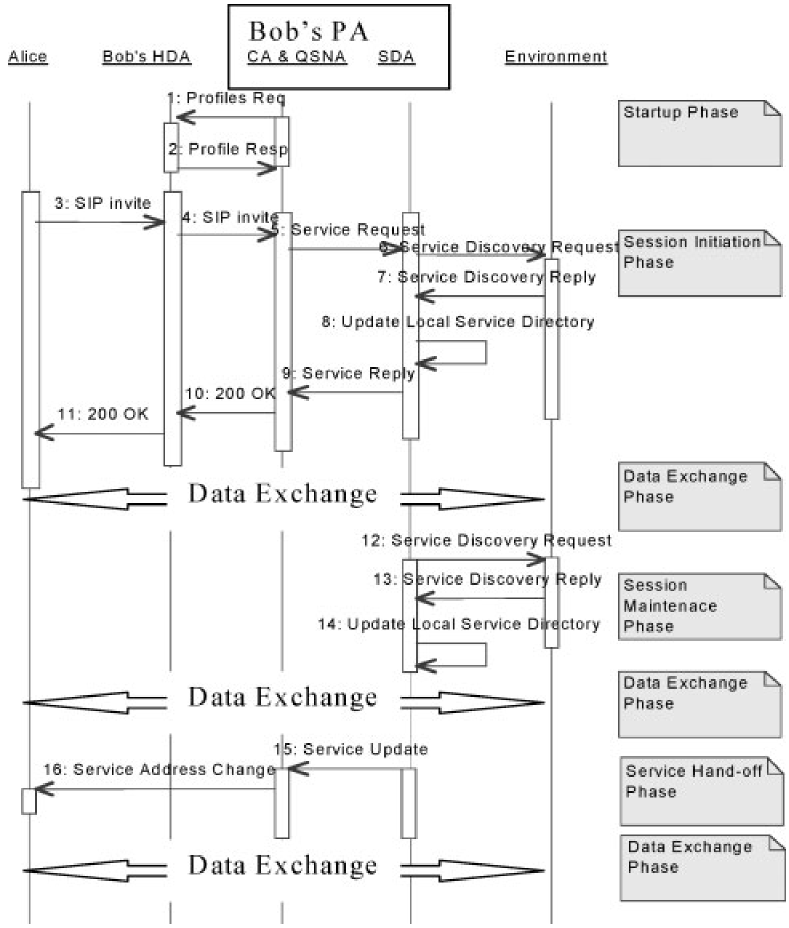

Fig. 2. Session establishment based on the PA.

services for the session based on the session requirements, the user profile, network and intermediaries, and device/service profile, as described in Section 5. The QSNA might also mix-and-match several devices to provide compound services. Bob's PA sends back to Alice's PA the information about the selected services (Messages 9–11).

- *Data Exchange Phase:* The data is exchanged between Alice's device and the selected devices from Bob's environment.
- *Session Maintenance Phase:* As long as the session is still running, the SDA periodically queries the environment for services that are similar to the services used in the session. This information is used to update the SR in order to reduce the delay in service mobility (as we discussed in Section 6). When Bob moves to the conference room, the SDA detects the audio and video services of the conference room (Messages 12–14).
- *Service Hand-off Phase:* In case, a service that is currently used becomes unavailable because of the mobility of the user, the SDA informs the QSNA of the replacement service(s) (in this scenario, the replacement services are the services of the conference room). The QSNA in turns sends an update message through the CA to Alice's PA with the information of the new services (Messages 15–16).

8. Experimentation and Evaluation

In order to get a better understanding of the system, we have developed a prototype of the architecture in our laboratory. We wanted to show how to provide personal and service mobility to the user, and in case of multiple similar services, how the selection algorithm can correctly select the service that is more satisfactory to the user. The prototype also allowed us to measure the performance of the system in order to study the feasibility of the system.

8.1. Experimental Environment

Our experimental environment consisted of a large room, with four services: two audio play-out services and two video display services as shown in Figure 3; Audio Service 2 (AS2) has a better audio quality than Audio Service 1 (AS1), and Video Service 2 (VS2) has a better quality (higher resolution) than Video Service 1 (VS1). We have selected two locations in the room, location 1 and location 2, and the user of the system can move in-between the two locations. While standing in location 1, the user's PDA can discover only AS1 and VS1, and while the user is in location 2, the PDA can discover all four services.

