Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users

Ali Awde¹, Manolo Dulva Hina¹,², Chakib Tadj¹, Amar Ramdane-Cherif², Yacine Bellik³

¹Electrical Engineering Department, Université du Québec, École de technologie superieure
1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Canada
²PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines
45, rue des Etats-Unis, 78035 Versailles Cedex, France
³LIMSI-CNRS, Université de Paris-Sud
B. P. 133, 91403 Orsay, France

{ali.awde.1@ens.etsmtl.ca, manolo-dulva.hina.1@ens.etsmtl.ca, ctadj@ele.etsmtl.ca, rca@prism.uvsq.fr, yacine.bellik@limsi.fr}

Abstract—A multimodal interface allows computing with input and output that best suits the needs of a user, including those with disability. In our multimodal multimedia (MM) computing system for visually-impaired users, the selection of media, modalities and types of applications for activation depends on user’s context and application data. The adaptation of a computing system to the needs of a mobile user is essential in order that the user could continue working on his task at anytime and anywhere, thereby increasing his productivity. Our system is adaptive because the user could access his information anytime and anywhere. This access to user information is made possible through wired and wireless networks. The user profile, user task and data, and the system’s knowledge database (KD) do “follow” the user wherever he goes, hence the system adapts accordingly based on the user’s condition. Our system detects user context and the user’s application data, consults the KD and selects the appropriate media, modalities and types of applications for activation. This work is an original contribution to the ongoing research in helping the visually-impaired users to become autonomous in using the computing system. Our aim is to improve the computing productivity of a visually-impaired user.

I. INTRODUCTION

A multimodal interface in a computing system makes computing more accessible to a wide range of users including those with impairments. Our multimodal multimedia (MM) computing system [1] is aimed at making informatics available to the visually-impaired users. In this system, the user has a task to do, and to accomplish such task, the user would be likely using different applications. For example, a student’s task could be solving his course assignment wherein he might be using different applications such as a web browser, a text editor or a video player, etc. Each data type associated to these applications would need a specific application supplier and a specific media or modality. As the user moves from one environment to another, the computing set-up is likely to change. However, the change in the set-up should not hamper the user from doing what he needs to do – that is, he could still do his task anytime and anywhere he wishes to. This is ubiquitous computing. Our system is designed so that it would correspondingly adapt every application’s supplier and the media/modalities appropriate for the application as the user transfers from one environment to another.

There is a research community that works on improving visually-impaired users’ access to informatics. Some modalities (e.g. Braille, etc.) were designed with them in mind. A walking stick that detects user context [2] and a talking Braille [3] in a building could help in their mobility. Our work is a part of this community; ours allows user and information mobility and accordingly select media, modalities, and types of applications based on user’s condition.

This paper is presented as follows: Related Work is discussed in Section 2; the rational for anytime anywhere computing is presented in Section 3. The details of our ubiquitous computing system are presented in Section 4 while a sample simulation is presented in Section 5. The paper is concluded in Section 6.

II. RELATED WORK

Recently, there have been techniques developed to assist visually-impaired users to make computing more accessible. Speech synthesis and the development of Braille terminal are just some of those techniques. Using a method such as that in [4], the blind people could work independently on a few applications. For accessing data, software such as GUIB [5] and vOICe [6] are able to translate visual information to speech. For web browsing, software like BrailleSurf [7] and WebbIE [8] would convert text data into either speech or Braille. Also mathematical software for the visually impaired such as VICKIE [9] and AudioMath [10] could transfer data expressed in LaTeX or MathML format into another form (i.e. Braille or speech). This collection of software is of great importance to the visually handicapped and hence we use them as applications suppliers in our work.

Previous important works on multimodality related to visually-impaired users include [11], but this could only do one application, and one type of data (i.e. text). What we wish to accomplish is the user doing a task (i.e. several applications, several data types) anytime, anywhere.
In concept, *multimodality* encourages system adaptation to various computing situations and user profiles. Having multimedia and multimodality in the system makes it possible to include devices that could replace those that cannot be used by handicapped users. For example, instead of regular screen that is inappropriate for people who have lost sight, a system could be designed with speech recognition system or with Braille terminal that the user could use.

Most of the literatures in *pervasive computing* [12] sometimes referred to as ubiquitous computing, such as Project Aura [13], are almost dedicated to regular users. The work in [14] also considers the case of visually-impaired users but its system does not consider the conversion of data from one form to another that fits the user’s needs. In contrast, ours is pervasive, does the data conversion to suit the needs of blind users, and considers the non-visual form of data (e.g. Braille).

