comMotion: a context-aware communication system

Natalia Marmasse

B.S., Computer Science
Empire State College, State University of New York, 1995

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
Massachusetts Institute of Technology
September 1999

Author: [illegible]
Program in Media Arts and Science
28 June 1999

Certified by: [illegible]
Christopher M. Schmandt
Principal Research Scientist, MIT Media Laboratory
Thesis Supervisor

Accepted by: [illegible]
Stephen A. Benton
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

comMotion: a context-aware communication system

Natalia Marmasse

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on June 28, 1999
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

How many times have you gone to the grocery store but left your shopping list on the refrigerator door? Wouldn't it be more efficient to have a reminder to buy groceries and the shopping list delivered to you when you were in the vicinity of the store?

How many times have you suddenly thought of something you must do or have just had a great idea that you want to write down, or record, for future reference? Wouldn't it be useful to record that thought and be reminded of it when in the relevant place and time?

Information delivery utopia would be to receive the exact amount of data we need (no more, no less), when and where we need it, and in a gratifying format, for example, in a pleasant voice or well-structured text. Although we are still far away from this goal, a partial solution to the information overload is to create systems which deliver timely information when the user is in the relevant context.

This thesis describes *comMotion*, a context-aware communication system for a mobile or wearable computing platform. A behaviour-learning agent automatically learns the salient locations in the user's life. Once these places have been determined, location specific to-do lists, electronic Post-it notes or reminders, and subscription requested information can be delivered to the user at the right place and time. The user interacts with the system through a visual and/or speech interface.

Thesis Supervisor: Christopher M. Schmandt
Title: Principal Research Scientist, MIT Media Laboratory

Thesis Committee

Thesis Supervisor: [illegible]

Christopher Schmandt
Principal Research Scientist
MIT Media Laboratory

Thesis Reader: [illegible]

Dr. Rosalind W. Picard
Associate Professor of Media Arts and Sciences
MIT Media Laboratory

Thesis Reader: [illegible]

Dr. Henry Lieberman
Research Scientist
MIT Media Laboratory

Preface

Faithful to the essence of my project, I would like to reach people from different contexts. I have tried to explain my ideas simply and comprehensively with the hope that this document will be read by a large community of colleagues and friends from different backgrounds.

People often give an extensive list of all colleagues, personal friends and family they acknowledge. The important people in my private life know who they are. Therefore, I will limit myself to acknowledging only people directly related to this thesis work.

First and foremost I thank my advisor Chris Schmandt who provided me with the opportunity and the environment in which to carry out my research. I thank him also for understanding why I had to be part of the Everest project, although this meant investing most of a semester in work unrelated to our group. At least he can boast having received e-mail from Everest base-camp.

I also wish to thank: Roz Picard and Henry Lieberman for being readers and for their comments and feedback; my colleagues from the Speech Interface Group, Keith Emnett, Stefan Marti, Nitin Sawhney and Sean Wheeler, for support, discussions and feedback; Brad Rhodes for discussions on context-aware issues; Nuria Oliver for discussions and feedback during the development of this project.

And last but not least, a big hug to Isabelle Marmasse my devoted editor.

Natalia Marmasse

June, 1999

Table of Contents

Table of Figures

* A colour copy of this figure is included in Appendix A

1. Introduction

1.1 Motivation

How many times have you gone to the grocery store but left your shopping list on the refrigerator door? Wouldn't it be more efficient to have a reminder to buy groceries and the shopping list delivered to you when you were in the vicinity of the store?

How many times have you suddenly thought of something you must do or have just had a great idea that you want to write down, or record, for future reference but were unable to? Wouldn't it be useful to record that thought and be reminded of it when in the relevant place and time?

Personal computers have not really lived up to their name: most of them sit on desktops where their interaction with the user is limited to a small part of the day. Over recent years, the trend has been to make smaller and faster PCs in the form of laptops and Personal Digital Assistants (PDAs), consequently making computers more mobile. The main function of the PDAs has been day-planners, notebooks and address books.

Moreover, we live in a world in which the information overload is part of our daily life. Information delivery utopia would be to receive the exact amount of data we need (no more, no less), when and where we need it, and in a gratifying format, for example, in a pleasant voice or well-structured text. Although we are still far away from this goal, a partial solution to the overload is to create systems which deliver timely information when the user is in the relevant context [1], taking into account his mobility.

The ideal personal device would understand you, that is, have a model of you. It would know your context from unobtrusive monitoring sensors and it could combine this knowledge to make the right decisions and present you with required information. You could communicate with it via a multi-modal interface but its dialogue would be context sensitive, for example, choosing speech if it knew you were in a hands and/or eyes busy situation.

Imagine you had a device which always knew where you were and learned about the important locations in your life. Imagine a device which would not only provide specific information upon request, but also proactively remind you of items on your relevant to-do list, pending electronic reminders or other associated information. This data might be text or recorded audio, the output of which could be visual and/or speech. Not only would you receive the appropriate information where needed but you could also type or record reminders and associate them with specific locations.

1.2 System overview

comMotion is a context-aware, mobile or wearable (depending on the hardware used) communication system. Emphasis is put on the mobility of the user and his context rather than on the mobility of the computer. Knowing where people are is the key to building location-aware applications. The ideal sensor for such a system would provide location information seamlessly both in- and outdoors. Currently such a sensor is not available, at least not at an affordable price. Therefore this version of *comMotion* focuses exclusively on outdoor tracking.

A behaviour-learning agent, using data from the Global Positioning System, tracks the user and progressively learns the frequented locations in his life. Once a salient location has been identified, the user only has to name it or indicate that it is to be ignored. Naming or defining a place converts it into a virtual location, for instance "home", as opposed to its physical, geographic, coordinates. No boot-strapping is required; the agent monitors the user and progressively learns on its own, hence relieving him from the tedious task of classifying data or filling out preliminary forms.

A to-do list is associated with each defined virtual location. These lists resemble their real world paper parallel, on which items are listed and can be ticked off once they have been done. However, in *comMotion*, the to-do items can be either text or recorded audio. When the user is in the relevant location, he will hear an auditory cue alerting him that he has items on the associated to-do list. Moreover, other users can also send him reminders to his virtual locations. These reminders resemble the common 3M Posts-its™ and can be sent via the regular e-mail system.

In addition, the user can subscribe to different information services, such as headline news, weather reports and current movie listings. Since requested information might vary depending on context, the subscription is per location and different schedules can be made for different days. For example, the user could request to receive a list of the movies showing at the local cinemas when leaving work on Fridays. In addition, *comMotion* can provide maps showing the user's current position together with neighbourhood locales, such as banks, movie theatres, grocery stores, etc.

comMotion has a multi-modal interface. The speech interface does not replace the visual one, but rather it complements it, enabling access to the main functions when the user is in a hands and/or eyes busy situation, such as driving.

comMotion exploits location, the main feature of mobility, and provides a high degree of timely and context-related information delivery, hence it also serves as a memory prosthesis. Once it has learned the important virtual locations, such as "home" and "work", it uses this knowledge to trigger relevant information to the user, in the form of to-do lists, reminders or virtual Post-it notes, as well as Web content information the user has subscribed to.

1.3 The plot thickens

Dramatis personae: typical family with Father, Mother and Child.

Father: has the car version of *comMotion*. He goes everywhere by car so when he enters his vehicle he just docks his mobile computer to the car panel and uses the built-in GPS. Ever since he has been using *comMotion* he is more time-efficient and he cannot remember when he last said: "I forgot to...". He used to receive e-mail from his wife asking him to pick up milk on the way home, however, these messages would interrupt what he was doing and by the time he was ready to go home he had totally forgotten the request. Now she just sends the reminder to be delivered to him at the grocery store. He goes past the shop on his way home anyhow, and the reminders are very effective. He once programmed the system to have a traffic report delivered when he leaves home and he has been using this service ever since. It has been very helpful in choosing which route to take on his way to work. In fact, he also likes having the headline news read to him while driving.

Mother: holds a job, runs the household, and reads poetry once a week at the literary club, so she has a very busy schedule and must have a grip on things. She has no time to forget stuff. She uses the full-fledged mobile version of *comMotion*.

Child: has the kid-*comMotion* version. After school he is allowed to go to any of his many friends, as long as he sends Mom a reminder (to be delivered immediately wherever she is). He is an absent-minded child and he often forgets, however Mom can easily query his whereabouts –she no longer needs to make ten phone calls to find out where he is. Mom also once programmed a reminder for him to be back home by seven o'clock in order to have dinner and do his homework. He gets this reminder every day since, when programming the reminder, Mom indicated to the system that this message was to be repeated daily.

The family is moving house. They will be staying within the same general area but far enough to change local grocery store, post-office, and so on. How will this affect them?

As for the Mother, after only three visits to new sites (grocery store, post-office, etc.) the system will learn the new physical locations and let her associate the old to-do lists with them. Moreover, she will be changing literary club and this is very import to her. Typically her friends send her snippets of book reviews by e-mail, which *comMotion* forwards to her at the club in the form of reminders (Post-its). She receives and reads these upon arriving at the club and hence they are fresh in her memory. She would definitely not want to miss any reviews because she likes to have some notion of the books people are commenting even if she has not found time to read them. Since this is very important to her, she teaches the system the new physical location of the club the very first time she goes there. She can do this easily by simply pushing a "here" button and *comMotion* will remember it from then on.

The Father will be keeping the same job, and the grocery store will also remain the same since it is conveniently close to his office. However, his other important locations, such as the soccer club, will be learned after three visits, therefore he is not concerned. What's more he has had changes in his life before and was amazed how easily *comMotion* adapted. He is sure it will deal with it well again, simply learning the new physical locations and, once he has named them, the system will associate all his old data and location preferences to the new physical site. For example, once *comMotion* has learnt (or been taught) the location of the new house, it will automatically deliver the traffic reports when he leaves home on weekdays, as well as the headline news. Therefore, he will not need to re-program these.

The Child will go to the same school and keep the same friends so he will not have any problems, and his Mother will not have to re-program anything either.

N.B.: Not all the features here described currently exist in the prototype (*cfr.* section 3.6).

1.4 Terminology

1.4.1 Context-aware, mobile, ubiquitous and wearable computing

Context-awareness is a term which describes the ability of the computer to sense and act upon information regarding its environment, such as location, time, temperature, user identity or user activity.

The context is the group of features of the environment not created explicitly to be input to the system. A context-aware application uses the context to perform something useful.[1]

Whether an application is context-aware depends on the user's motivation. For example, suppose a user lives in a "smart-house" that turns the heat down whenever there is nobody home. If the user were to leave home to go buy something, his action (leaving) would not explicitly be meant as input for the system, therefore it would be context-aware. However, if the user left the house in order to explicitly trigger the thermostat, then that input would no longer be context, but rather direct input.

Mobile computing is the area of technology which aims to provide computing resources in a mobile environment. It enables a user to travel from place to place and still have computing facilities at the new location, or while on the move.

Ubiquitous computing is a term closely related to context-awareness and mobile computing: the idea is to provide transparent access to computer resources where and when they are needed via devices embedded in the user's environment [2]. As Weiser stated, our goal should be to have personal relationships and not personal computers; the computing should be put in the wall paint.

Wearable computing [3] is a term which reflects the belief that mobile computers should be taken a step further: they should not be carried but rather worn, just like a watch or clothes are worn. Wearable computers should interact with the user based on the context of the situation. Ideally this type of computer should:

[1] Definition attained after a long discussion of the MIT Media Lab context-aware group.

- be highly portable;
- enable hands-free use;
- have sensors to learn about its environment;
- know how to get the user's attention even when it is not actively being used, and
- always be on.

Both wearable and ubiquitous computing seek to increase human capabilities with computers. Wearable computing tries to do so by directly augmenting the user, whereas ubiquitous computing augments the surrounding physical environment with a network of machines and sensors.

1.4.2 Global Positioning System

The best system for outdoor location sensing is the Global Positioning System (GPS). GPS is a satellite-based radio-navigation system developed and operated by the U.S. Department of Defense [4]. It consists of a constellation of 24 satellites (six planes of four satellites) which permit users to determine their three-dimensional position (latitude, longitude, altitude), velocity and time, anywhere on the planet. The civil GPS system has a predicted accuracy of less than 100 metres, however, more precise positioning can be obtained with Differential GPS (DGPS), which typically provides a two-metre accuracy. In urban areas, the use of GPS systems may be limited since the signals can be shadowed by buildings, in the so-called concrete canyons.

The idea behind GPS is to use satellites in space as reference points for location on Earth and to triangulate from them. Knowing the distance from a single satellite confines location to the surface of a sphere that is centered around that satellite. Measuring the distance to a second satellite delimits position as a circle: the intersection of the two spheres (the perimeter of the striped area in Figure 1.1). Measurement from a third satellite narrows the position down to two points where the third sphere cuts the circle that is the intersection of the first two spheres. So from three satellites it is possible to narrow a position down to just two points in space (Figure 1.1). To determine which is the true location, a fourth measurement can be taken, however, one of the two points is usually a ridiculous answer (either too far from Earth or an impossible velocity) and can be rejected without another measurement. To triangulate, a GPS receiver measures distance using the travel time of radio signal (the time traveled by each signal describes a sphere around the satellite), and very accurate timing is required. Besides distance, it is necessary to know exactly where the satellites are in space.

This is achieved by high orbits and careful monitoring, though it is also necessary to correct the delays the signals experience as they travel through the atmosphere.

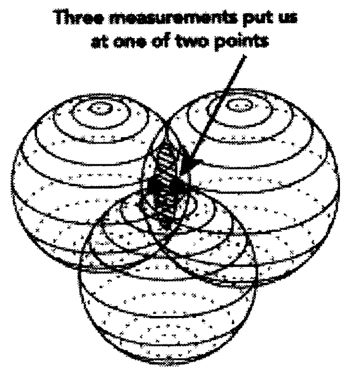


Figure 1.1: GPS triangulation from three satellites, showing how position is narrowed down to two points

In DGPS a reference station at a known location is used. This fixed base station receives the same GPS signals as the mobile receiver but instead of using timing signals to calculate its position, it uses its known position to calculate timing. It computes what the travel time of GPS signals should be and compares this with what they are actually, and henceforth calculates the error. Subsequently, the inaccuracy information can be broadcast to other GPS receivers in the area so they can correct their measurements in real time. If two receivers are within a few hundred kilometres of each other, the signals that reach both of them will have traveled through virtually the same slice of atmosphere, and so will have virtually the same errors. DGPS can provide a position with an accuracy of less than two metres.

Latitude is expressed in degrees, minutes and seconds and an N or S to indicate North or South of the Equator. Longitude is also expressed in degrees, minutes and seconds and an E or W to indicate East or West of the Prime Meridian in Greenwich, England. One degree of latitude is roughly equivalent to 111,000 metres. One degree of longitude, in metres, equals the cosine of the latitude times 111,000.

1.4.3 Cellular Digital Packet Data

Cellular Digital Packet Data (CDPD) is a packet data protocol designed to work over Advanced Mobile Phone Service (AMPS –the standard system for analog signal cellular telephone service) or as a protocol for Time Division Multiple Access (TDMA –a digital air interface technology used in cellular and personal communications services). Software on the client's portable PC transforms data into standard Internet protocol packets before transmitting them to the cellular system where they are inserted into the space between voice conversations on the cellular network. Currently the transmission rate is at best 19.2 Kbps, and often only 10 Kbps.

1.4.4 Global System for Mobile communication

Global System for Mobile communication (GSM) is a digital mobile telephone system which is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1800 MHz frequency band. The transmission rate is 9.6 Kbps and requires dial-up.

1.5 Speech technology

1.5.1 Speech recognition

Speech recognition allows computers to accept speech as an input medium. The content of an utterance can be transcribed (for machine dictation) or interpreted (for speech understanding) to perform some action. What makes speech recognition difficult is the wide acoustic and linguistic variability among different people, therefore, current technology has several limitations. There are *speaker-dependent* systems that need to be acoustically trained to a new user while other systems are *speaker-independent*. Some systems accept *continuous speech* whereas others require the user to make a small pause between each word –known as *discrete word recognition*. All prevailing systems work with limited vocabularies, that is, from a few thousand to tens of thousands of words (only for machine dictation), and on specific domains, for example, commercial letters, medical reporting, air-traffic control, etcetera.

Modern systems are based on a search algorithm which looks for the most probable word sequence that matches with a suitable parametric representation of the input speech signal (Figure 1.2). The search is performed on a probabilistic graph which represents all possible word sequences and their corresponding phonetic transcriptions. Probabilistic matching with the input signal is made at the level of phonemes by means of a statistical acoustic model, which is capable of modeling both duration and frequency variations of each phoneme. The acoustic model is trained on a large sample of speech recordings. The search graph is initially built from a dictionary and a statistical language model, and is then trained on a text body representing the application language.