We start the experiment with the user in location 1 when he receives an incoming call through the 802.11b wireless interface of the PDA. The call requires an audio service and a video display service. The PA on the PDA analyses the request and, locates the two services, AS1 and VS1. CA sends a message to the caller's CA containing the information of these services. The caller's media server sends the data directly to these services.

When the user moves to location 2, two better services, AS2 and VS2 also become available to the user. The PA discovers the new services, and based on the user's profile, decides to use the new services since they are more satisfactory to the user. The CA sends an update message to the caller's PA with the information of the new services, and the audio and video are now switched to AS2 and VS2. When the user moves back to location 1, AS2 and VS2 become unavailable again, and the data is sent again to AS1 and VS1.

8.2. Results

After building the prototype, we were able to collect a number of performance metrics. We placed the media server of the caller and the audio and video services on the same local area network (LAN), while the PDA was connected to the LAN through an 802.11b wireless access point. We were interested in measuring the following metrics:

- *Total Signaling Time:* The time it takes for the signaling messages to travel from the caller to the callee and return.
- *Media Server Initialization Time:* The time it takes the CA of the caller to read the reply message from the network and to initialize the Java Media Framework audio and video servers, responsible for capturing and sending the audio and video data.
- *Data Transfer and End-Service Initialization Time:* The time difference between the time when a video frame is sent to the callee, and the time when the frame is actually presented to the callee.

We have perform five run tests, and the average results (in seconds) are presented in Table I.

The data transfer and end-service initialization time included in the table does not include any delay incurred by intermediate adaptation services. We also note that some additional delay (not mentioned in the above table) can be expected due to service

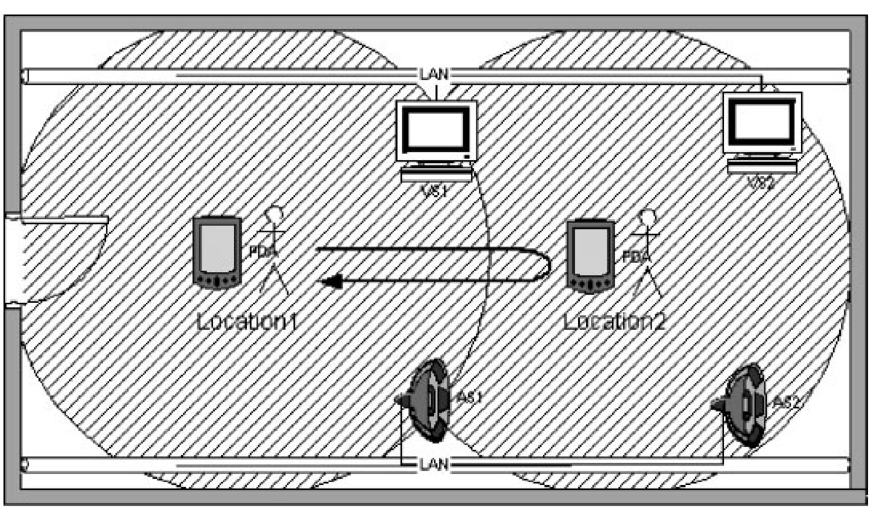

Fig. 3. Experiment environment layout.

Table I. Session setup time.

	Total signaling time	Media server initialization time	Data transfer and end-service initialization time	Total setup time
Time (s)	2.2	5.3	2	9.5

mobility. This delay is a result of the fact that the SDA runs periodically, and not continuously. In our prototype, we set the discovery agent to run periodically every 30 s, to balance between the incurred computation load on the PDA and the speed and accuracy of service discovery. Additionally, the SDA spends 10 s, using the sdptool [48], trying to discover all the available services. In this setting, the sdptool consumes around 18% of the CPU resources when running on the iPAQ 3870 Compaq handheld with 206 MHz RAM processor. As a result, the total service mobility time could vary between 19.5 s (10 s for the service discovery and 9.5 for the signaling and data transfer) if the user reached the new location right before the service discovery agent started the periodic check, compared to 49.5 s if the user moved into the new location right after the discovery agent had finished the service discovery process.