This work is an important contribution to the ongoing research that finds ways to improve access to informatics among disabled users, including blind users – integrating them to the developments in information technology, making them more autonomous and improving their productivity.

### III. THE NEED FOR ANYTIME, ANYWHERE COMPUTING

Consider a visually-impaired user who works on a task and that no ubiquitous infrastructure is available. Figure 1 depicts such scenario. At home, he would probably have all the resources he needs, such as the Braille keyboard, a speech synthesizer, a microphone, etc. and has all the applications suppliers he wants. Now, consider the same user going into a park to continue working on his task. Since there is no available ubiquitous computing infrastructure, he might as well create his own, that is, constructing the network and brings in his own hardware including the network tower. Since the data has to be made available, a disk/diskette should be always carried by the user wherever he goes. Indeed, it was not long time ago that we were all carrying diskettes from home to work and vice-versa so that we could continue working on a task.

To say the least, the system depicted above is inefficient, impossible and impractical. What is needed is an infrastructure that could realize anytime, anywhere computing.

Figure 2 illustrates a ubiquitous MM computing system for visually-impaired users. The system has components that migrate as the user moves from one computing environment to another. The user logs onto a computer terminal which is connected to a server that shares user information with other members of the server group, all of them connected to each other via wired or wireless communication channel. This makes computing possible anytime and anywhere.

In general, upon logon, the user is connected to a server that is located nearest to his location. On logout, the server would communicate with other server group members so that their own copies of the user information are also updated. Redundancy of data repository is essential in ubiquitous computing because it makes the network system fault-tolerant, that is, a failure in one server is not going to paralyze the whole system because other servers are still functional. In Figure 2, the components (descriptions in next section) of our MM computing system are always present whether the user is at home, at work, in a park, or anywhere the user may be.

### IV. THE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM FOR THE VISUALLY-IMPAIRED USERS

In this section, we provide an overview of the architectural components of our system, as well as the details for the media and modalities selection as a visually-impaired user moves from one environment to another.

#### A. The Architectural Framework

The components that make up the ubiquitous MM computing system for the visually-impaired users are shown in Figure 3. Its main components are as follows:

- **Control Agent (CA)** – it is responsible for the retrieval of user application’s data and files, and of user profile.
- **Application Agent (AA)** – it manages the user’s task by instantiating the applications with appropriate suppliers and quality of service parameters.
• **Environment Agent (EA)** – it determines the user’s context by determining the noise level in the environment and with data type and the context of the user.
• **Converter Agent (CoA)** – it is responsible for conversion, if needed, of data from one form to another (e.g. text to speech).
• **Modality Agent (MA)** – it manages and decides the modalities that are appropriate to use given the application.
• **Device Manager Agent (DMA)** – it selects and activates the media that are suited to user’s needs and context.
• **Machine Learning Agent (MLA)** – it is responsible for machine’s acquisition of knowledge which is based on user scenarios.

As user moves from one computing environment to another, these system components also migrate to continuously manage the user’s computing needs.

---

**B. The User Profile**

To realize ubiquitous computing for the blind, the user profile is needed in order to set-up the computing environment to suite the user needs. Figure 4 demonstrates a sample profile of a certain user (Melanie in this case).

![Fig. 4. An example of a user profile.](image)

In general, the user profile is composed of three major parts:

1. **The user identity** – contains the username, password, and all computers that the user has access.
2. **The application data preferences** – this contains the user’s preferred suppliers for certain applications (e.g. MS Word for text editor) and his preferred quality of service (QoS) parameters values.
3. **The modality data preferences** – this contains preferences of user with regards to the designated modality, such as text to speech, text to Braille, etc.

In concept, a user profile is created for every new user of the computing system. This profile can be modified by the user himself as often as he wishes to. Overall, the system consults the profile to determine the application supplier and QoS parameters that will be used in instantiating a data file, and the configuration of the modality that will be activated for the user.

**C. The User Task**

The term **user task** refers to the work that the user wishes to accomplish in using a computing system. The user would usually utilize several applications to accomplish his task. Figure 5 illustrates a sample user task. The task is composed of the latest files of the last applications the user used. This setting is necessary in instantiating the user’s computing environment so that it would optimally appear the same as that of configuration of his previous computing environment.

![Fig. 5. A sample user task.](image)
using the previous application supplier, using the same application setting, then the outcome appears to the user as if he is just continuing on his interrupted work. In general, computing anytime and anywhere is realized by providing the user with the means to continue working on his applications files in the same format as his previous computing environment.