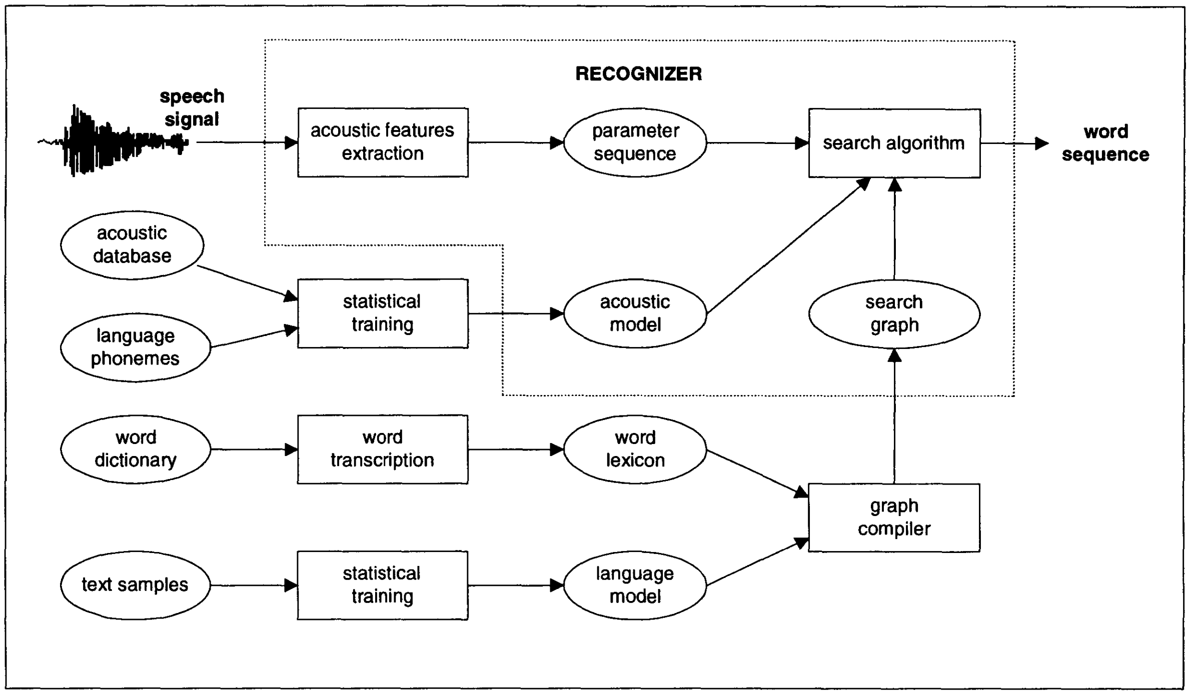


Figure 1.2: Architecture of a speech recognizer

1.5.2 Speech synthesis

Speech synthesis programs convert written input to spoken output by automatically generating synthetic speech. Speech synthesis is often referred to as Text-to-Speech or TTS.

As there are several algorithms, the choice will depend on the task they are going to be used for. The easiest way is to simply record a person's voice uttering the desired phrases. However, this method is only effective if a restricted volume of phrases and sentences is used, for instance, messages in a train station, or schedule information via phone. Needless to say, quality depends on the way the recording was done. Nevertheless, this method is inadequate for free-form text.

There are also algorithms, more sophisticated but inferior in quality, which split the speech into smaller pieces. The smaller the units, the fewer required, but the quality also decreases. A frequently used factor is the phoneme: the smallest linguistic unit. Depending on the language, there are about

35-50 phonemes in western European languages, that is, there will be 35-50 single recordings. Since fluent speech requires smooth transitions between the elements, the problem lies in combining them. If there are few recordings the intelligibility will be low, but the memory required will be small. A solution to this problem is to use diphones, that is, instead of splitting at the transitions, the cut is made in the middle of the phonemes, thus leaving the transitions themselves intact. This produces about 400 elements (20*20) and the quality increases. The longer the units, the more elements there will be, but quality increases along with the memory required. Other units which are widely used are: half-syllables, syllables, words or combinations of them; for example, word stems and inflectional endings. This method of lining is known as concatenative synthesis.

An alternative to the concatenating of digitized speech units is the parametric or formant synthesis method. In this case, speech is generated by modifying parameters that control a model of the vocal tract, and by doing so, over time, speech-like sounds are created. The vocal tract (the throat from the vocal cords to the lips) has certain resonant frequencies which change as the configuration of the vocal tract changes. The resonant peaks in the vocal tract transfer function (frequency response) are called "formants", and it is by the formant positions that the ear is able to differentiate one speech sound from another. Parametric synthesizers mimic the speech production mechanism using digital oscillators, noise sources, and filters (formant resonators); similar to electronic music synthesizers.

1.5.3 Speech technology used in *comMotion*

comMotion has a Graphical User Interface (GUI) as well as a speech interface. An important consideration when defining the user interaction is the type of speech technology which will be integrated: discrete word recognition versus continuous speech; speaker-dependence versus speaker-independence; user activated speech recognition (by means of a push-to-talk button) or continuous speech recognition. Moreover, issues of performance, operating system compatibility and ease of integration must also be contemplated.

comMotion uses a combination of speech technologies. Speech recognition is used to accept speech commands as an input medium. Speech synthesis is employed to convert textual input, such as shopping lists, to spoken output. In order to avoid the need for extensive training and provide the user with natural spoken interaction, a continuous speech, speaker-independent recognizer was chosen.

Speech commands in *comMotion* are used to :

- navigate between different to-do lists or within the items of a specific list, regardless of whether these items are text or recorded audio;
- record an audio item for a specific to-do list which can later be heard as recorded audio (the system does not have dictation capabilities, therefore, this item will not be seen as text in the GUI, but rather it will appear as a button to play the audio file);
- navigate between the reminders (electronic Post-it notes) of different locations;
- browse the reminders of a specific location.

Speech is also used in auditory cues, as an alerting mechanism, and in audio playback. For example, as the user approaches his grocery store, the following dialogue (Table 1) may take place.

Speaker	Speech	Technology
CM:	"psst"	(auditory cue)
	"you have five items on your grocery list"	(verbose mode alerting)
User:	"read item"	(speech command)
CM:	"milk and bagels"	(output - synthesized speech)
User:	"next item"	(speech command)
CM:	"vegetables for salad"	(output - synthesized speech)
User:	"next item"	(speech command)
CM:	plays back a recorded audio item	(audio playback)

Table 1: Dialogue between *comMotion* (CM) and the user showing the different speech technologies employed

As the user enters the vicinity of his grocery store, the system alerts him with an auditory cue, which serves as a subtle reminder. This cue has been pre-chosen by the user and is associated with his to-do lists. If operating in verbose mode, the system will also explain why the user is getting the alert; in this case, because he has pending items on the relevant to-do list. The user may then choose to act upon the reminder and interact with the system by means of speech commands. In this example, the user navigates through the list with simple two-word commands. If the item on the to-do list was entered as text, the user will hear it as synthesized speech. However, if the item was recorded audio, the system will play back the recorded speech.

By combining the different speech technologies, it is possible to operate the main features of *comMotion* in a hands and/or eyes busy situation, that is, using speech as both input and output.

1.6 Document overview

Chapter 1 – explains the motivation behind *comMotion*, gives a basic overview of what the system does and defines terms and technologies used.

Chapter 2 – presents research done and applications designed in fields related to this thesis, that is, mainly in the domain of mobile messaging and context-awareness, and points out the similarities and differences between *comMotion* and some of the other systems.

Chapter 3 – discusses design issues related to this work; the problematic aspects involved in the creation of *comMotion*, such as the precision of the position information, or those which could evolve with the system itself, such as privacy.

Chapter 4 – lays out the architecture and the back end of the program, for example, the location learning algorithm.

Chapter 5 – gives a detailed description of both the graphical and speech interface of each module.

Chapter 6 – presents the conclusions, as well as some preliminary user evaluations, and discusses future work.

Appendix A – includes a colour copy of some of the figures which appear in the text body. Colour is an important feature in the maps and in the visual component of the reminders (for example, the yellow immediately associates them with the very common 3M Post-its™).

2. Related work

comMotion builds on other research, applications, and ideas which have been developed over the last few years, mainly in the fields of mobile messaging and context-awareness –both indoors and outdoors. Although similarities with these systems can be found, there are also a number of differences.

2.1 Context-aware systems – indoors

For interior location sensing, some type of active badge or beacon is usually used. Many of these badges use infra-red (IR) and, therefore, require line of sight between the transmitter and receiver. A number of systems based on this technology have been implemented, several of which also offer context-aware messaging for the office environment. Typically, the beacons are placed in passageways or transit areas between adjacent rooms, hence, the granularity of the information is at room-level.

2.1.1 The Active Badge Location System

The Olivetti Active Badge [5] is a beacon which transmits a unique code for approximately one tenth of a second every 15 seconds. These periodic signals are received by a network of sensors placed around the host building; a master station, also connected to the network, polls the sensors for badge "sightings". One of the applications implemented using this infrastructure was an aid for a telephone receptionist: the system provided a list of names and dynamically updated the field indicating the telephone extension number they were closest to, as well as a description of that location.

2.1.2 Forget-Me-Not

The Forget-Me-Not [6] was a wearable device which would record interactions with people and devices, and store this information in a database for later query. It interacted via wireless transmitters in the rooms and with equipment in the area to remember information such as who was there, who was being talked to on the telephone, and what objects were in the room. In this way, it allowed queries like: "Who came by my office while I was on the phone to Mark?". Its main function was to handle queries on information regarding daily activities (places visited, people met, documents submitted, etc.) by using time as a retrieval key. Both the program and the data resided in a ParcTab device [7] developed by Xerox PARC.

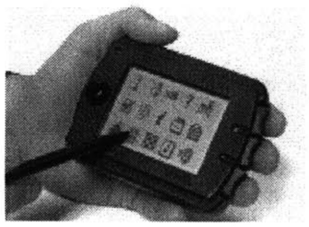


Figure 2.1: The ParcTab personal digital assistant (PDA)

2.1.3 Locust Swarm

The Locust Swarm [8] is an environmentally-powered, networkless location and messaging system which supplies information to the wearable computer user, who can then control how much of this information is to be shared either with others or the installed infrastructure. The Locust [9] transmits its pre-specified location information, and the user's system can listen to this broadcast without being detected and decide whether or not to announce his presence. The Locust also monitors if a user is trying to upload an annotation to it: when a message has been added, the Locust will intermittently broadcast its location and the message number. When another user "hears" the message number, he can either ignore it or retrieve the message content from the database.

2.1.4 C-Map

C-Map (Context-aware Mobile Assistant Project) [10] is a tour guidance system which, based on location and individual interests, provides information to visitors at exhibitions. It uses Olivetti's Active Badge System [5] for location information. C-Map provides the user with both a geographical plan of the exhibit area and a semantic map indicating the exhibits recommended for his particular interests. To chart the latter, semantic relationships are established between the user's choice of listed keywords, and the actual displays.

2.1.5 Activity Server and Watcher

The Activity Server [11] is a system which periodically reports users' locations and activities, based on information it receives from: a finger server (which provides information on user activity on the local network), a phone server (providing information on use of phones), and a location server (which provides physical location of users wearing Olivetti Active Badges [5]). This system gives location and activity information, such as, "Joe is alone in his office, active on computer-A". Watcher [38] is a graphical interface to the Activity Server system.

2.1.6 RadioSpace

Using IR-based position tracking, RadioSpace [12] was designed to provide location awareness of nomadic wearable computer users and their presence in physical space and time. The system enables users to send messages to others or to different rooms. Moreover, dynamic Wed-based maps allow them to view the location of other users and passively listen to cues indicating their activity. A position-server maintains a history of user interaction and movement.

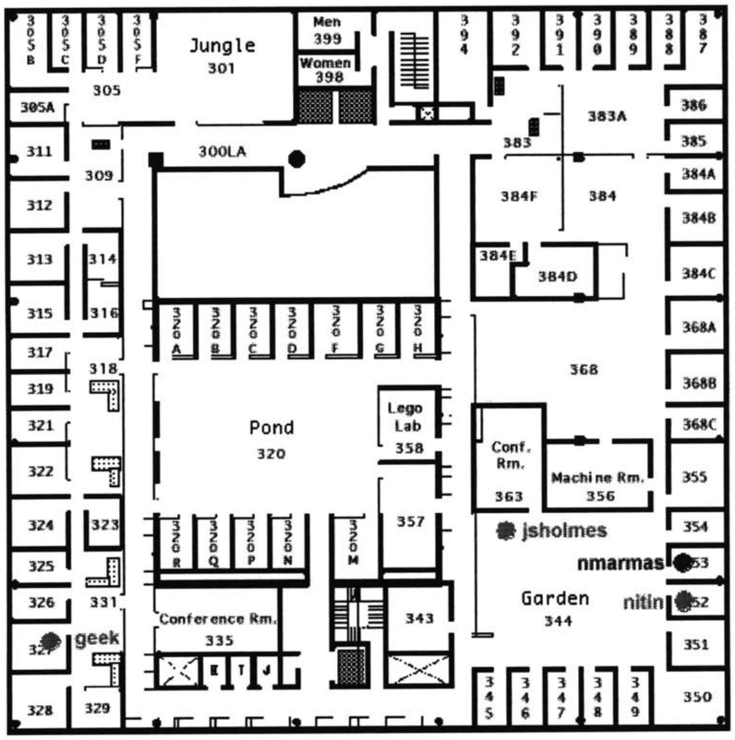


Figure 2.2: The RadioSpace Web-based dynamic map showing users' physical location

2.1.7 The Remembrance Agent

The Remembrance Agent [13] is a program which continuously monitors the wearable computer user and displays one-line summaries of note-files, old e-mail, papers, and other text information which might be relevant to the user's current context. These summaries are listed in a few lines at the bottom of a heads-up display, so the wearer can read the information with a quick glance, however, to retrieve the whole text described in a summary line, he must use a chording keyboard.

2.1.8 DyPERS

DyPERS (Dynamic Personal Enhanced Reality System) [14] is an audio-visual memory assistant which, using perceptual cues, prompts the user at relevant times. By means of a head-mounted camera and a microphone, it sees and hears what the user perceives, and thus collects an audio-visual memory. The resulting multimedia database can be indexed and played back in real-time. The user indicates to the system which visual objects are important memory cues, and in this way it will learn to recognize them in future encounters and associate them with the recorded memories. DyPERS was implemented in an application for a museum-gallery scenario: the user was given a tour with explanations of the paintings he was seeing. If and when the user sees the same paintings again, the audio and video clips with the explanations he once heard, will be immediately triggered.

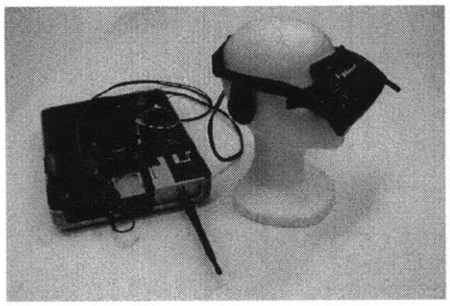


Figure 2.3: The DyPERS system

2.2 Context-aware systems – outdoors

There are other systems which use GPS or bar-codes to track users and deliver information, for example, in guided tours of cities or a university campus. Most of these systems are not user-dependent, that is, the information is triggered according to physical location regardless of the user's identity.

2.2.1 CyberGuide and Savoir

The CyberGuide project [15] includes a series of prototypes of hand-held intelligent tour guides which provide information to tourists based on knowledge of their position and orientation. Prototypes for both indoor and outdoor tours were developed. The idea behind the project is that the size of a PDA is similar to that of the guidebook a tourist might take on vacation. Although the book might provide important practical information, such as locations of interesting sights, hotels and restaurants, it does not know where the tourist is located when he needs the facts. The application tries to predict what the user is attending to and supply relevant information.

The Savoir (Somewhat Assisted Voice-Only Interaction Research) project [16] explored issues related to context-awareness and voice-only computer interactions. A prototype, which enabled users to receive information from the Internet by means of a standard wired or cellular telephone, was developed. A version of CyberGuide, with an interactive tour guide facility, using the Savoir infrastructure and a GPS receiver, was implemented.

2.2.2 MetroNaut

MetroNaut [17] is an application developed for schedule negotiation and guidance instructions for a visitor to a university campus. It uses a bar-code reader for information input and position location, a two-way pager for communications, and an ARM processor for computation. The visitor's position is determined by reading bar-codes at information signs around campus. Modifications to the schedule are negotiated with the campus computing infrastructure by means of the two-way pager.

2.2.3 Stick-e Notes

Stick-e Notes [18] is an electronic version of the Post-it note. This application enables users to wander around outdoors attaching Stick-e Notes to locations and later have the notes trigger when passing through the area again. The system is user-dependent and GPS is used for positioning information.

2.2.4 FolksFinder

FolksFinder [19] consisted of a centralized Web-site which served as a clearinghouse for information produced by wearable computers equipped with GPS and cellular modems. Participants with wearable computers continually broadcast their positions to the centralized server from which they, and other non-wearable users, could access location information or be notified if a user was approaching a certain place.

2.2.5 City Guide

The City Guide application is part of the OnTheMove project [20]. It enables a user to know his geographical position by means of a digital map; as he moves, the map is scrolled to reflect his change in position. The user is able to view maps other than the one he is located on, scroll them, zoom them and search for restaurants and hotels. Position information is obtained by GPS; requested services are received via WaveLAN or GSM. Larger and more detailed maps can be received if the connection is made through WaveLAN.

2.2.6 Augment-able Reality

The Augment-able Reality system [21] allows users to dynamically attach digital information, such as voice notes or photographs, to the physical environment, through wearable or desktop computers. Attached data is stored with contextual tags such as location IDs and object IDs –obtained by wearable sensors. Wearable users can then access the data when they come to the same context, regardless of whether the data was posted by another user.