We should mention that for an IP telephony service to be accepted by users, the time delay of the service should be equal if not better than the plain old telephony service. We acknowledge that the measured delays in our prototype are large, but there are a number of ways to improve the performance in a streamlined implementation. The first improvement would be to replace the freeware SDP tool with another faster implementation of the native SDP protocol. Even the native SDP tool suffers from long service discovery delays since each node is obliged to establish a connection with every other node before it can perform service discovery. Some existing work [49] promises to reduce the service discovery time in Bluetooth.

Additionally, we can also improve the data transfer and end-service initialization time. This delay results mainly from the time it takes to initialize and start the end-service. In our prototype, we just register the end-services, but never initialize them. This initialization time could be virtually eliminated if the services are already up and running on the device.

Finally, our current implementation does not take the capabilities (battery, CPU and memory capacity) of the PDA into consideration when executing the code for the PA. Depletion in any of these resources would definitely affect the performance of the service and personal mobility. One could foresee that the service discovery period becomes a function of the available resources. Another approach would be to let the user switch off the service discovery when he/she is not moving. The PA may also run the service discovery agent less frequently when there has not been a recent change in the discovered services.

9. Conclusion

In this paper, we have presented an architecture for supporting personal mobility in ubiquitous environments. The architecture allows nomadic users to benefit from the availability of large number of hidden services in a ubiquitous environment to establish communication sessions. To construct this architecture, we introduced a new component that we called the PA that acts on behalf of the user during the service discovery and selection process. The PA also provides support for service mobility through periodic updates of currently available services into a local service registry. We have also shown the functionality of the PA during a typical communication session using an example scenario. Additionally, we have presented a prototype of the architecture and some performance measurements. We have also discussed a number of recommendations to improve the performance of the prototype. In future work, we are planning to implement the recommendations before we can study the usability of the architecture.

References

1. http://www.ubiq.com/weiser
2. Haartseen J, Naghshineh M, Inouye J. Bluetooth: vision, goals and architecture. *ACM Mobile Computing and Communications Review* 1998; **2**(4): 38–45.
3. Schulzrinne H. Personal mobility for multimedia services in the internet. In *European Workshop on Interactive Distributed Multimedia Systems and Services (IDMS)*, Berlin, Germany, March 1996.
4. Kahane O, Petrack S. *Call Management Agent System: Requirements, Function, Architecture and Protocol. IMTC Voice over IP Forum Submission VOIP97-010*, 1997.
5. He X, El-Khatib K, Bochmann Gv. *A Communication Services Infrastructure Including Home Directory Agents. Technical Report*. University of Ottawa, Canada, May 2000.
6. Anerousis N, Gopalakrishnan R, Kalmanek CR, Kaplan AE, Marshall WT, Mishra PP, Onufryk PZ, Ramakrishnan KK, Sreenan CJ. TOPS: an architecture for telephony over packet networks. *IEEE Journal of Selected Areas in Communications* 1999; **17**(1): 91–108.