D. The User File System

In ubiquitous computing, the file system is replicated from the local storage (i.e. cache) to the servers. Figure 7 illustrates the structure of the files stored in a cache. The local machine could be used by many users; hence, every user would have his own storage space in the Documents and Settings directory. While there are $n$ private individual user directories for $n$ different users, there shall be only one directory for the KD. The rationale is because the KD is global and its contents could be adopted by any user.

![Fig. 7. An example of a user file system.](image)

The structure depicted in Figure 7 (Left) is shown as for local machine’s cache (C:\) but the concept of this structure could also be applied to every member server of the server group. In Figure 7 (Right), we showed the contents of the file directory of a user, named “Melanie”. In our system, Melanie’s directory would contain the usual file structure present in Windows XP operating system plus the user’s profile, local settings and user task (a.k.a. application registry). The user task directory is a collection of registry of all applications used by the user. For each application, its registry would contain the supplier of the service, the last file opened in the application, and the features of application. For each application is shown in generic format. This is because every registry for every application is distinct. The settings, for example, of a text editor application are obviously different from the setting of, say, web browser application. In actual implementation, each application registry must be developed to accommodate all features common to every possible supplier of such application.

E. Machine Learning and the A Priori Training Set

In concept, our ubiquitous MM computing system for visually-impaired users detects user context and type of user files (i.e. collectively known as pre-condition scenario), and then determine supplier application and the media and modalities to be activated (a.k.a. post-condition scenario). Since there are finite possible combinations of the pre-condition scenarios and their corresponding post-condition scenarios, creating database records for such relationship makes a lot of sense. These records form the a priori training set of our system. The a priori training set (see Figure 8) is the initial knowledge of the machine.

![Fig. 8. The a priori training set that selects media and modalities and application supplier based on user context and application data.](image)

Initially, our system’s capacity for adaptation to computing environment is based completely on this training set. If the system is to depend only on this training set, there is a possibility of a system stall or crash if a scenario that is not available in the training set arises, such as the needed media or modality is missing or defective. Indeed, to make the system fault-tolerant, incremental learning is necessary. Hence, our system adapts incremental ML in the form of finding a replacement to a missing/defective component. Due to space limitations, the incremental ML system is not presented in this paper. Details of this concept is available in our work in [1].

V. SYSTEM DEMONSTRATION THROUGH SCENARIO SIMULATION
A. The Protocols of Ubiquitous Computing

At home, the user Melanie is connected in a wired network via a wide area network (WAN) topology. There is an internet service provider (ISP), normally a business group (e.g. Videotron in Montreal), that supplies Internet service to private individuals for a fee. Melanie is connected to a router that links her to the ISP. Different branches of ISP are connected to each other; the ISP itself is connected to the rest of the world via WAN. The transmission control protocol (TCP) is responsible for verifying the correct delivery of user data from a sender (i.e. the user or the client) to the receiver (i.e. the server). Data are sent as packets. This is represented as activities (b), (c), and (d) in Figure 9. The Internet protocol (IP) is responsible for moving packets of data from one node to another. In Figure 9, this is demonstrated by activity (T) wherein user information moves from one server to another.

In the park, Melanie is connected in a wireless network via wireless local area network (WLAN) topology. The ISP provides an access point (AP) with which Melanie’s laptop is connected. The laptop itself contains a network interface controller (NIC) which permits wireless communication. Between a server and an AP, data is transferred via IP protocol. Between AP and Melanie’s laptop, data is transferred via TCP protocol. The IEEE 802.11 is the standard adopted for wireless local area network (WLAN). It defines the media access control (MAC) and physical (PHY) layers for a LAN with wireless connectivity. It addresses local area networking where the connected computers communicate over the air to other computers (including server) that are within close proximity to each other. It has two topologies: the ad-hoc network (where almost every computer node could communicate with other nodes, and is not suited for our framework) and the infrastructure network (the configuration that fits our needs). In Figure 9, Melanie’s laptop communicates with an AP which is linked to the server that contains her user information. The AP’s support the MAC and PHY layers of the IEEE 802.11. The rest of the AP device acts as a bridge to convert the 802.11 protocol to MAC and PHY layers of the backbone distribution service (DS) which is typically an IEEE 802.3 Ethernet LAN.

B. A Sample Ubiquitous Computing Scenario

In Figure 9, Melanie leaves home and continues her on task in a park. There, her ubiquitous computing system is connected to a nearby server via an access point. To insure that this user can do her task even during disconnection, our system stores user’s information and system KD in the cache of the local machine. During disconnection, the system continuously tries to detect, reestablish and repair connection with the network and coordinate with its services.