2.3 Mobile messaging systems

Pagers and cellular phones are mobile communication devices, which are usually unaware of the user's context or physical location. Clues [22], a dynamic filtering system which has been in use over the last couple of years by some members of the MIT Media Lab and has been built into several messaging systems, infers geographic location from entries in the user's calendar, however, it does not know the user's actual location. Outdoor location information can be obtained from devices such as cellular phones through triangulation between base stations, in which case the location granularity is usually a cell. However, it is important to take into account that, in general, only service providers can know the location of a cellular phone and that this information is not usually available to the user. Still, cellular phones with built-in GPS units are starting to appear and, judging by the new-fashioned PDAs and cellular phones, connectivity, mobile messaging and access to personal organizer data, are becoming expected features.

2.3.1 Phoneshell

Phoneshell [23] permits mobile access to desktop applications. Using a touch-tone telephone, one can access e-mail, voice mail, calendar entries, Rolodex cards and hourly news broadcasts. Data entry, such as answering e-mail and adding new calendar items, is possible using DTMF tones or recording audio. Phoneshell is two-way, provided the user is motivated enough to type using the telephone keypad.

2.3.2 Chatter

Chatter [35] is a system which was built to provide the main functionality available in Phoneshell [23], however, with an improved voice interface. It used continuous, speaker-independent speech recognition for input and in this way enabled many more commands than is possible with DTMF tones. A discourse manager tracked the user's place in the dialogue and maintained context among separate tasks. For example, after reading a message from someone, the user could look up his number in the Rolodex and call. In addition, the system observed the user's history of actions and used this knowledge to predict the user's next action. For instance, if e-mail from a certain sender was always saved, the system would suggest: "Save it?".

2.3.3 Portico

Portico [36] is a virtual assistant which manages voice mail, e-mail, calendar and address book programs, and satellite news feeds. Interaction with the system is by means of natural language (as opposed to terse speech commands) through a phone, or via a Web-browser. The system can answer the phone, route calls or messages, make appointments, and prioritise and read e-mail messages. Certain personal information managers, such as the PalmPilot, can synchronise with Portico over the Web.

2.3.4 Webley

Webley [37] is a personal communications assistant that places, screens, and takes calls in addition to forwarding e-mail, voice mail, faxes or pages to one or more locations. Moreover, other communication can also be controlled over the phone (with speech recognition or with touch tone keys), or over the Web. For example, it is possible to hear an e-mail message and dictate a reply; hear a voice message, record an answer and have the system return the call and play your recorded answer; or place a conference call.

2.3.5 Nomadic Radio

Nomadic Radio [24] is an audio-only wearable interface which permits one-way access to remote information services, such as e-mail, voice mail, hourly news broadcasts and personal calendar events. The messages are automatically downloaded to a wearable device throughout the day and users can browse them using speech recognition and a tactile input. Nomadic Radio uses the Clues [22] dynamic filtering system and the user's level of attention (inferred from his degree of interaction with the system) to offer timely messages. To provide an unobtrusive interface for nomadic users, the audio/text information is presented using a combination of ambient and auditory cues, synthetic speech and spatialized audio. Moreover, it's alerting system is adaptive.

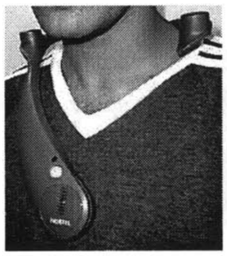


Figure 2.4: Nortel's Soundbeam Neckset wireless telephone, adapted for Nomadic Radio

2.3.6 Knothole

Knothole [25] is a system which uses two-way pagers as a mobile interface to a desktop computer, hence, it combines PDA functionality, communication, and Web access into a single device. Instead of putting the intelligence into the portable appliance, it relies on a wireless network to connect to services which enable access to multiple desktop databases, such as a calendar or Rolodex, and external sources, such as news, weather, stock-quotes, and traffic.

Figure 2.5: The Motorola PageWriter 2000

2.3.7 Active Messenger

We have abundant channels through which a message can reach us, such as, e-mail, voice-mail, wired and cellular phones, different pagers, and fax machines. Active Messenger [26] infers the user's location based on which device he last used, and modifies its filtering and forwarding rules accordingly. For example, use of your business phone would indicate you were in your office, whereas use of an iridium pager would probably indicate that you were out in the middle of nowhere.

2.3.8 AT&T PocketNet Service

PocketNet [27] is a wireless information-access service provided by AT&T which enables users to link to their desktop and access their e-mail, calendar and Internet sites. The e-mail and personal organizer data is synchronized with the desktop. Currently, two manufacturers are offering wireless phones that are compatible with AT&T PocketNet service: the Samsung Duette and the Mitsubishi 100. Furthermore, software companies such as Umail [28] are providing services, based on PocketNet, which enable the delivery of Internet-based services to wireless telephones or a PalmPilot.

Figure 2.6: A PalmPilot with a Minstrel 19.2 Kbps wireless IP modem

2.3.9 Ericsson, Motorola and Nokia

American Personal Communications (APC), a subsidiary of Sprint, is using GSM as the technology for a broadband Personal Communications Service (PCS). This will ultimately have more than 400 base stations for the palm-sized handsets being made by Ericsson, Motorola, and Nokia. The handsets include a phone, a text pager, and an answering machine.

Nokia's 9000il Communicator [29] is a fully integrating GSM 1900 digital phone with a personal organizer, fax capabilities, data and messaging services, as well as Internet access. However, the user must sync-up with the standard calendar programs and other applications on his PC.

Figure 2.7: The Nokia 9000il Communicator with personal organizer, fax capabilities, data and messaging services

2.3.10 3Com's Palm VII

A widely anticipated product is 3Com's Palm VII. This device, currently in field tests, will combine the functions of a standard Palm hand-held organizer with a wireless connection for Web and intranet access, as well as two-way messaging.

2.3.11 GPS and mobile phones

GPS receivers have been combined with mobile phones for a number of functions. An interesting application of location information and messaging is one to protect vehicles: the GPSS system [30]. The GPS and mobile phone electronics are hidden in the vehicle to be protected. The system requires a PC running GPSS, and a modem connected to a regular telephone; the computer could be located anywhere in the world. If the car is stolen, and you want to know where it is, you simply use the telephone to dial up the number, and the GPS in the car then sends back GPS data every second, without those in the car knowing. The data is displayed on maps (GPSS software) running on the PC.

2.3.12 Garmin Navtalk GPS / Cell-phone

The Garmin Navtalk [31] is a combination of an analog cell-phone and a GPS receiver, with touch-tone location reporting and numeric paging. On the display, you can see exactly where you are calling from and this information can be sent to another unit. It is a full-featured GPS receiver and includes an internal database with cities and roads in North and South America.

Figure 2.8: The Garmin Navtalk GPS / Cellular Phone

2.4 How *comMotion* differs

comMotion has integrated and learned from many of the aforementioned systems. As a result, it correlates with some of these in the sense that it uses position data to provide context-aware information. Nevertheless, *comMotion* also differs in several regards, namely time, place and information. However, what sets it apart is its independent-learning property.

2.4.1 Time

Systems such as the Forget-Me-Not [6], the Remembrance Agent [13] and DyPERS [14] serve as a memory prosthesis and augment memory for past events. In *comMotion*, the to-do lists and reminders are amongst the main features, that is, the system relates to events in the future. Since *comMotion* does not keep a record of past events, it could be a good complement to a system such as the Remembrance Agent. In practice, a user could choose to never delete old items from a to-do list or old reminders, and in this way have a record of the past. However, this would be very inconvenient since the to-do lists would become endless and would contain mostly checked items. Furthermore, outdated reminders would not be triggered since the context (which includes the date) would never match. So although a record of the past would be possible, it would defeat the purpose of the system.

2.4.2 Place

The Locust Swarm [8] enables leaving messages at specific indoor locations and retrieving them, yet the user must be in the particular site to upload a message to the Locust and likewise in order to retrieve it. Stick-e Notes [18] permits the user to attach electronic Post-it notes in different locations but, once again, the user must be in the locale in order to retrieve the note. Unlike these, *comMotion* allows messages to be posted regardless of the user's location and in this way is more related to RadioSpace [12], which enables messages to be sent from any Web browser. A *comMotion* user can also choose to view all of his to-do lists and reminders, regardless of his current physical location, and likewise browse any reminders, past or future, that he has received from others.

Mobile messaging systems using the Clues [22] dynamic filtering, infer geographic location based on users' calendar entries. The Active Messenger [26] tries to infer location from the person's usage of the different devices and uses this knowledge to deliver timely messages to the most appropriate channel, or device. Since *comMotion* knows the users' exact, as opposed to inferred whereabouts, it could be a good complement to the Active Messenger. It could not only provide the Active Messenger with the virtual location ("home", "work", "grocery store") of the user, but also serve as an obvious communication channel for timely message delivery.

2.4.3 Information

Certain ideas in the CyberGuide [15] and City Guide [20] projects are related to those implemented in *comMotion*, however, these systems access predefined databases of maps and site information, as opposed to user-definable information. *comMotion* users can also access maps and locale information; in addition they have personal location specific to-do lists and reminders, that is, user-defined information, as well as the possibility to subscribe, per location, to other Web available content, such as weather, headline news, and movies.

2.4.4 Behaviour learning

What is unique in *comMotion* is its behaviour-learning module. Behaviour learning in DyPERS, for example, relies on user input, that is, the user must classify the data and specify whether the system should remember this information. Moreover, none of the systems that I am aware of observe the user's mobility data and identify frequented locations. Although the *comMotion* user can explicitly teach the system, for instance, when visiting a place for the first time, this type of teaching is not necessary. Once *comMotion* has located a site, after the third visit, it will learn it on its own and simply ask the user for feedback in the form of a virtual location name ("home", "work", "grocery store", etc.) or a command to ignore, for example, in the case of a bus-stop. In that sense, *comMotion* does not rely on the user for its learning process.

To sum up, *comMotion*, through its multi-modal interface, uses context-awareness to provide the mobile user not only with timely information, but also with the possibility of choosing the most appropriate type of interaction, that is, text and/or speech. No boot-strapping is required, rather, the behaviour-learning agent monitors the user and progressively learns the salient locations in his life, hence relieving him from the tedious task of classifying data or filling out preliminary forms. Once these frequented locations have been established, the system uses this knowledge to deliver to-do list items, relevant reminders and other information, when and where it is needed.

3. Design issues

The main objective of this work was to build a personal system which constantly knows the user's location and exploits this knowledge to deliver timely information. For the device to be most effective it should be highly portable and, hence, should be with or on the user at all times. Furthermore, it must enable multi-modal interaction to accommodate the user's different possible contexts.

comMotion, with the appropriate hardware, and with some modifications in the software could accommodate different architectures. A number of scenarios can be envisioned, each adapted for different life-styles, or different modes of mobility:

- a wearable on-the-go architecture for the highly active, such as, cyclists;
- a car architecture for the more sedentary;
- a briefcase architecture for the mobile individual, for example, knowledge workers; or
- a stripped down kids architecture.

These are all variations on the same *comMotion* system, tailored to different needs. What changes is the hardware (*cfr.* section 3.6) and the features included in the system which range from full-fledged to stripped down variations.

Although the different architectures represent different paradigms, they share many design issues.

3.1 Why speech?

Computers are becoming smaller and faster; consequently they are becoming more mobile. However, their traditional interface (namely the display and keyboard) is simply being proportionally scaled down. As computers become more portable, our expectations of them change. For example, if we have a computer with us in the car or while walking down the street, why not be able to use it in these situations? Small displays and keyboards do not necessarily meet this new demand where our hands and eyes might be busy and we may be attending to another task, such as driving. In such circumstances, a speech interface is beneficial and auditory alerting essential (*cfr.* section 3.3.1).

Speech , on the one hand, provides portability since a microphone is much smaller, lighter and easier to carry than a keyboard, and it can be used even when one's hands and/or eyes are busy. Likewise, it is also a more natural, faster and an almost effortless form of input, as opposed to text. Speech is much more expressive than text, as information is often contained in prosodic cues, that is, in the different intonations. Furthermore, it has been shown that when doing multiple tasks, performance improves when attention is divided amongst channels of different modalities [32], in other words, it is easier to attend to a visual and an auditory task simultaneously, than it is to two visual or two auditory ones.

On the other hand, speech recognition is still fragile and this may prove frustrating for the user, and eventually lead to mistrust in the system. Speech commands may not be recognized or they may be mis-recognized and an unwanted action performed. Speech input may also be problematic if ambient noise levels are high. Since audio is temporal and serial, it is slow to absorb and difficult to retain in large quantities. Output speech, especially if it is synthesized, places a higher cognitive load on the user [39]. Needless to say, it is much easier to glance at than hear a to-do list and quickly pick out and remember the important items: such browsing and scanning, that is, random access, cannot be done with audio. Furthermore, audio output is much more public than a visual display, hence, privacy issues must be considered.

Therefore, when information is submitted as speech, great attention must be placed on how it is presented and how it is to be navigated. Nevertheless, speech interfaces can be effective and beneficial for specific tasks or under certain circumstances. The speech interface in *comMotion* does not replace the graphical one but rather complements it and grants the user functionality in circumstances where his hands and/or eyes are busy.

3.2 Learning

A system which requires the user to fill out an extensive form of data before it can operate, will typically not be used. Generally, a user may be inclined to make this effort once he has experienced the benefits of the system, but not before he believes that it will pay-off. If *comMotion* requested the user to initially define his ten most important frequented-locations and their coordinates, it would be used by no one but the developer. Besides, most people do not know the latitude and longitude of their home, let alone other places; and why should they?

A behaviour-learning agent monitors the user's mobility data and learns the frequented locations. Once such a location has been identified it is necessary for the user to determine if it is an important place, that is, a place where context-relevant information should be received. This is done by simply naming the place, or indicating that it should be ignored. At that point it is acceptable to ask for feedback from the user. The system can identify whether a place is frequented, however it cannot determine the meaning of this place to the user.

This independent location learning is crucial since it implies that no boot-strapping is necessary. The more locations the system is aware of, the more functional it will be. So during the initial days, the system will mainly be monitoring and learning, while providing little context-related information. Although it is possible to explicitly teach the agent, this instruction is not essential.

3.3 Information delivery

3.3.1 Notifications

When a user enters a certain context which triggers information, he must be notified that there is relevant information awaiting attention, and this data must be delivered. It is important to remember that the user will typically be attending to some primary task, for example driving, therefore the notification must draw his attention, in the background, but not totally distract him from the task at hand. Notifications are given as auditory cues, which are chosen by the user and so can be as subtle and graceful as he likes, depending on personal choice. Moreover, different cues can be selected and associated to the different types of data: to-do lists, reminders and subscribed information.

Notification in the form of auditory cues can be given when approaching a location or when departing. If approaching a location, the timing of the cue is very important and should be correlated with the user's mode of transport. For example, notification that there are items on the grocery list can be given within a couple of metres from the store entrance, if the user is walking or cycling; however, if the user is driving, such short notice could prove dangerous and in any case might not allow time to act upon it. Therefore, the user's acceleration is also an important element of his context. Currently, the radius which defines the vicinity around a location is static. Future versions will define the radius as a function of velocity: the slower the user is moving, the smaller the radius. This should improve the timing of the notifications, but given GPS only has an a 100-metre accuracy, it will still be far from ideal.

3.3.2 Interfaces

A good interface should be coupled with a user's mental model. The simpler it appears to the user, the more intuitive, powerful and useful it will be. The GUI implemented has a simple metaphor which mimics its real world parallels. For example, the to-do list is a visual component which resembles a list on which items can be checked (similar to a paper note); the reminders match, in shape, size and colour, the widespread 3M Post-its™. However, a metaphor, or mental model, for an audio interface is much more intricate. The speech commands chosen are all simple one or two-word commands and should be intuitive as to their corresponding function. For example, "first item", "last item", "next item" and "previous item" are used to navigate within a specific to-do list, and "check item" is used to mark an item as checked. When using an effective visual interface the user quickly perceives if the command was understood and implemented. For instance, when marking an item as checked after it has been done, the user will notice if the wrong one was ticked. However, in an audio interface, such feedback must be explicitly given, especially to the novice user until confidence in the system is attained. Even experienced users often prefer explicit feedback in order to catch and be able to correct mis-recognitions of the speech recognizer.

3.3.3 Modes

comMotion can be run in terse or verbose mode, in which case more auditory feedback is given. Operating in verbose mode implies that, in addition to generating an auditory cue associated with a to-do list, the system will say: "You have five items on your home to-do list", or in response to the command "check item", it will say: "item: get milk, is now checked".

The to-do lists serve as a prosthesis and augment memory. Cued recall is known to be more effective than uncued recall. If we are explicitly told that we have five things to-do, it is easier to remember what they were than if simply asked: "What were the tasks on the list?". Although *comMotion* is not intended to improve a person's memory, it may have this slight effect. When the user is cycling by the grocery store and receives an auditory cue and a verbose alert: "You have five items on your grocery list", he may not need to explicitly ask for an audio list of the data. The verbose cue may be sufficient to trigger information recall. This of course also depends on the length of the to-do list.

3.3.4 Interaction

For such a system to be useful, it must be aware of the user's context and attentive to his possible mode of interaction. Ideally, the user should at any point in time be able to speak and interact with the system. Continuous speech recognition is generally not enabled since the application does not know when it is directly being spoken to, or when the speech is ambient conversation, and hence may act upon speech which was not intended as direct input. Generally a push-to-talk button is used to disambiguate between these two cases. However, if the user is not in a hands-free situation, he cannot be expected to activate a push-to-talk button. Therefore, *comMotion* uses a combination of continuous speech recognition and a push-to-talk mode. When the user has been notified (auditory cue) of context-aware information, continuous speech recognition is turned on, thus giving him a chance to act upon the cue with a speech command. As long as the user interacts with speech, the continuous recognition will remain activated. However, if the user, while walking down the street, has suddenly thought of an idea he wants to record for future reference, he will have to use the push-to-talk button in order to activate the speech recognition and only then can he record his thought.