7. Roussopoulos M, Maniatis P, Swierk E, Lai K, Appenzeller G, Baker M. Person-level routing in the mobile people architecture. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, October 1999.
8. Kahane O, Petrack S. *Call Management Agent System: Requirements, Function, Architecture and Protocol. IMTC Voice over IP Forum Submission VOIP97-010*, 1997.
9. Wang H, Raman B, Chuah CN, Biswas R, Gummadi R, Hohlt B, Hong X, Kiciman E, Mao Z, Shih JS, Subraimanian L, Zhno BY, Joseph AD, Katz RH. ICEBERG: An Internet-core Network Architecture for Integrated Communications. *IEEE Personal Communications: Special Issue on IP-based Mobile Telecommunication Networks* 2000; **7**(4): 10–19.
10. Barbeau M, Azondekon V, Liscano R. *Service Selection in Mobile Computing Based on Proximity Confirmation Using Infrared. MICON 2002*.
11. Schneider G, Hoymann C, Goose S. Ad-hoc personal ubiquitous multimedia services via UPnP. In *Proceedings of the IEEE International Conference on Multimedia and Exposition*, Tokyo, Japan, August 2001.
12. Beigl M, Schmidt A, Lauff M, Gellersen HW. The Ubicomp-Browser. *Proceedings of the 4th ERCIM Workshop on User Interfaces for All*, October 1998, Stockholm, Sweden.
13. Smart Space Laboratory. http://www.ht.sfc.keio.ac.jp/SSLab/
14. Kohtake N, Matsumiya K, Takashio K, Tokuda H. Smart device collaboration for ubiquitous computing environment. *Workshop of Multi-Device Interface for Ubiquitous Peripheral Interaction, Fifth International Conference on Ubiquitous Computing (UbiComp2003)*, USA, 2003.
15. El-Khatib K, Bochmann Gv. *Profiles in Content Adaptation*. Technical report. University of Ottawa, Canada, December 2003.
16. http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
17. Lei Z, Georganas ND. Context based media adaptation in pervasive computing. In *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2001)*, Toronto, Canada, May 2001.
18. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
19. Ubiquitous Computing (Xerox PARC). http://www.ubiq.com/hypertext/wiser/UbiHome.html
20. Contextual Computing Group (Georgia Tech). http://www.cc.gatech.edu/ccg/
21. Ambiante (Collaborative Buildings) (GMD). www.darmstadt.gmd.de/ambiente
22. Dey AK. *Providing Architectural Support for Building Context-Aware Applications*. Ph.D. thesis, College of Computing, Georgia Institute of Technology, December 2000.
23. Schmidt A, Beigl M, Gellersen H. There is more to context than location. *Computers and Graphics* 1999; **23**(6): 893–902; http://www.elsevier.com
24. *Wireless Application Forum*, http://www.wapforum.org/
25. Quality of Service Extension to IRML. IETF INTERNET-DRAFT. draft-ng-opes-irmlqos-00.txt. July 2001.
26. Katchabaw M, Lutfiyya H, Bauer M. *Driving Resource Management With Application-Level Quality of Service Specifications*. ICE 98, ACM Press 1998, pp. 83–91.
27. Poellabauer C, Abbasi H, Schwan K. Cooperative run-time management of adaptive applications and distributed resources. In *Proceedings of Multimedia'02*, ACM Press 2002.
28. Wu D, Hou YT, Zhang Y. Scalable video coding and transport over broad-band wireless networks. In *Proceedings of IEEE* 2001; **89**(1): 6–20
29. Chandra S, Ellis C, Vahdat A. Application-level differentiated multimedia web services using quality aware transcoding. *IEEE Journal on Selected Areas in Communications* 2000; **18**(12): 2544–2565.
30. Chandra S, Ellis CS. JPEG compression metric as a quality-aware image transcoding. In *Proceedings of USENIX 2nd Symposium on Internet Technology and Systems*, 1999; pp. 81–92.
31. Floyd R, Housel B. Mobile web access using enetwork web express. *IEEE Personal Communications* 1998; **5**(5): 47–52.
32. Fox A, Gribble SD, Chawathe Y, Brewer EA, Gauthier P. Cluster-based scalable network services. In *Proceedings of 16th ACM Symposium on Operating Systems Principles*, Saint-Malo, France, 1997; pp. 78–91.
33. Hollfelder S, Kraiss A, Rakow TC. A client-controlled adaptation framework for multimedia database systems. In *Proceedings of IDMS'97* Steinmetz R, Wolf LC (eds), Springer: Darmstadt, Germany, 1997; 397–409.
34. Mohan R, Smith JR, Li C-S. Adapting multimedia internet content for universal access. *IEEE Transaction on Multimedia* 1999; **1**(1): 104–114.
35. JPEG Compression Metric as a Quality Aware Image Transcoding, Surendar Chandra and Carla Schlatter Ellis. In *Second Usenix Symposium on Internet Technologies and Systems (USITS '99)*, October 12, 1999.
36. Han R, Bhagwat P, LaMaire R, Mummert T, Perret V, Rubas J. Dynamic adaptation in an image transcoding proxy for mobile WWW browsing. *IEEE Personal Communication* 1998; **5**(6): 8–17.
37. Smith JR, Mohan R, Li C-S. Scalable multimedia delivery for pervasive computing. *ACM Multimedia '99*, October 30–November 5, 1999.
38. JINI (TM): http: //java.sun.com/product/JINI/. 1998
39. Guttman E, Perkins C, Veizades J, Day M. Service Location Protocol. Version 2. http://ietf.org/rfc/rfc2608.txt
40. W3C. WSDL: Web Service Description Language, http://www.w3.org/TR/wsdl, 2002.
41. IEEE802.11: http://grouper.ieee.org/groups/802/11/main.html
42. Bluetooth Special Interest Group (SIG): Service Discovery Protocol, SIG Specification version 1.0, Volume 1 Core, part E, pp. 233–384.
43. Beigl M, Schmidt A, Lauff M, Gellersen HW. The Ubicomp-Browser. In *Proceedings of the 4th ERCIM Workshop on User Interfaces for All*, October 1998, Stockholm, Sweden.
44. Horvitz E, Jacobs A, Hovel D. Attention-sensitive alerting. In *Proceedings of UAI '99, Stockholm, Sweden*, July 1999, pp. 305–313. [San Francisco: Morgan Kaufmann.]
45. Richards A, Rogers G, Witana V, Antoniades M. Mapping user level QoS from a single parameter. In *Second IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, Versailles, November 1998.
46. Hafid Abdelhakim, Bochmann Gregor V. An approach to quality of service management in distributed multimedia application: design and an implementation. *Multimedia Tools and Applications* 1999; **9**(2): 167–191.
47. Handley M, Schulzrinne H, Scholler E, Rosenberg J. SIP: Session Initiation Protocol. RFC2543, IETF, March 1999.
48. http://sourceforge.net/
49. Haase M, Sedov I, Preuss S, Cap C, Timmermann D. Time and energy efficient service discovery in Bluetooth. In *Procedings of the 57th IEEE Semiannual Vehicular Technology Conference*, Jeju, Korea, April 2003.