With reference to Figure 9, the meanings of the letters and numbers indicating user activities are given below:

a. Melanie logs out the system and left home for a park. She previously edited a text and navigated a web page.

b. When the system receives a logout signal, it uploads the user’s information (user task, user profile) onto the server.

c. The server copy of KD, if applicable, is updated.
The user’s application registry is all sent to the server.

6. After uploading user information, the system logs off user.

T. The user information and KD are updated on all servers.

1. In the park, Melanie logs into the system via her laptop.
   After identity checking, the system downloads her profile.

2. Now, the system being connected to the distributed data
   downloads user’s task and files (i.e. text, web page).

3. The .txt file, web pages, and previous applications (via task
   registry) are retrieved and loaded onto her computer.

4. (4-a) The application database is used to instantiate the user
   applications. Normally, this step and (4-b) are skipped if the
   user’s computer has the application software. (4-b) A list of
   applications and suppliers for each application available in
   the system is provided.

5. (5-a) The control agent passes user task onto the application
   agent which will instantiate the application. This agent
   selects the appropriate application and supplier for
   accomplishing user task. (5-b) The QoS values of the
   selected applications are supplied to the application agent.

6. (6-a) The user’s computing device is made known to the
   MLA through CA. (6-b) The application used by the AA is
   also conveyed to the MLA. (6-c) The EA passes the user
   context to the MLA, in this scenario; we assume that the
   noise level is high. (6-d) The MA informs the MLA which
   modalities are currently available in the system. In this case,
   speech recognition system and keyboard are all available.

7. The MLA retrieves the previous scenarios (i.e. previous
   knowledge) from the KD. In this case, we assume that there
   is similar scenario found in the KD. Hence, there is no
   new ML training to be made, and the system would simply
   apply the post-condition scenario found in the KD record.

8. The post-condition scenario is to be implemented. With
   reference to the user profile and context, the CA would
   convert text into speech.

9. DMA receives converted data. DMA would then decide
   which available media have to be activated/deactivated.

10. (10-a) Keyboard is activated. (10-b) The headset
    and the speaker will be activated. (10-d) The microphone
    is obviously a no choice in this park due to the high noise
    level in this location, so it will be deactivated.

VI. CONCLUSION

This paper reiterates the assertion that user’s productivity
would increase if it is possible to provide anytime, anywhere
computing. That is exactly what our system is all about. This
paper illustrates the infrastructure of a ubiquitous MM
computing system for the visually-impaired users. The
infrastructure makes user’s profile, data files and the
machine’s knowledge omnipresent, making them accessible to
the user whenever and wherever he goes. We make user’s task
transportable from one computing environment to another –
user application files as well as the application settings are
transported, making computing appear as continuation of the
user’s interrupted work in his previous computing
environment. This system also adapts incremental ML, and as
such the knowledge database of the ML system is also
transported to wherever the user goes. This paper provides the
details of user information access in our ubiquitous MM
computing system and the wireless networking protocols that it
adapts to realize mobile computing and communication.

This work clearly integrates the visually-impaired users into
the mainstream computing as it adapts the user’s media and
modalities and user applications based on user’s context and
preferences. This research work is ongoing, and future works
include the dynamic reconfiguration of the system architecture
to keep the system fault-tolerant when there are cascaded
failures of components.

ACKNOWLEDGMENT

We acknowledge the funding awarded by Natural Sciences
and Engineering Research Council of Canada (NSERC).

REFERENCES

Paradigm of a Pervasive Multimodal Multimedia Computing System for
the Visually-Impaired Users,” presented at the 1st International Conf. on
Grid and Pervasive Computing, Tunghai University, Taiwan, 2006.

Locomotion Assistance Device for the Blind,” presented at ICCHP 2004,
9th International Conference on Computers Helping People with Special
Needs, Université Pierre et Marie Curie, Paris, France.

Computing Network for Orientation and Wayfinding” presented at
Proceedings of the 7th International ACM SIGACCESS Conference on


Handicapped People,” in Conference CSUN. Los Angeles, USA, 1999.


[9] V. Moco and D. Archambault, ”Automatic Conversions of Mathematical
Braille: A Survey of Main Difficulties in Different Languages,” in ICCCHP

[10] H. Ferreira and D. Freitas, ”Enhancing the Accessibility of Math for


Towards Distraction-Free Pervasive Computing,” IEEE Pervasive
Computing, Special Issue on Integrated Pervasive Computing

Incremental Learning Paradigm of a Ubiquitous Multimodal
International Conference on Wireless and Mobile Computing,
Networking and Communications, Montreal, Canada, 2005.