Continuous speech recognition is automatically enabled when the system expects speech commands. Otherwise, when the system does not anticipate them, the speech recognition must be manually activated by means of the push-to-talk button. In either case, once activated, the continuous recognition remains enabled as long as commands are received.

3.4 Privacy

A system which monitors the mobility of a user can have numerous advantages but it also raises many questions regarding privacy and social fears: an immediate reaction being one of "Big Brother is watching". Users fear the use or misuse of their personal information and are concerned if it can be accessed without them even knowing that it is being released. Therefore the user's privacy must be taken into account at the design level. The GPS data and behaviour-learning agent's monitoring is done on the client device, and this information is accessible only to the client. Consequently, the information is safe to the extent that the portable device is safe.

Nevertheless, there are cases in which the user may want his location to be accessible to others, so a query engine was also implemented (*cfr.* section 4.3). A *comMotion* user can define position information access priority per user and per location, hence enabling maximum privacy and avoiding the feeling of "many little brothers" monitoring him. The query request goes through the server to the specific client. If the request was sent by an authorized person, he will receive the location information, for example, "user-X is at home"; however, if queried by someone unauthorized, the response will be "user-X is incognito". In either case, the queried user can log by whom and when location information was requested. It is important that the user have full control of what information is given, and to whom.

The only data that the server stores is a list of the users, their defined virtual locations and priorities, but no physical location information is attached to these virtual tags. This data is necessary to enable users to send messages to each other's virtual locations, provided they have been authorized to do so. For example, user-A has the following defined virtual locations: "home", "work", "grocery store", "hardware store" and "Jim's place". User-A has authorized user-B to know about "work", "home" and "grocery store". Consequently, user-B can only send a reminder to user-A to one of these three locations, and if he queries the whereabouts of user-A, unless user-A is in one of those locations, he will be told that "user-A is incognito". Although user-B knows that user-A has defined "work", "home" and "grocery store", he only knows that they exist as virtual locations but their actual physical location is unknown to him.

When a new location has been identified and defined, it should automatically be added to the corresponding file with the default priority chosen by the user. The system should have a simple interface notifying him by whom he is being queried and whether or not information was released. In this way, the user will have maximum privacy but will also be aware of whom was granted or denied information, and he can easily modify the priorities set. This prototype has no specific interface for these functions but they will be included in the next version. Currently, the location must be added manually and its priority set and the log file must be viewed in a text editor.

3.5 Precision of position

When designing location-aware applications it is vital to know where people are. The ideal sensor for such a system would provide location information seamlessly both in- and outdoors. Nevertheless, at present, such a sensor is not available, at least not at an affordable price.

Buying a GPS-receiver and connecting it to a computer's serial port is simple and inexpensive. Global position information becomes immediately available and is maintenance-free as far as the user is concerned. In contrast, indoor positioning systems are limited to the area in which they are installed and they require constant maintenance, such as replacement of power supply. Typically, a user would only need an interior positioning system in his work area, where being in the office as opposed to a conference room may imply very different contexts. At home, knowing that the user is in the kitchen as opposed to the living room, would probably result in little added benefit. However, it has been suggested that knowing the exact aisle a user is in at the grocery store, may be advantageous.

Therefore, *comMotion* focuses solely on outdoor tracking and user-behaviour learning based on data acquired by means of a GPS-receiver.

GPS has an accuracy of less than 100 metres. Although position information may be inexact, in many cases it is sufficient to know the vicinity (within 100 m) in which the user is located. This is true provided the defined locations do not have overlapping, 50-metre radius, boundaries. In the real (physical) world, a user cannot be in two places at the same time, however when virtual locations overlap, the user may be in two (virtual) locations at the same time. The system, however, will not recognize both locations; one location will shadow the other. For example, if a bank and a book-store are adjacent locales, the user could get reminders relevant to the bank <u>or</u> those pertinent to the bookstore, but not both. Whether the bank will shadow the bookstore, or vice versa, depends on the GPS data. The user could be approaching from the bank side and still get the bookstore reminders, due to the lack of precision in the received data. This was not accounted for in the original design, and in the future, it will be necessary to look at virtual location clusters. DGPS could be used to provide two-metre accuracy, and the virtual location boundaries could be reduced in size when two adjacent boundaries overlap, however it will still be necessary to identify the clusters. Only thus will the user get all the context-relevant information for all defined locations in his proximity. Proper clustering will eliminate the need for the precision offered by DGPS though more precise position information would enable more efficient timing of the auditory notifications. In order to have real-time DGPS, a base station would have to be setup to transmit corrections (*cfr.* section 1.4.2) to a radio receiver, which would have to be included in the *comMotion* hardware. Besides additional costs of the base station (computer + $200 GPS unit) and radio receiver ($1,500) it implies added weight and volume to be carried by the mobile user.

Not only is GPS data not very accurate, but it is also not always available. GPS signal is lost when entering most buildings and the so-called concrete canyons in urban areas make reception difficult. The fact that most buildings are GPS opaque was exploited advantageously, permitting a simple learning mechanism of the locations of buildings. Section 4.1 explains the learning algorithm and indicates how the reception problem was resolved.

3.6 Hardware

Ideally, for such a system to be most beneficial to the user, it should be with him at all times; preferably as part of a wearable computer. It should be forgettably small, lightweight, and long-lived between battery replacements or recharging. A PalmPilot sized device which could be attached to your wrist or easily fit in a shirt pocket, together with a wireless directional lapel microphone, and a wireless earphone speaker, would be perfect. The main surface area of the device would be a display used by the GUI, which complements the speech interface.

There are many social issues associated with talking to yourself (a perceived impression when using speech commands), as well as with the use of earphones –people do not like to be looked upon as having a physical impairment. Therefore, the possibility of attaching the device to your wrist is important as it reverses the effect and causes others to perceive you as "cool" as opposed to "weird". Although the above scenario sounds futuristic, it is only a matter of time before the ideal hardware is created: one which will be neither a physical nor a social burden.

Until the ideal hardware exists, other scenarios can be envisioned and built. Different architectures can be adapted for different life-styles, or different modes of mobility, and modifications in the software can be made to accommodate these architectures. In other words, variations on the same *comMotion*, tailored to different needs.

Possible scenarios include:

- a wearable on-the-go architecture for the highly active, such as, cyclists;
- a car architecture for the more sedentary;
- a briefcase architecture for the mobile individual, for example, knowledge workers; or
- a stripped down kids architecture.

However, this is by no means an exhaustive list since different architectures could be adapted for different needs and life-styles.

The car architecture would include a computer system installed permanently in the user's car. It would include a GPS receiver although many cars already come with a built-in receiver, and it would be possible to interface to the existing one. The car computer unit would have a full-fledged version of *comMotion.* Additionally, it would include options to download (via the IR) the to-do lists or reminders to a PalmPilot. Imagine driving in the vicinity of your grocery store and getting an auditory cue, alerting you that you have items on your shopping list. You pull into the parking lot, place your PalmPilot near the car computer and download the list of groceries you need. As you shop, on the PalmPilot list, you tick off the items purchased. Back in the car, you once again place the PalmPilot near the car computer and sync the shopping lists. This architecture could be built with existing hardware.

A first *comMotion* prototype was built in order to evaluate its feasibility and usefulness. Its architecture corresponds to that envisioned for mobile individuals, namely the briefcase architecture. Since the futuristic on-the-go architecture is yet down the road, the prototype built was used in highly active situations (cycling) by inserting all the components in a fanny pack or carried in a shoulder bag.

A kids stripped down architecture could be built on a PalmPilot with a CDPD modem and a GPS receiver. Its main functionality would be for the parents to know where the child was and for them to be able to send each other messages or reminders. Such an architecture would not include a speech interface.

Currently, the hardware used for *comMotion* (Figure 4.1) is any portable PC, which runs Win95. The audio interface requires the use of a microphone and speaker. Although headphones could be used, they are considered anti-social and they limit the user from hearing other sound sources, which could be dangerous if cycling or driving. The Jabra, a single-ear earphone with a built-in bone conductive microphone, was chosen since it is more discrete and does not prevent the user from hearing other ambient sounds. However, a wireless solution would be more adequate as wires tend to get in the way.

Chapters four and five provide an in-depth description of the *comMotion* architecture and interface, and show how the aforementioned design issues were resolved.

4. Architecture

comMotion consists of a client application, operated on a mobile or wearable platform, which communicates wirelessly over the Internet to remote servers running on a Sun SPARCstation. The client application was developed in Java 1.1 and has been used on a Mitsubishi Amity Vp mobile computer with a 586-133MHz processor and on a Digital HiNote with a Pentium 233 MHz processor.

Java was favoured since it is platform independent, thus allowing the client to be run in the future on other Pentium-based wearable computers. The different server processes were written in both Perl and Java.

The hardware (Figure 4.1) includes a portable PC, a GPS receiver, a CDPD modem and a Jabra earphone speaker with a bone conductive microphone. In spite of the fact that smaller GPS receivers are available, that is, either OEM boards or PCMCIA cards, *comMotion* uses the Garmin receiver because its display shows the satellites in sight and the strength of their signals (a useful feature when debugging). A PCMCIA card CDPD modem would also be preferable.

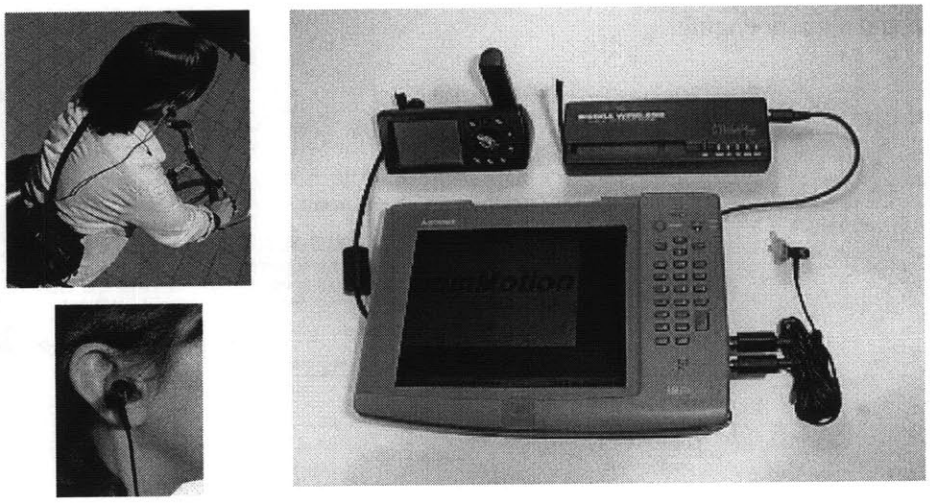


Figure 4.1: Hardware components of the *comMotion* system

The human-computer interface, on the client side, is composed of both a speech and a graphical interface. The former includes speech recognition and text-to-speech synthesis and was developed using AT&T's Watson SDK (software development kit) [33]. The Watson product is an integrated, Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) synthesis system which complies with the Microsoft Speech API (SAPI). The ASR engine uses phoneme-based sub-word analysis and, therefore, supports speaker-independence and continuous speech recognition. Watson runs on 32-bit PC platforms using Win95 or NT. The *comMotion* speech server, developed with the Watson SDK, operates on the client device.

The client application communicates via TCP/IP sockets to all the different server processes, hence, these processes could easily be transported to the client device or to any other computer with Web-server capabilities. In the current setup, with only the speech server on the client device and all other servers on a remote station (Figure 4.6), even if connectivity were lost the user would still have full access to his to-do lists and any reminders which had previously been downloaded. Furthermore, since all position tracking and analysis are done on the client device, these would not suffer from lack of connectivity. Reminders sent from other users are immediately downloaded to the client device where they are stored until delivery time. If the server cannot access the client, these new reminders will be saved until connectivity is re-established and they can be downloaded. Lack of Internet connectivity means information from on-line sources will not be accessible; likewise, no maps or related information can be downloaded.

The system can be divided into three main modules (Figure 4.2): a behaviour-learning agent, a message engine and a query engine.

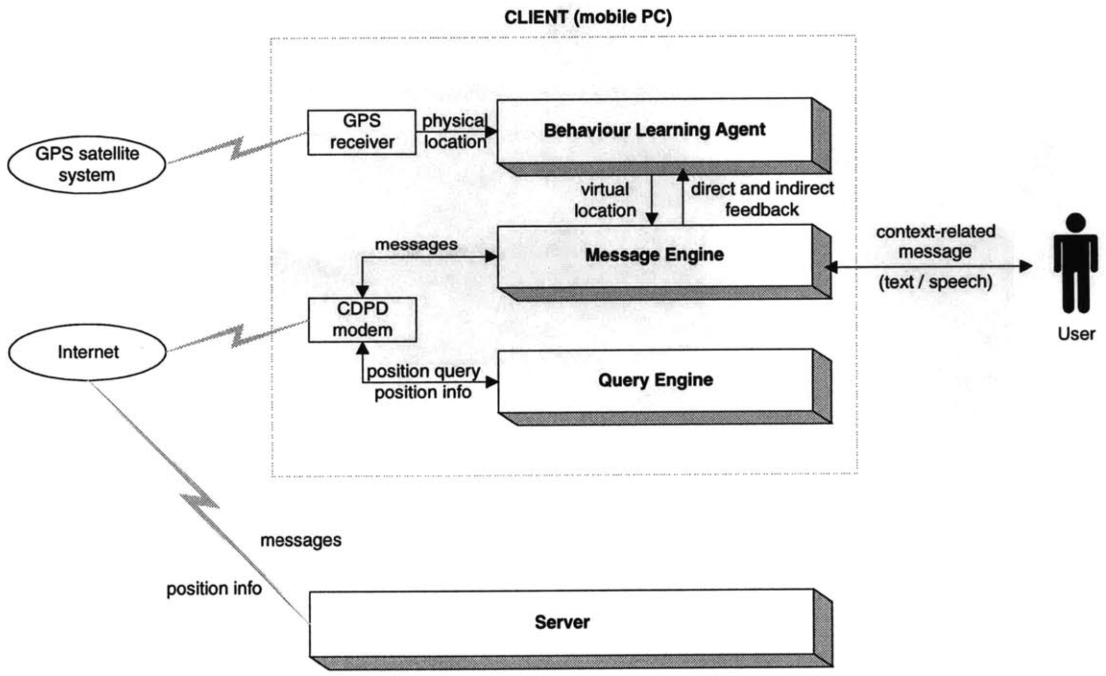


Figure 4.2: The architecture of *comMotion* showing the three main modules of the client application and its connection to the server

4.1 The behaviour-learning agent

Latitude and longitude coordinates are obtained via a GPS receiver connected to the client's serial port, using the NMEA-0183 protocol. All data is analysed for frequented locations, however, at present, only locations such as buildings can be identified, since the system recognizes locations where GPS signal is lost. After losing signal within a 50-metre radius, on three different occasions, the agent infers that this must be a building and marks it as a salient location. At this point the user is prompted for a location name, which he can either designate or tag at a later stage by seeing the location on a map. Once tagged with a name, the virtual location has a to-do list associated with it. However, if the location is of no interest, the user can indicate that it is to be ignored. For example, a frequented T-stop (metro) would be identified by the agent but the user would typically not want to have context-related information (to-do list items, electronic reminders and subscribed information) delivered there.

Initially, locations were identified both as the user arrived at them as well as when he departed from them. However, it was found that the GPS receiver often took several minutes to acquire its location when exiting a building and, therefore, when the signal was regained, the user was several minutes away from the place he had left. Depending on his mode of transportation, this could be a considerable distance, hence, the user's frequented locations were identified but so were several false ones. To compute its position, a GPS unit must receive a signal from various satellites: three for a 2D fix (latitude and longitude) and four in the case of a 3D fix (latitude, longitude, and altitude). In urban areas, shadowing from tall buildings often occurs, and leads to long delays in position acquiring.

The GPS receiver is polled for data every ten seconds. The time is recorded just before the polling, and subsequently the data received. Hence the GPS data on any specific line corresponds in fact to the geographic position at ten seconds before the time recorded on the following line. If the receiver has at least four satellites in sight, data will be received whenever polled and therefore the GPS coordinates on a specific line will correspond to the position at the time it was polled.

$$\text{if } (\text{Time}_{X+1} - \text{Time}_X) = 10 \text{ secs.} \Rightarrow \text{Pos}_X \text{ corresponds to } \text{Time}_X$$

but

$$\text{if } (\text{Time}_{X+1} - \text{Time}_X) > 10 \text{ secs.} \Rightarrow \text{Pos}_X \text{ corresponds to } \text{Time}_{X+1} - 10 \text{ secs.}$$

Date	Poll time	GPS data
03/30	09:43:53	$GPGLL,4223.351,N,07104.613,W,144235,A*3B
03/30	09:44:03	$GPGLL,4223.153,N,07104.625,W,144305,A*3C
03/30	09:44:13	$GPGLL,4223.092,N,07104.641,W,144315,A*33
03/30	09:44:23	$GPGLL,4221.867,N,07103.786,W,145427,A*39
03/30	09:54:14	$GPGLL,4221.870,N,07103.787,W,145429,A*30
03/30	09:54:24	$GPGLL,4221.872,N,07103.789,W,145431,A*35

Figure 4.3: GPS data showing how signal is lost when arriving at a building

As can be seen in the GPS data (Figure 4.3), at 9:44:13 the user arrived at location (4223.092N, 7104.641W). Ten seconds later the receiver was polled for data but a signal was received about 10 minutes later, locating him at (4221.867N, 7103.786W). If the GPS receiver loses signal two more times within a 50-metre radius of (4223.092N, 7104.641W), then this is understood to be a building and, consequently, a frequented location.