Authors' Biographies

Khalil El-Khatib received his B.S. in computer science from the American University of Beirut (AUB) in 1992. From 1992 to 1994, he worked as a research assistant in the Computer Science Department at AUB. In 1996, he received his M.Sc. in computer science from McGill University. His research topic was Dynamic Load Balancing for Parallel Discrete Event Simulation. In 1996, he joined the High Capacity Division at Nortel Networks as a software designer. After 2 years, he joined the Distributed System Research Group at the University of Ottawa as a Ph.D. candidate under the supervision of Prof. G.v. Bochmann. His research work includes QoS for multimedia applications, personal mobility, IP telephony, feature interaction for VoIP and ubiquitous computing. He joined the National Research Council of Canada in February 2002, as a member of the Network Computing Group, researching into security and privacy issues for the internet and ubiquitous computing environments.

Zhen-Eric Zhang is a master student in computer science program at the School of Information Technology and Engineering at the University of Ottawa, after working for years in telecommunication industry. He did research on wireless network, quality of service, multimedia and information security. His present work is aimed at security support for mobile user access services in ubiquitous environment.

Nabila Hadibi is currently working on her Ph.D. in electrical engineering at the University of Concordia in Montreal. She received her M.Sc. degree from the University of Ottawa, in 2002 and her Engineering degree in Electrical Engineering, from the National Institute of Electronics and Electricity (INELEC), Algeria; in 1999. The current focus of her research, with the collaboration of (LMC) Ericsson Research Canada, is on telecommunication service engineering, IP multimedia services and service composition and personalization. In the past, she did research in the field of wireless networks, service interaction and selection for telecommunication services and voice over IP architectures.

Gregor v. Bochmann is a professor at the School of Information Technology and Engineering at the University of Ottawa since January 1998, after working for 25 years at the University of Montreal. He is a fellow of the IEEE and ACM and a member of the Royal Society of Canada. He did research on programming languages, compiler design, communication protocols and software engineering and published many papers in these areas. He was also actively involved in the standardization of formal description techniques for communication protocols and services. His present work is aimed at methodologies for the design, implementation and testing of communication protocols and distributed systems. Ongoing projects include, quality of service management for distributed multimedia applications and optical networks.