Date	Poll time	GPS data
03/29	16:34:09	$GPGLL,4221.686,N,07105.339,W,214234,A*3D
03/29	16:42:24	$GPGLL,4221.687,N,07105.341,W,214236,A*31
03/29	16:42:34	$GPGLL,4221.685,N,07105.392,W,214238,A*33

Figure 4.4: GPS data showing how signal was acquired only several minutes after departing from a building

In this second example (Figure 4.4), the user turned on the unit and left the office at 16:34:09, however, the GPS receiver took several minutes to acquire a position after exiting the building. Data was received ten seconds before the next entry, that is, at exactly 16:42:14. So in this case, data was received eight minutes and five seconds later –this includes the time it takes to descend three flights of stairs besides the acquiring time once outside the building. The user was identified at location (4221.686N, 7105.339W) which is, in fact, somewhere between the office building and the parking lot. Figure 4.5 shows the office building on Ames Street and the location where the user was identified –the shaded area, North of the river, corresponds to the MIT campus.

The algorithm was modified and improved to emend this problem. Locations are now identified solely when arrived at and not when departed from. So from the data in the first example, (4223.092N 7104.641W) would be considered a location to analyse, however (4221.867N, 7103.786W) would be ignored. This means that it takes the agent twice as long to identify a location but the false ones formerly recognized were eliminated. If previously your "home" location could be identified in two days, that is, after a sequence of arrive-depart-arrive or depart-arrive-depart, with the new algorithm it takes a sequence of three arrivals.

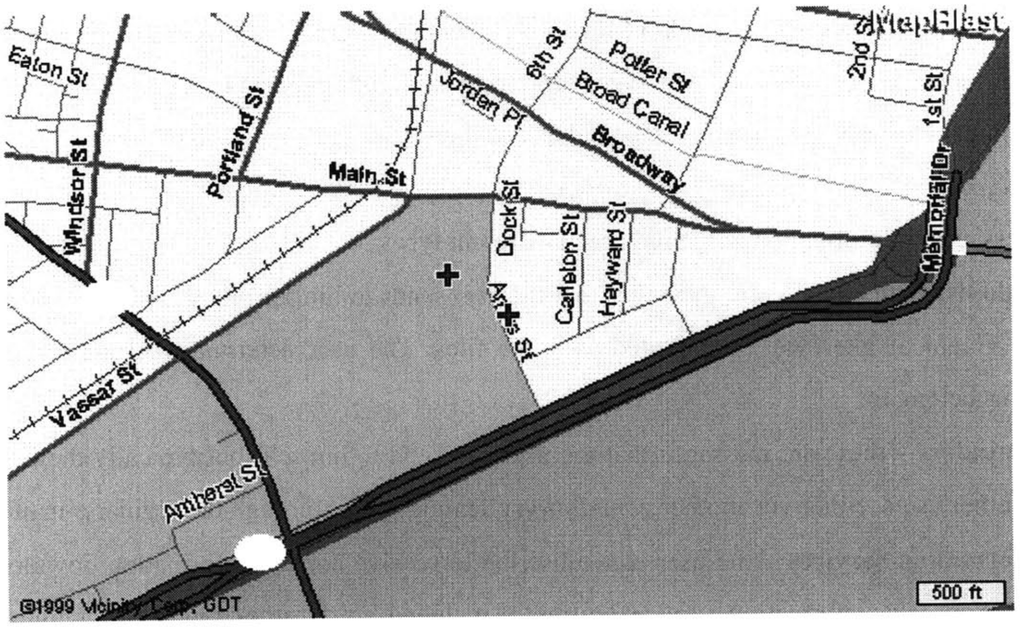


Figure 4.5: Map showing distance between location of departure (Media Lab, on Ames St) and location where identified by the GPS receiver (part way across campus, the grey area)

It will be necessary to further modify the learning algorithm since not all buildings were found to be GPS-opaque, that is, the GPS signal is sometimes received while within a building. Therefore, the data must also be analysed for stationary points. Even when a GPS receiver is static, the data indicates fluctuations of many metres. Hence the acceleration will also have to be considered to differentiate between a static receiver showing fluctuations and real fluctuations occurring due to movement in adjacent locales. This analysis to identify a stationary receiver is also necessary for the *comMotion* car architecture, where the GPS receiver will be permanently installed in the car, as opposed to in a mobile device with the user. In this architecture, it will be necessary to identify when the car is stationary, for example in the grocery store's parking lot.

Any monitoring system raises many issues regarding privacy (*cfr.* section 3.4), that is why in *comMotion* all the location tracking and analysis is done solely on the client device. Hence, the user's privacy is safe to the extent that the client device is safe.

4.2 Message engine

The user's geographic position is acquired from the GPS-receiver and analysed by the behaviour-learning agent. If the data is found to correspond to an already defined location, it is translated to the virtual location equivalent and passed on to the message engine, where the existence of relevant pending messages is checked. Messages are triggered and delivered to the user based on his geographical location, date, time of day, and day of the week.

The messages or information delivered are of different types:

- **to-do list items** – these are messages that the user sends to himself; they are either typed in, using the graphical interface, or recorded as audio files. The user determines which to-do list these items belong to.
- **reminders** – these are messages that the user can send to himself, but typically they will be sent by other users, either via another *comMotion* client device or through the regular e-mail system.
- **information services** – the user can subscribe to receive content information downloaded from the Web. The information services currently included are weather reports, headline news, and movies.
- **maps** – maps and map associated information is downloaded from the Web upon request.
- **calendar entries** – when arriving at "work" the calendar entries for that day will be delivered.

The message engine communicates with several task-specific servers (Figure 4.6), namely the *main server* through which the reminders are sent and received, the *info server* which downloads data from the Web, and the *map server* which downloads maps and locale related information.

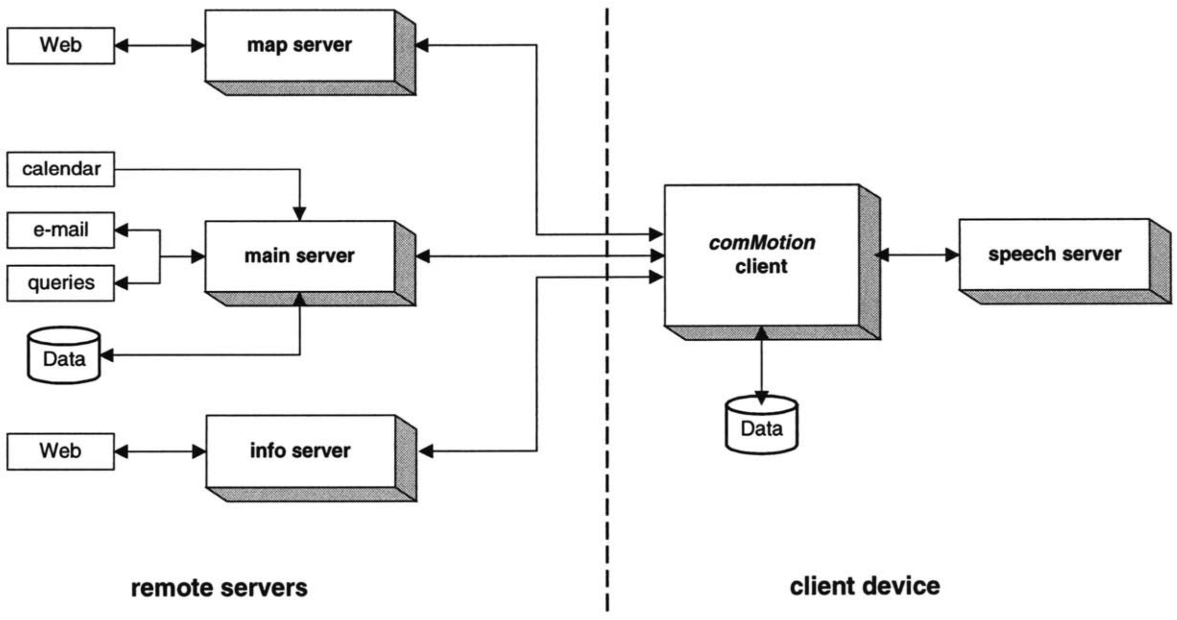


Figure 4.6: Detailed client-server architecture showing the different local and remote server processes and their communication with the *comMotion* client application

Data storage:

1. To-do lists are automatically created on the client computer when a new location has been identified and named. The to-do list items are created and saved solely on the client device.
2. Reminders are also saved eventually on the client's disk. If they are sent from a different user, either via another *comMotion* client or through the regular e-mail system, they will reach the user by way of the main server. If the user is not logged into the system, these reminders will be held on the remote server until they can be delivered.
3. The remote main server has a separate directory for each *comMotion* client where queued-up reminders are stored until they can be delivered. This directory also contains a file specifying the priority level of each defined location. Priority levels can be assigned such that they are per user and per location (*cfr.* section 4.3).

The map server receives information requests from the client application. It currently taps into the MapBlast [34] database, downloads and parses the relevant HTML (HyperText Markup Language) documents and passes the requested information back to the client. This information is not being used for commercial purposes, in which case rights would have to be acquired.

Similarly, the information server taps on-line Web databases to retrieve data such as weather reports, headline news and movie listings. Both the map and the information servers, since they download and parse publicly accessible and changing HTML, can be quite brittle.

4.3 Query engine

A user can be queried via a *comMotion* client or via the customary e-mail system. The query request goes through the main server where the priorities are checked. The *comMotion* user can establish priority levels per location and per querying user. If the query request was sent by someone with privileges, that person will receive the location information, for example, "user-X is at home". If queried by someone without privileges, or if the user is in an "unprivileged" location, the response will be "user-X is incognito". In either case, the queried user can log by whom and when location information was requested.

At present, the priority levels can only be defined by editing a file on the server; that is, there is no user interface for this task. By default, no priorities are given unless the specific location is manually added to the file and the priority stated. In future versions, an appropriate interface enabling simple defining of priorities will be included. Likewise, this version does not have a specific interface to view the log file containing information of whom requested to know the user's location. Currently, it can only be seen through a text editor.

5. *comMotion* client interface design

Chapter three discussed some of the design issues to be considered when building a system such as *comMotion* and chapter four laid out the architecture and the back end of the program, for example, the location learning algorithm. This chapter will give a detailed description, per module, of the interface, for instance, what the user will experience once a location has been identified.

An interface will appear simple and intuitive if it matches a mental model shared with the user. The simpler it appears to the user, the more powerful and useful it will be. *comMotion* has a multi-modal interface and it can be operated in both modes seamlessly. The GUI implemented has a simple metaphor which mimics its real world parallels. Finding a metaphor or mental model for an audio interface is much more intricate.

5.1 Behaviour-learning module

The behaviour-learning agent progressively learns the important locations of the user's life (*cfr.* section 4.1 for learning method). This feature eliminates the need for filling out extensive forms and providing this data before using the system, that is, no boot-strapping is required. Once it has learned the important virtual locations, such as "home" and "work", it uses this knowledge to trigger relevant information to the user, in the form of to-do lists, reminders or virtual Post-it notes, as well as Web content information the user has subscribed to.

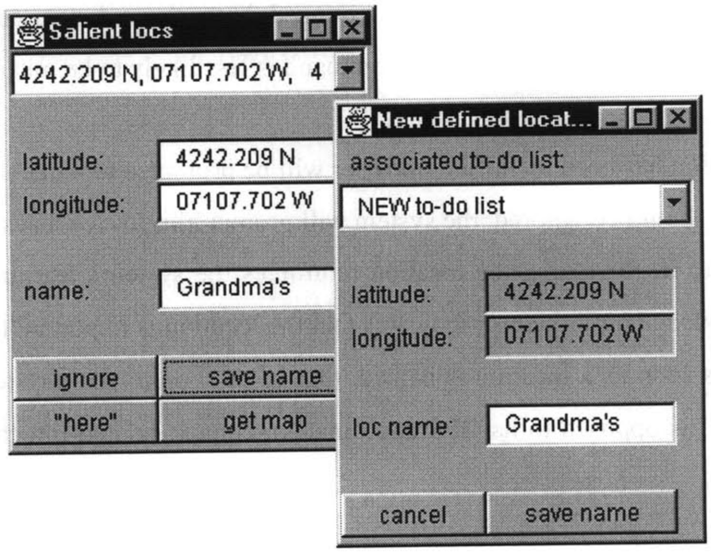
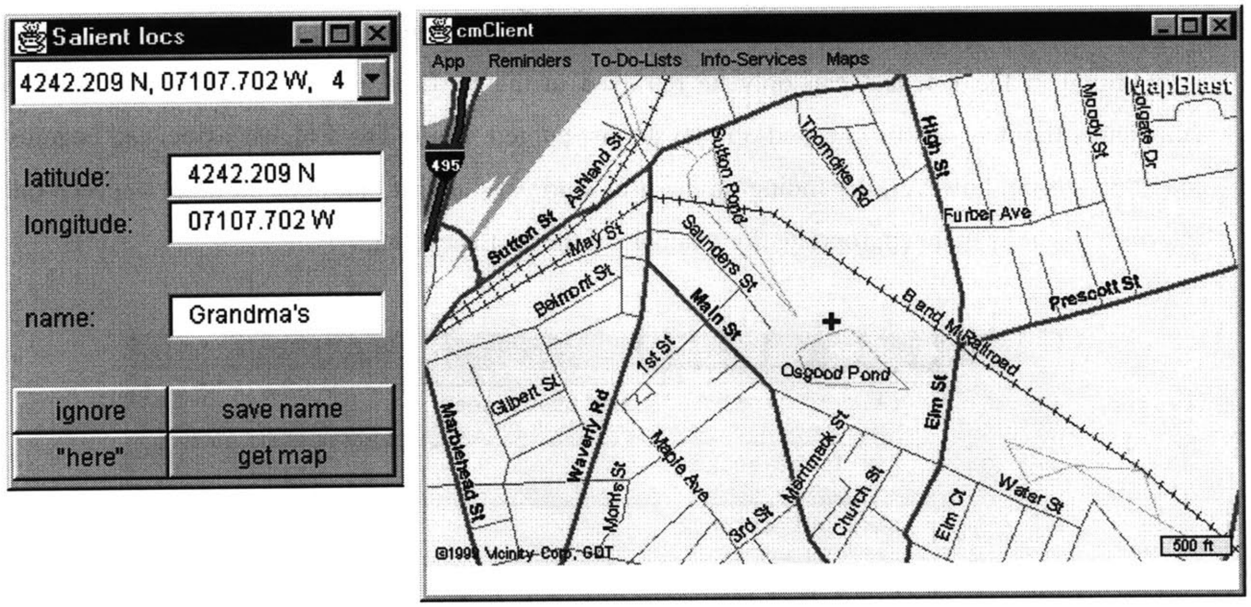


Figure 5.1: Visual interface components used to name a virtual location

The user's position (latitude and longitude) coordinates are obtained via a GPS receiver and are then analysed to determine salient locations. Once a frequented site has been found, the user is prompted to tag it with a virtual location name such as "home", "grocery store", etc. The system initiates the speech dialogue with: "This is a frequented location. Would you like to name it?". A negative answer (speech command: "no" or "nope"), or no answer whatsoever, will terminate the dialogue, whereas a positive answer (speech command: "yes", "yep", "sure" or "okay") from the user will cause a visual component to appear (Figure 5.1), through which the user specifies the location name and associates a to-do list with it. A visual component is essential since the location name must be text, and not audio. By default, the associated to-do list name is identical to the location name, however, an existing to-do list can also be chosen, since these lists can either be site specific or shared. Typically, a user who shops at three different grocery stores would probably only want one shared grocery list. On the other hand, places such as "home" and "work" would presumably have very different tasks associated with them, and, therefore, separate to-do lists would be preferable.

The tagging can also be done at a later stage, in which case the user can see the exact location on a map (Figure 5.2). Not only the corresponding latitude and longitude coordinates, but also the number of times the agent has "seen" the user at this location, are displayed. However, if the location is of no interest the user can indicate that it is to be ignored. For example, a frequented T-stop (metro) would be identified by the agent but the user would typically not want to have context-related information delivered there. Indicating that a location is to be disregarded is done by pressing the ignore button in the graphical interface or by saying "Ignore this location", when prompted for a location name. In this case, a simple two-word speech command was not used, but rather a longer command was chosen in order to minimize mis-recognitions –a longer command is less likely to be mistaken for a shorter one.

Once a salient location has been identified, the user will be prompted to name it. If he neither names it nor indicates that it should be ignored, the system will prompt him for feedback, again and again, each time he goes back to it. Revisiting the location reinforces the system's learning, however nothing is done with this knowledge until the user indicates that the location is important (by naming it) or that it is to be ignored. As soon as a location is named, that is, as soon as the physical location becomes a virtual one ("home" as opposed to its GPS coordinates), context-related information can be received there.


Salient locs

4242.209 N, 07107.702 W, 4

latitude: 4242.209 N

longitude: 07107.702 W

name: Grandma's

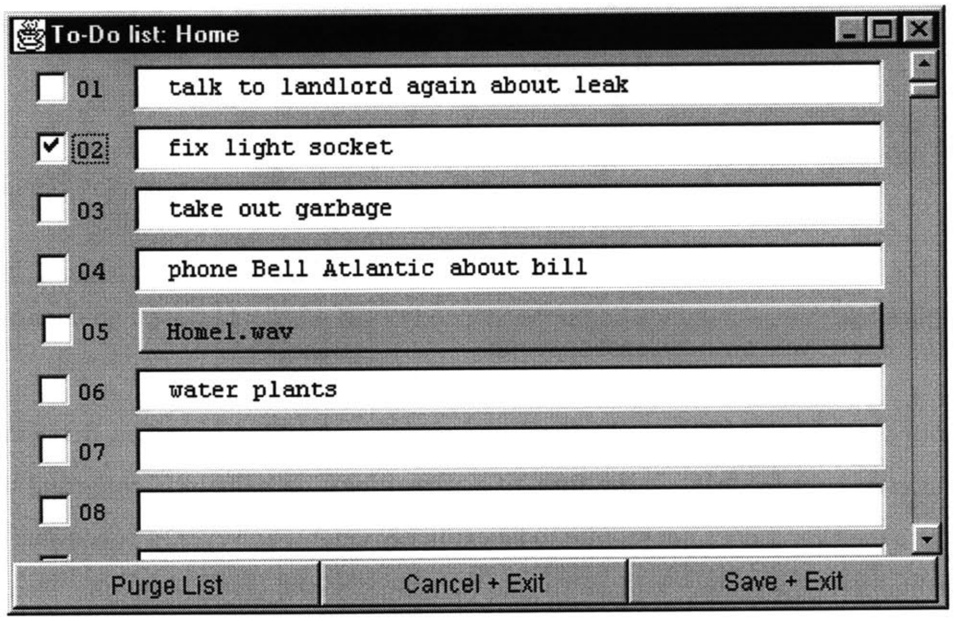
Figure 5.2: Visual interface components used to see a salient location on a map and tag it with a virtual location name

The behaviour-learning agent observes the user's mobility data and identifies frequented locations, simply requesting user feedback in the form of a virtual location name or a command to ignore; that is, it learns on its own. It does not require to be explicitly taught, however, it is possible to do so. The user, for example when visiting a place for the first time, could simply press the "here" button and name the location.

5.2 To-do lists

A to-do list is associated with each defined virtual location. Visually, the list resembles its real world paper parallel, on which items are simply listed and can be checked once they have been done. However, in *comMotion*, the to-do items can either be text, typed in using the graphical interface, or recorded audio.

When the user is in the relevant location, he will hear an auditory cue indicating that he has items on the associated to-do list and the visual component will be displayed (Figure 5.3). Since the auditory cue should alert the user without totally distracting him from his primary task, it should be subtle enough to attract attention but still remain in the background; preferences may vary and therefore the cue can be chosen by the user. The list items can be viewed and scanned using the visual interface or scanned and heard using the speech interface. Text items will be synthesized to speech, whereas audio items will simply be played. The recorded audio items are not transcribed to text, rather they appear as a button in the list, and can only be retrieved in the form of audio. Once the task has been completed, the item can be checked-off and the list purged. Audio has a higher associated cognitive load than vision, however, depending on the user's primary task at that point in time, a speech-only interface might be more appropriate, for example, if the user is cycling.


To-Do list: Home

☐	01	talk to landlord again about leak
✓	02	fix light socket
☐	03	take out garbage
☐	04	phone Bell Atlantic about bill
☐	05	Home1.wav
☐	06	water plants
☐	07	
☐	08	

Figure 5.3: A to-do list with both text and audio items

The speech commands used to navigate within the list are simple two-word commands which should be intuitive to the user: "first item", "last item", "next item", "previous item" and "read item". If operating in verbose-mode, after each command the system will repeat the command before actually performing it. The command "check item" is used to toggle the check mark next to an item. If operating in verbose-mode, after performing the command, the system will say, for example, "Item: fix light socket, is now checked". The verbose feedback is useful for the user to gain confidence in the system and to understand the system's behaviour in the case of mis-recognitions when a command is misunderstood for another. If the command is not recognized, the system will answer: "Say that again".

The audio items can be recorded using either the visual interface (Figure 5.4) or through speech commands (Table 2). It is only necessary to choose the to-do list the audio item is for, that is, no file names must be specified. In the graphical interface, the user can start and stop the recording, and play back the audio until satisfied with the recording. Saving the file will automatically insert an audio item button in the specified to-do list, enabling future playback.

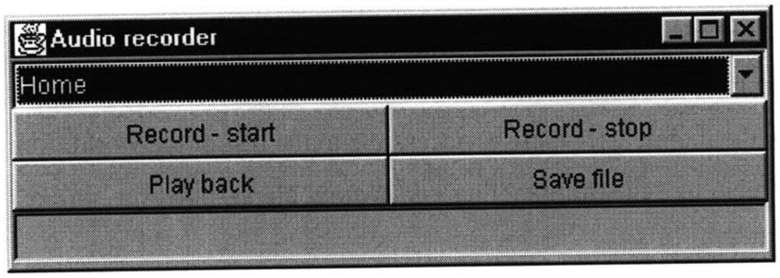


Figure 5.4: Visual interface component used to record audio to-do items

The user can initiate the dialogue to record an audio item by saying: "Choose list" (Table 2). The system will then recite the names of the lists, one at a time, until the user says "yes" to one of them. If the user does not want a specific list, he can say "no", or simply say nothing –if no response is heard after five seconds, the next list name will be recited. When recording an audio item via the speech commands, it is not possible to stop the recording once it has started. The system will automatically record during 20 seconds. An auditory cue, a beep, is heard just before the recording begins, and once again after the 20 seconds timeout.

Speaker	Speech	Explanation
User: CM:	"Choose list" "Say 'yes' when you hear the list you want, otherwise say 'no'." "Bank"	List name
User:	"No"	
CM:	"Book store"	List name
User:		User doesn't answer
CM:	"Grocery store"	
User:	"Yes"	
CM:	"You chose the Grocery store list" "Start recording after the beep" *Beep*	Beginning of recording
User:	Audio recorded during 20 secs.	
CM:	*Beep* "You may play back and save the recording"	End of recording
User:	"Play back recording"	
CM:	Recorded audio is played back "You may save the recording"	
User:	"Save recording"	Item inserted in list

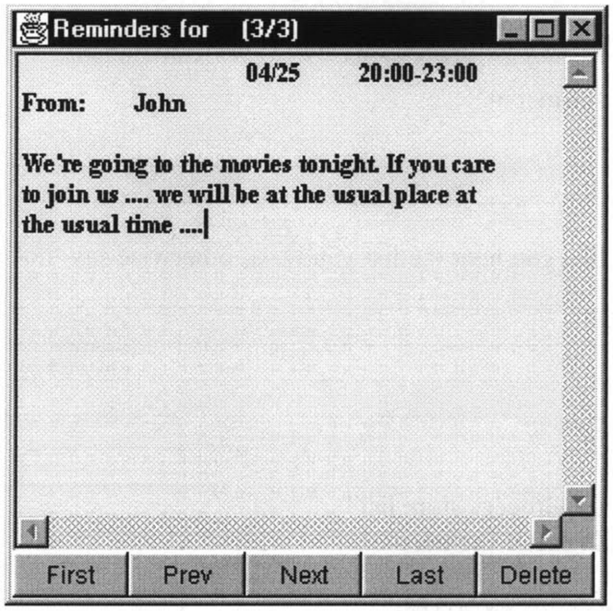
Table 2: Dialogue between *comMotion* (CM) and the user showing how audio to-do items are recorded using speech commands

5.3 Reminders

The metaphor used for the reminders is the widespread 3M Post-its™; the visual component resembles them in shape, size and colour (Figure 5.5).

Context-aware reminders for *comMotion* mobile users can be sent via the regular e-mail system. This is important since the mobile user must be accessible to the rest of the world. Reminders, or electronic Post-it notes, can also be sent from another *comMotion* client, however, unlike the to-do items, they are text-only.

The reminders are posted to a specific location and can be constrained to a certain date or date and time range. When the user is in the relevant context, he will hear an auditory cue indicating that he has some relevant reminders and a visual component resembling a Post-it note will appear on the display (Figure 5.5). These notes can then be viewed and scanned using the visual interface or scanned and heard as synthesized audio by means of the speech interface.



04/25 20:00-23:00
From: John

We're going to the movies tonight. If you care to join us .... we will be at the usual place at the usual time ....

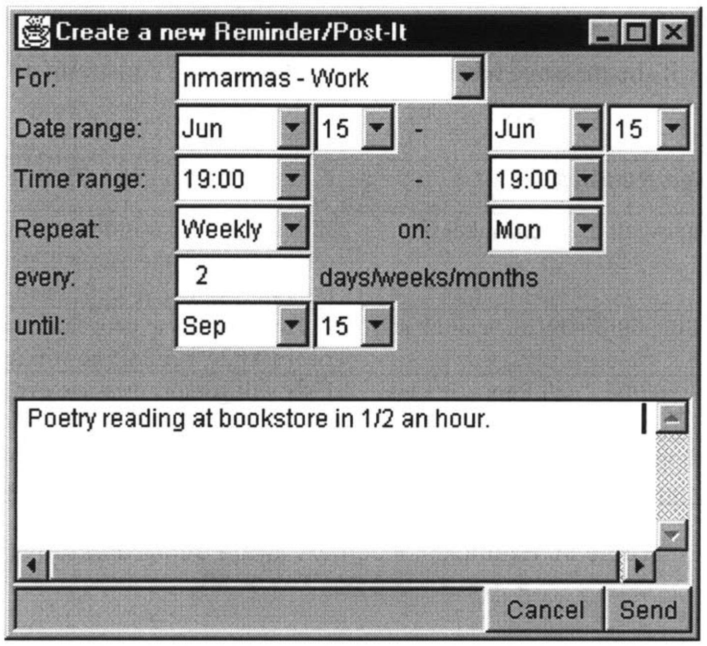
Figure 5.5: Visual interface component to view and scan reminders

The speech commands used to navigate between the reminders are: "first reminder", "last reminder", "next reminder", "previous reminder", "read reminder" and "delete reminder". If operating in verbose-mode, after each command the system will repeat it before actually performing the command.

When creating a new reminder, it is possible to have it automatically repeated in the reminder file. To do so, the user must specify:

- The frequency of repetition – daily, weekly, monthly, or none;
- the repeat increment;
- the day of the week; and
- the date range.

For example, you could create a reminder for a certain event that takes place on Mondays, every other week, for the next three months. All these parameters can be defined whether sending the reminder from a *comMotion* client or via e-mail, however, the *comMotion* client provides a more convenient interface for this task (Figure 5.6). In the example below, the user will receive the reminder on June 15th at seven o'clock in the evening, and subsequently every other week, on Mondays until September 13th (the last Monday before September 15th).



For:	nmarmas - Work			
Date range:	Jun	15	-	Jun 15
Time range:	19:00		-	19:00
Repeat:	Weekly		on:	Mon
every:	2	days/weeks/months		
until:	Sep	15		

Poetry reading at bookstore in 1/2 an hour.

Figure 5.6: Visual interface component used to create a new reminder

When sending a reminder via the *comMotion* interface, both the user and the location are chosen from a list, which is composed of all this particular user's locations, as well as all the locations of other *comMotion* users to which he has privilege.

Format of a context-aware reminder sent by e-mail:

user, location, time range, periodic, date range, msg

where:

user - is the recipient operator

location - is the virtual site as defined by the recipient

time range – is the time span validity of the message; from time-to time (hh:mm-hh:mm)

periodic – is the recurrence: daily-x / weekly-x / yearly-x (default x=1)

date range – is the date span validity of the message; from date-to date (mm/dd-mm/dd)

msg – is the text of the reminder

If the format cannot be parsed out correctly, the e-mail sender will receive a reply indicating the problem and the correct format to be used.

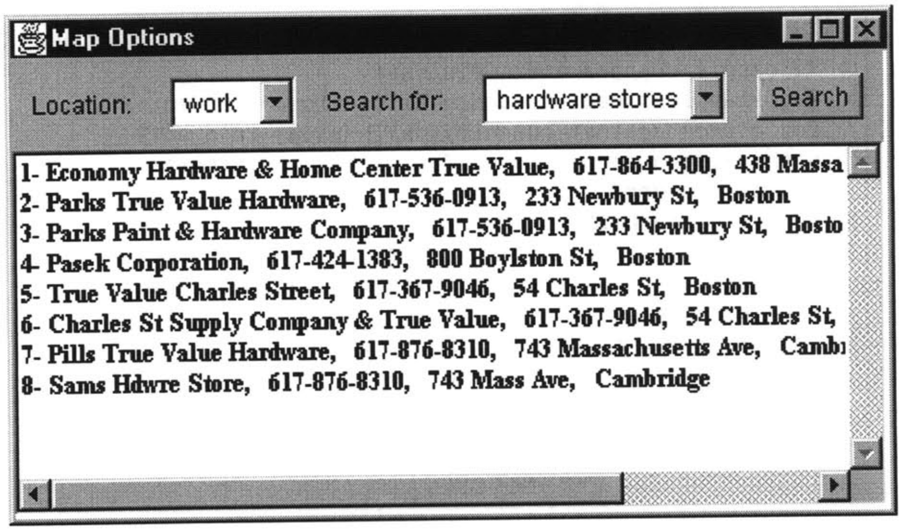
The table below (Table 3) shows examples of context-aware reminders sent by e-mail and how the different parameters affect the delivery of the messages. If a parameter is irrelevant, it can simply be omitted, for example, if the message has no location, time or date constraint as in the fourth example.

Message/Reminder	Delivery
Lucienne, grocery store, ,week-2, , "remember the special cat food"	If this reminder was defined for Mondays, then every other Monday, the user will get the message when in the proximity of her grocery store.
Tom, home, 16:00-23:30, , 09/27-09/28, "call N. at the office"	If Tom gets home between four and eleven thirty, on February 27th or 28th, he will receive the reminder.
nmarmas, work, , year-1, 07/06, "call Rapi – it's his birthday"	User will get this message every year on July 6.
Alan, , , , , "get in touch with John Smith ASAP!!"	User will get this message immediately, since no constraints were placed on time, date or location.

Table 3: Examples of context-aware e-mail messages

5.4 Map module

At any given time a user can view a map of his current location together with neighbourhood locales, such as banks, movie theatres, grocery stores, etc. This information is acquired from MapBlast's [34] Web database. Besides current location, it is also possible to choose one of the already defined virtual locations. The information received is a list of the eight closest requested locales together with their address and telephone number (Figure 5.7). In addition, a map is displayed showing the current location and the position of the eight sites (Figure 5.8).



1- Economy Hardware & Home Center True Value, 617-864-3300, 438 Massa
2- Parks True Value Hardware, 617-536-0913, 233 Newbury St, Boston
3- Parks Paint & Hardware Company, 617-536-0913, 233 Newbury St, Bosto
4- Pasek Corporation, 617-424-1383, 800 Boylston St, Boston
5- True Value Charles Street, 617-367-9046, 54 Charles St, Boston
6- Charles St Supply Company & True Value, 617-367-9046, 54 Charles St,
7- Pills True Value Hardware, 617-876-8310, 743 Massachusetts Ave, Cambı
8- Sams Hdwre Store, 617-876-8310, 743 Mass Ave, Cambridge

Figure 5.7: List of eight closest requested locales, shown on accompanying map

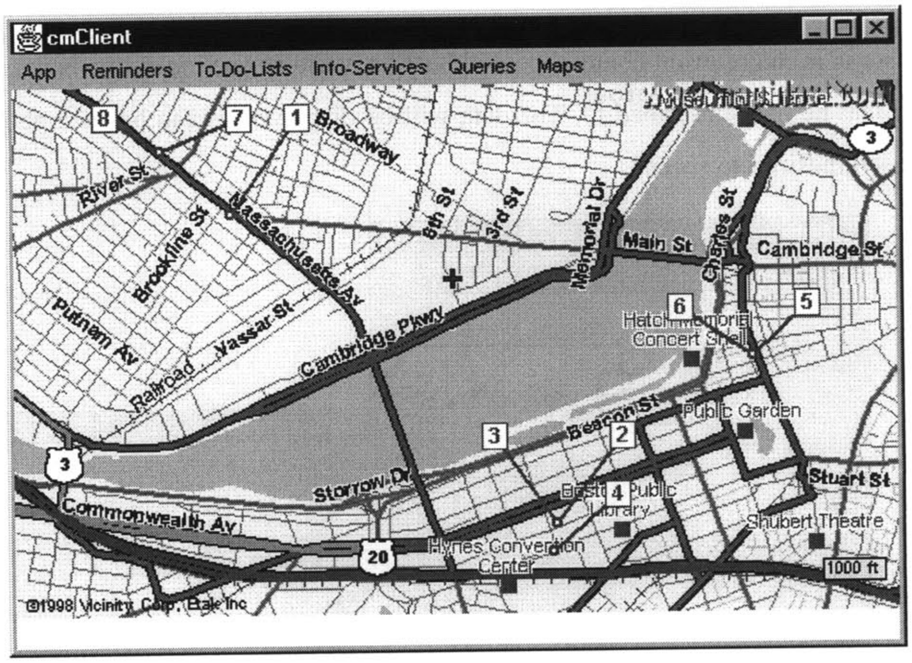
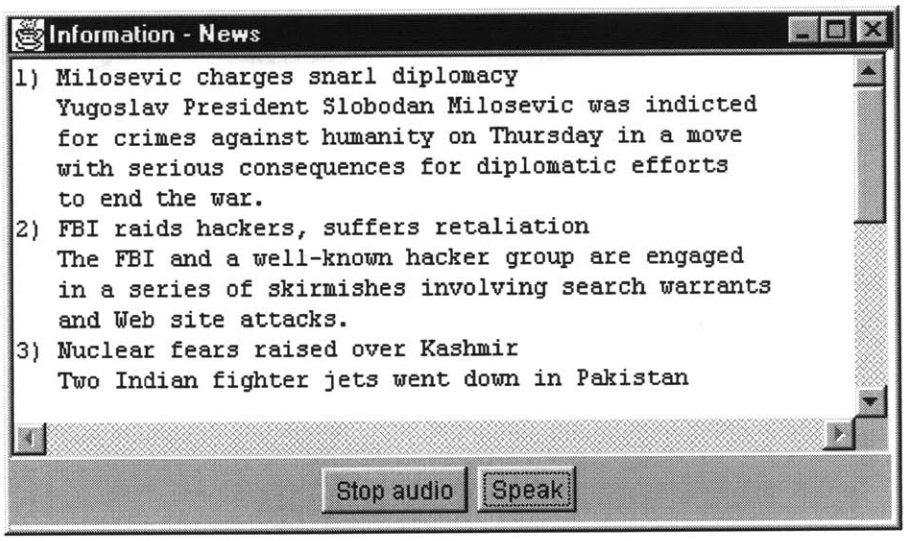


Figure 5.8: Map indicating current location and eight closest requested locales

5.5 Information services

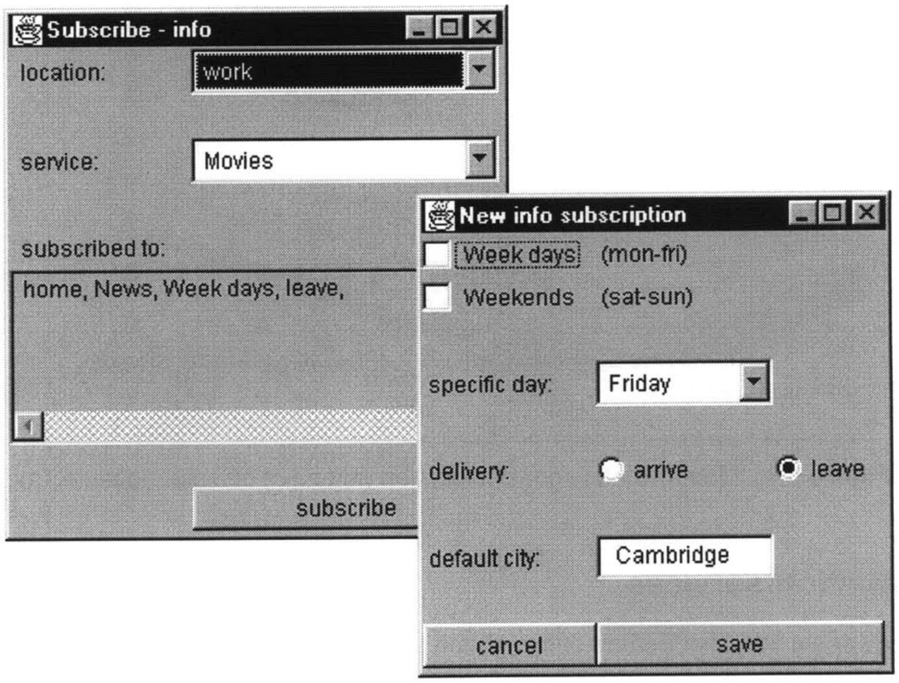
The message engine can also deliver content such as headline news (Figure 5.9), weather reports and current movie listings from information sources on the Web. As the requested information might vary depending on context, the user can subscribe to different information services based on location, and special schedules can be made for different days. For example, the user might request to receive a list of the movies showing at the local cinemas when leaving work on Fridays (Figure 5.10).



```
1) Milosevic charges snarl diplomacy
   Yugoslav President Slobodan Milosevic was indicted
   for crimes against humanity on Thursday in a move
   with serious consequences for diplomatic efforts
   to end the war.
2) FBI raids hackers, suffers retaliation
   The FBI and a well-known hacker group are engaged
   in a series of skirmishes involving search warrants
   and Web site attacks.
3) Nuclear fears raised over Kashmir
   Two Indian fighter jets went down in Pakistan
```

Figure 5.9: Visual component displaying headline news

When the user is in the relevant context, he will hear an auditory cue indicating he has location-based information that he has subscribed to. The data will be displayed as can be seen above, however, it may also be synthesized to speech.



location: work
service: Movies
subscribed to:
home, News, Week days, leave,

[ ] Week days (mon-fri)
[ ] Weekends (sat-sun)
specific day: Friday
delivery: ○ arrive ● leave
default city: Cambridge

Figure 5.10: Interface component used to subscribe to location-based content information

Headline news is downloaded and parsed from the msnbc.com site; weather reports are taken from wunderground.com, and the movie listings are from the sidewalk.com site. The default city for the weather information is Boston but this can be modified by the user. Likewise, the default city for the movies (Cambridge) can be changed. The weather site supports most large cities in North America as well as several others around the world. Nevertheless, the sidewalk.com site only supports a subset of the larger, or most important cities in the US. The data is parsed out and formatted for the visual display. Although the text can be synthesized to speech, its format was not specifically adapted for that purpose.

5.6 Query module

A user can be queried via a *comMotion* client or through the regular e-mail system. In either case, the request goes to the main server where the priorities are checked. A *comMotion* user can establish priority levels per location and per user; at present this must be done by editing a file on the remote server. Querying from a *comMotion* client is done by choosing from a list of other *comMotion* user names (Figure 5.11) and the reply is consequently displayed. If the request was sent by an authorized person, a reply such as "is at work" will be displayed below the user's name; however, if queried by someone unauthorized, the response will be "is incognito!" (Figure 5.11).

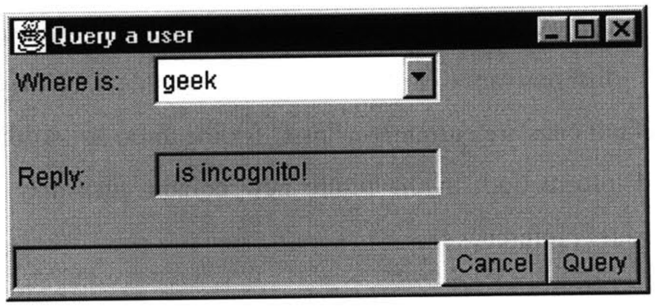


Figure 5.11: Visual interface component used to query the whereabouts of another user

A query request via e-mail is performed by sending a message, to the *comMotion* user who is being queried, with "cm query" as the subject. If the request was sent by an authorized person, he will receive an e-mail in response containing, for example, "user-X is at home" or, if unauthorized, "user-X is incognito!"

The queried user can log by whom and when location information was requested. The log file includes: by whom he was queried, where the queried user was at the time, what information was released, and the date and time of the query. This version of *comMotion* does not include a specific interface for viewing the log file, however it can be viewed through any text editor.

5.7 Speech UI

Although speech has many limitations (*cfr.* section 3.1), speech interfaces can be effective and beneficial for specific tasks or under certain circumstances; for example, while driving a car or cycling. *comMotion* uses a combination of different speech technologies to enable access to the main features using audio as both input and output. The speech interface does not replace the graphical one but rather complements it and grants the user functionality in circumstances where his hands and/or eyes are busy.

A good interface should be coupled with a user's mental model, hence making it intuitive. Finding a metaphor for an audio interface is much harder than finding one for the visual equivalent. For example, an effective visual metaphor for a reminder is a component which mimics the common 3M Post-its™. But how can this be represented in audio, or what alternative acoustic model can be used?

Different auditory cues are associated with the various types of information (to-do lists, reminders and subscribed content information). Whenever the user has pending messages relevant to his context, the pertinent audio cue is heard. As the purpose of these cues is to alert the user without distracting him from his primary task, different ones can be chosen and in this way they may be as subtle as he wishes. The system default cues are currently a "psst" for the to-do list, a "ding" for the reminders and chimes for subscribed information. In the future new default cues may be chosen, based on the consensus of several users' preferences.

The user interacts with the system via speech commands, which are mostly two-word commands and should be intuitive (Table 4).

	Function	Command	Result
1	Navigation between different to-do lists	"go to item list"	The system will recite, one by one, the different list names until the user answers "yes"
2	Browsing to-do items within a list		
		"read item"	TTS or audio playback of the to-do item, that is, delivery of the item

		"first item"	Delivery of first item on list
		"last item"	Delivery of last item on list
		"next item"	Delivery of next item on list
		"previous item"	Delivery of previous item on list
		"check item"	Toggles the check field of the item
3	Navigation between different location reminders	"go to reminders"	The system will recite, one by one, the different list names until the user answers "yes"
4	Browsing reminders of a specific location		
		"read reminder"	TTS of reminder text, that is, delivery of the reminder
		"first reminder"	Delivery of first reminder
		"last reminder"	Delivery of last reminder
		"next reminder"	Delivery of next reminder
		"previous reminder"	Delivery of previous reminder
		"delete reminder"	Deletes the reminder
5	Recording an audio to-do item	"choose list"	The system will recite, one by one, the different list names until the user answers "yes"
		"start recording"	The system will play a beep and then record during 20 secs.
		"play back recording"	The system will play back the last recorded file
		"save recording"	The system will insert the recorded audio item into the relevant to-do list
6	Affirmative answer	"yes", "yep", "sure", "okay"	
7	Negative answer	"no", "nope", or the absence of an answer	
8	Speech control	"speak faster"	Speeds up speech
		"speak slower"	Slows down speech
		"speak louder"	Increases volume
		"speak softer"	Reduces volume
		"reset speech levels"	Resets speech volume and speed levels to defaults
9	Ignore location	"Ignore this location"	System does not include location in mobility analysis, from then on

Table 4: Speech commands supported by the system

comMotion can be run in terse or verbose mode. When in verbose mode, the user will receive much more auditory feedback:

- After each speech command is recognized the system will repeat the command before performing the task. For example, if the user says "read item", the system will repeat "read item", before it synthesizes or plays back the relevant to-do list item.
- Once an auditory cue has been given, it will explain the reason for the cue. For instance, "You have five items on your grocery list".
- When choosing one of the locations from the list of names, after the user says "yes", the system will respond, for example, "grocery list".

Speech recognition is fragile; speech commands might not be recognized or they could be mis-recognized causing an unwanted action to be performed. Unlike in a visual interface, where the user can generally perceive what the system is doing, an audio interface is not transparent to the user. Therefore, explicit feedback is beneficial, especially for novice users. For example, when marking a to-do item as checked: in a visual interface the user will easily see if the wrong item was ticked, however in a speech interface he has no way of knowing unless the system gives explicit feedback, or he has enough experience with the system to trust it.

In *comMotion,* speech recognition is not continuously enabled since ambient conversation is often picked up and misinterpreted as direct input. However, since the objective of the speech interface is to enable system functionality in a hands and/or eyes busy situation, the user cannot be expected to manually activate the recognition by means of a push-to-talk button. Consequently, *comMotion* automatically activates the continuous recognition when it expects to receive speech commands. As long as the user interacts with the system, the continuous recognition remains enabled. For example, if the system alerts the user (with an auditory cue) regarding pending messages, the continuous speech recognition is activated, giving the user the opportunity to ask for the messages to be read. As long as the user gives speech commands, the recognition stays on. However, unanticipated speech interaction requires manually initiating the recognition. Once it has been activated, it will remain enabled as long as it picks up audio.

The following chapter describes some preliminary user evaluations and suggests ways to improve both the graphical and the speech interfaces.

6. Conclusions

A first prototype of *comMotion* was built in order to evaluate its feasibility and usefulness. This chapter presents some preliminary user evaluations of the prototype, by both technical and non-technical minded users. Furthermore, the contribution of this project is pointed out and future work to be done on the system is suggested.

6.1 Evaluation

For *comMotion* to be most effective, it should be used on a regular basis over a period of time. The system must first learn the user's frequented locations as only after these virtual sites have been established can context-related information be delivered. Therefore, two different parts of the system must be evaluated: the location learning feature and the delivery of information.

6.1.1 Location learning

GPS data was collected over a couple of months by two different members of the speech group. This data was used to evaluate and make iterative improvements to the location learning algorithm. For instance, *comMotion* initially identified locations based on when the user arrived and when he departed; however as GPS can take several minutes to regain its location after leaving a building the user was often at quite some distance. This gave rise to problems so the algorithm was modified and now locations are only identified when arrived at. The problems encountered and how they were resolved are described in detail in section 4.1. At present, the *comMotion* learning algorithm effectively identifies GPS opaque buildings after they have been visited three times.

6.1.2 Information delivery

Preliminary tests to evaluate the effectiveness of information delivery were carried out and both the graphical and speech user interfaces were assessed. Three subjects were given the same task to perform and feedback was received from them both during the assignment and afterwards.

The *comMotion* system was taught three virtual locations: a local post office, a bookstore and a bank; all are within the same two blocks. Three to-do list items were created for the post office, two for the bank (on separate lists) and one to-do item and a reminder was sent to the bookstore. The users were informed that the system had already learned the locations and that they had pending items on their to-do lists specific to each location. They were asked to walk in the general area of the different locales, choosing whichever route they preferred.

User-A

The first evaluation was done by User-A, who is a member of the speech group and therefore has a prior understanding of the system and the concepts implemented.

User-A indicated that he would like a little more description in the alert received as he approached a building. Instead of *psst* (audio cue), "You are near the bank", he would have preferred something like "You have two messages for the bank". This was later implemented and the system, when operating in verbose mode, now says for example: "You have three items on your bank list and two reminders". When in terse mode, only the audio cues are heard, without an explanation as to why the alert was given.

He also stated that he would rather have one set of speech commands to navigate the two different kinds of messages (to-do list items and reminders), instead of having to decide which command is appropriate according to type of audio cue. The to-do lists and the reminders have different metaphors, as is quite apparent in the visual interface, where the former resemble a paper checklist and the latter look like a 3M Post-it™. These are two different paradigms and they serve different purposes: the to-do lists contain items (text or audio) which are from the user himself, while the Post-it reminders are typically from others. A user can also send himself a Post-it reminder but this would probably have a different meaning than a to-do item; for example, something more urgent. As has been mentioned before, conveying the different metaphors in an audio only interface is difficult. A unique set of commands to navigate both types of reminders could be added. Either the user could indicate which type of messages he wanted, for example, "Go to my list" or "Go to my reminders", and navigation within the two could be achieved with the same set of commands. Alternatively, for the speech interface, the two could be concatenated and hence navigated as one single list; first the reminders would be read and then the to-do items, or vice-versa.

Furthermore, he suggested it might be useful to prioritize the individual messages. For example, if you have important messages you should be given an alert when you are outside or near the location: "You have two important messages for the bank". On the other hand, lower priority messages would only trigger a reminder if you actually entered the location. The idea is that there are certain messages which should cause you to alter course and go to the location, and others are perhaps only useful if you actually go the location, but should not remind you to go there if you were not planning on it already. Moreover, he noted that it was a subtle thing and was not completely sure it would be all that useful.

For most of the time the user wore the system in a pouch hanging from his belt. He did not find it to be uncomfortable, however, he said he would only use such a system, on a full time basis, once the hardware were smaller (perhaps the size of a PalmPilot or a cellular phone) and with a wireless ear-phone. During the evaluation, some GPS shadowing from buildings was experienced, as well as overlapping virtual location boundaries (*cfr.* section 3.5).

User-B

The two other users are non-technical people and neither of them has much experience with mobile computing devices.

The system was described to User-B and he was shown a simulation, before actually experiencing it on his own. Although he had voluntarily agreed to test the system, once on the street he was reluctant to appear to be talking to himself or to a wired microphone going to an unseen device. He asked to remove the pouch, wired microphone and speaker, and simply walk with the computer, as if he were carrying a book (in this case a PC-card GPS would have been much more practical). When an auditory cue was received, he used the visual interface on the pen-based computer to view the to-do items and even marked some of them as done. He stated that he saw that such a system could be beneficial but he could not really see himself carrying "all that stuff" around. Nonetheless, he liked the idea of the car architecture and the possibility of simply placing a PalmPilot near the computer in the car and having the list transferred to it –he could definitely see himself using that.

User-C

The system was described to User-C. He did not want to see the simulation, rather just try it. Initially he used the system in its hands/eyes busy mode, that is, with just the speech interface. This proved to be somewhat frustrating. It must be noted that English is not his native language and not having a "standard American accent" makes it very hard to have commands recognised. Consequently, he used the visual interface. The user seemed to enjoy seeing the visual components pop-up on the screen and receiving the auditory cues. He started testing approaching the locales from different directions and seeing how that affected –since the bank and the bookstore are very close to each other, sometimes the system gave alerts for the back and other times for the bookstore (*cfr.* section 3.5 for explanation of why this happens). He then requested to try out the speech only interface again. Whenever an auditory cue was received and he wanted to navigate within the relevant to-do list, he asked that I speak the command to the system –voice recognition improved tremendously. He stated that the system could be very useful to him, albeit the lack of precision in position information, and he would be willing to "learn how to talk to it". He was not sure about the wires but would be willing to try a wireless microphone and speaker version. He suggested having some tactile cue, for example a vibration instead of or in addition to the auditory cue. His concluding comment was: "If you could just teach it to know exactly where I am, and to understand when I speak, it would be just perfect!". (That of course is easier said than done.)

These results are not conclusive and more extensive evaluation must be done, however it seems clear that there is place and desire for such a system. For *comMotion* to be really beneficial some of the problematic issues must be resolved. Primarily:

- **precision of position** – more accurate position information could be received with DGPS, however this may be solved by the software when improvements, such as virtual location clustering (*cfr.* section 3.5), are added.
- **timing of alert** – the users' mode of locomotion should be considered. The radius of the virtual location boundary should be a function of the speed of travel. The faster you are going, the larger the radius should be. Hence the auditory cues could be given at a delta time from the location.
- **hardware** – smaller and lighter hardware are essential in order to make the system truly mobile.

6.2 Contribution

comMotion shares features with other context-aware applications, however, it can also be distinguished from them. Chapter 2 presents related research and section 2.4 gives a detailed description on how *comMotion* differs in several areas. These are summarized below:

- **learning** – *comMotion*'s most unique feature is its independent-learning property. An agent monitors the user's mobility data and learns the frequented locations. Once a salient location has been identified, the user only has to name it or indicate that it is to be ignored. *comMotion* does not rely on the user for its learning process; it can identify the places on its own, however it does not know the importance of the location or what it should be called. Only at that stage is the user's feedback required. This independent location learning is crucial since it implies that no boot-strapping is necessary; no locations must be pre-defined. Although it is possible to explicitly teach the agent, this instruction is not essential.

- **information subscription per location** – much information we consume is relevant only to specific locations. With *comMotion,* a user can subscribe to receive content information when arriving or leaving a certain location. For example, receiving a list of movies playing in the local cinemas when leaving work on Friday evenings.
- **accessibility of information** – the to-do list items and reminders are accessible to the user regardless of his place and time. In certain systems, users can only send and/or receive messages when in the relevant context; in *comMotion*, the messages are automatically triggered depending on the user's context but they can be viewed or sent from any context.
- **memory prosthesis for future events** – there are systems which remind users of past events which occurred in the same context; for example old class notes might be recalled when entering a classroom in which a class was taken. *comMotion* does not remind users of things they did in a certain place, but rather of things they should do.

6.3 Future work

The future work envisioned includes a wide range of elements, from enhancements to the interface and additional features to the broadening of the existing concept.

6.3.1 Short-term

In the short-term several improvements should be made to the existing prototype, such as:

i) Adapt the speech server to include a dynamic grammar, as opposed to a pre-defined one. When recording an item via the visual interface, the user simply chooses the location name from a list. The same function via speech commands is much more tedious since the user must listen to a synthesized list of the existing location names and answer "yes" when he hears the one that he wants. If the location names were dynamically loaded to the grammar, the user could directly say which location he was recording for.

ii) When recording an audio item via the visual interface, the end of the recording is indicated by pressing a button. When recording using only speech commands, all audio is recorded during 20 seconds, regardless of the length of the speech. The recording mechanism should be adapted to include pause detection, hence terminating the recording once the user has stopped speaking.

iii) A user's location can be queried by other users, provided they have authorization. Privileges can be defined per user and per location. An interface to enable the user to easily define and modify these privileges must be added. Likewise, he should be able to easily view by whom and when he was queried and what information was released. This information exists as a log file but the user currently does not have easy access to it.

iv) Speed of travel should also be considered as part of the user's context and the timing of the audio alerts should be given as a function of it. The slower the user is moving, the later he can be alerted, though still giving him time to react. If the user is walking, he can be alerted very close to the locale, however, if the user is driving, the alert must come much sooner, giving him time to act upon the notification.

v) Adjacent virtual locations should be clustered so that the user will get reminders for all the locations in that vicinity and not just the first one identified (*cfr.* section 3.5).

vi) The learning algorithm should be modified to include the analysis of stationary points. This will solve the problem both for the car architecture and for buildings which are not GPS opaque.

vii) More information services should be added. Besides headline news, weather and movie listings, which currently exist, the system could provide traffic reports, stock quotes and others.

viii) More user evaluation and iterative interface modification can be made even before including all of the above-mentioned changes.

6.3.2 Long-term

i) The learning algorithm should be modified to include route learning and not just frequented locations. The agent could then track the user's mobility patterns and use this knowledge to provide relevant information or suggestions. For example, suggest one of the alternative routes home if the user has to shop and the grocery store is on the other route.

ii) Route learning could be used to calculate average travel time to different way points. This knowledge could then be utilized to predict the arrival time to a destination. If the user's whereabouts were queried, the answer could be: "User-X should be home in ten minutes". Additionally, arrival times could be combined with calendar entries; for instance, if the user were going to be late to a meeting, the system could automatically send a message to his party (obtaining the e-mail address from the Rolodex).

iii) The user's preferences should be learned and the information delivery adapted accordingly. For instance, if the user never shops on the way to work, but only on the way home, he should not be alerted regarding shopping when going to work. The system must have an interface showing the rules it has inferred and permitting the user to edit or delete them, hence giving him total control. These preferences will be learned from direct user feedback, such as telling the system to shut-up, and from indirect feedback like never acting upon a certain alert. The system is multi-modal and user's preferences as to audio or visual information delivery can be learned; for example, if the user always requests to have the grocery list read to him or if items are only checked via the visual interface. Additionally, the order in which the items are checked in a to-do list can also be learned. For example, if a user always ticks milk and cheese before any other items in a grocery list, it can be inferred that they are both in the first aisle in the store. Future grocery lists could be automatically re-ordered to reflect the user's shopping habits.

iv) *comMotion* is currently a reactive system; information delivery is triggered by the user's context. The system could be made to be more proactive: it could remind the user that he has things to do at one of the locations he has not frequented in some time, or suggest a route home which passes by the hardware store.

v) Some common sense knowledge could be integrated into the system, helping it to be more proactive. For example, a reminder about a party next Friday could cause a to-do item to be added to the grocery list or perhaps a reminder for the liquor store. The system would have to be used over a period of time and the reminders then analysed to see if recurrent relevant keywords could be found.

vi) In the current prototype an e-mail reminder can be sent to a user at a specific location (providing the sender has the required privilege). The sender defines the location the e-mail message is for, that is, it is the sender who is doing the filtering and not the system. *comMotion* could be integrated to work with the Clues [22] dynamic filtering, so that all e-mail of a certain priority would be sent to the user as a reminder, or as a new data type. Nevertheless, it would be even better to have an agent which analysed the body of the incoming e-mail, and by using common sense knowledge, tried to determine which virtual location ("home", "work", etc.) the message should be filtered to. This, of course, is a non-trivial problem, but even a subset of common keywords used to filter a few of the incoming e-mails could prove beneficial. *comMotion* could also inform other agents, such as Active Messenger [26], so that the incoming e-mail could be sent to another device, for instance a pager.

vii) A smart alarm clock could be built into *comMotion*. Half an hour before it is set to ring, the system would check the weather and traffic. If both weather and traffic conditions were favourable, the clock could reset itself, letting the user sleep in X minutes more. Conversely, in bad conditions, the alarm clock could ring earlier.

Information delivery utopia would be to receive the exact amount of data we need (no more, no less), when and where we need it, and in a gratifying format –in a pleasant voice or well-structured text. Although we are still far from this goal, a *vade mecum* such as *comMotion* can help alleviate the information overload problem. The more the information can be filtered and channeled to the right location, the more effective the system will be. This is true provided it is sensitive to the user's context, for example to hands and/or eyes busy situations, and provided the interface is intuitive –a bad interface will simply add to the cognitive load. *comMotion* can be deployed across various architectures and adapted for users with different needs and different life-styles. Therefore, it can truly be a constant companion, helping users keep a handle on all they have to do and providing them with timely information. *De comMotionis nil nisi bonum.*

Appendix A – colour plates

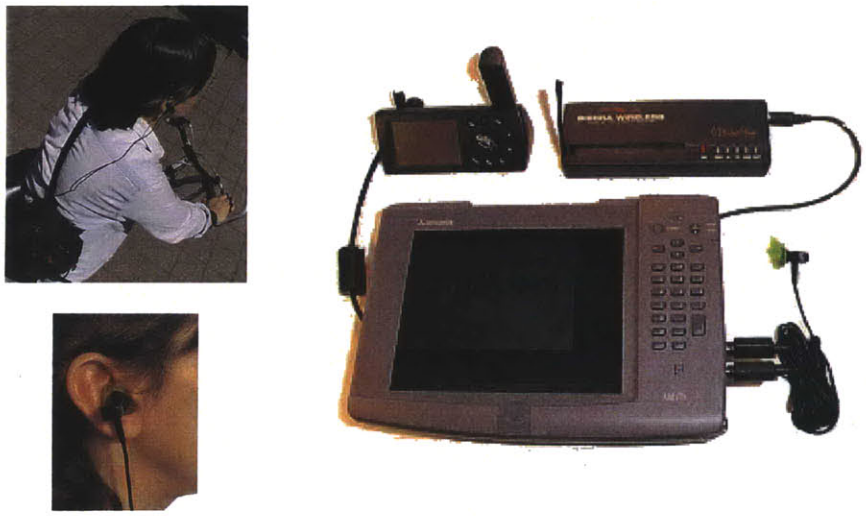


Figure 4.1: Hardware components of the *comMotion* system

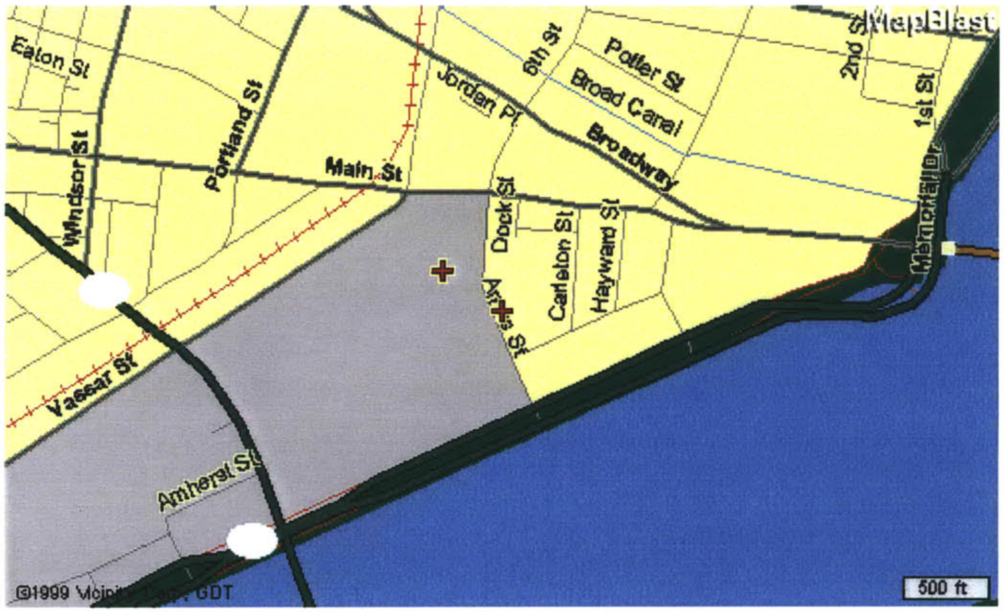
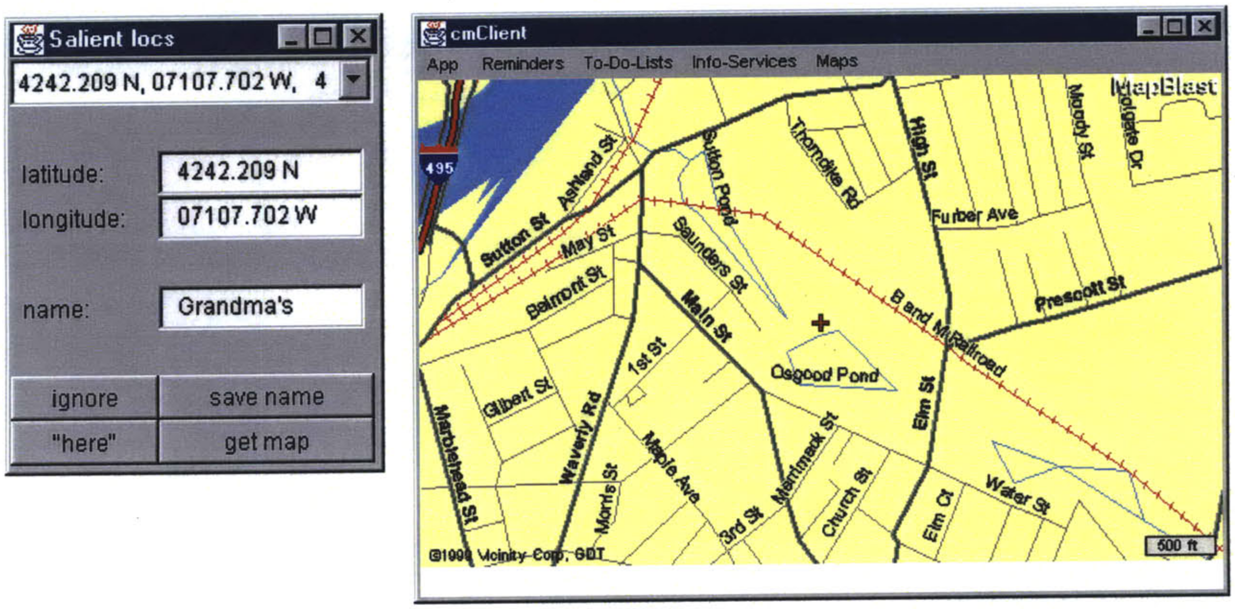


Figure 4.5: Map showing distance between location of departure (Media Lab, on Ames St) and location where identified by the GPS receiver (part way across campus, the grey area)



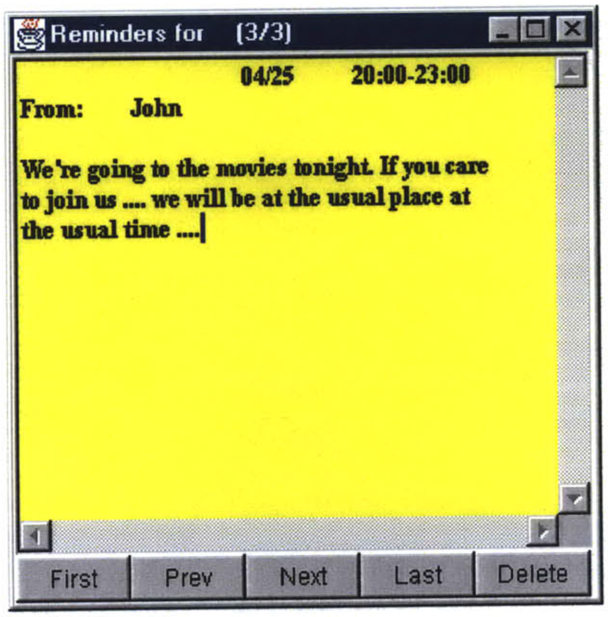
4242.209 N, 07107.702 W, 4

latitude: 4242.209 N

longitude: 07107.702 W

name: Grandma's

Figure 5.2: Visual interface components used to see a salient location on a map and tag it with a virtual location name

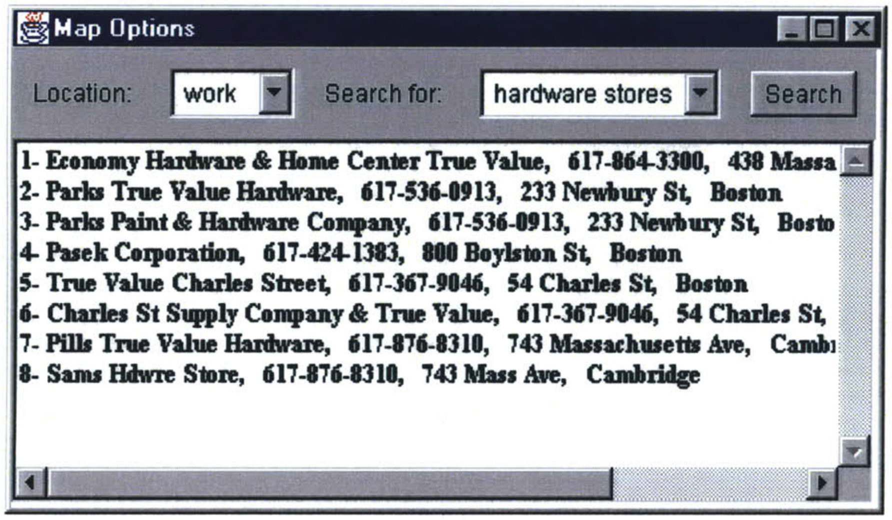


04/25 20:00-23:00

From: John

We're going to the movies tonight. If you care to join us .... we will be at the usual place at the usual time ....

Figure 5.5: Visual interface component to view and scan reminders



1- Economy Hardware & Home Center True Value, 617-864-3300, 438 Massa
2- Parks True Value Hardware, 617-536-0913, 233 Newbury St, Boston
3- Parks Paint & Hardware Company, 617-536-0913, 233 Newbury St, Bosto
4- Pasek Corporation, 617-424-1383, 800 Boylston St, Boston
5- True Value Charles Street, 617-367-9046, 54 Charles St, Boston
6- Charles St Supply Company & True Value, 617-367-9046, 54 Charles St,
7- Pills True Value Hardware, 617-876-8310, 743 Massachusetts Ave, Camb
8- Sams Hdwre Store, 617-876-8310, 743 Mass Ave, Cambridge

Figure 5.7: List of eight closest requested locales, shown on accompanying map

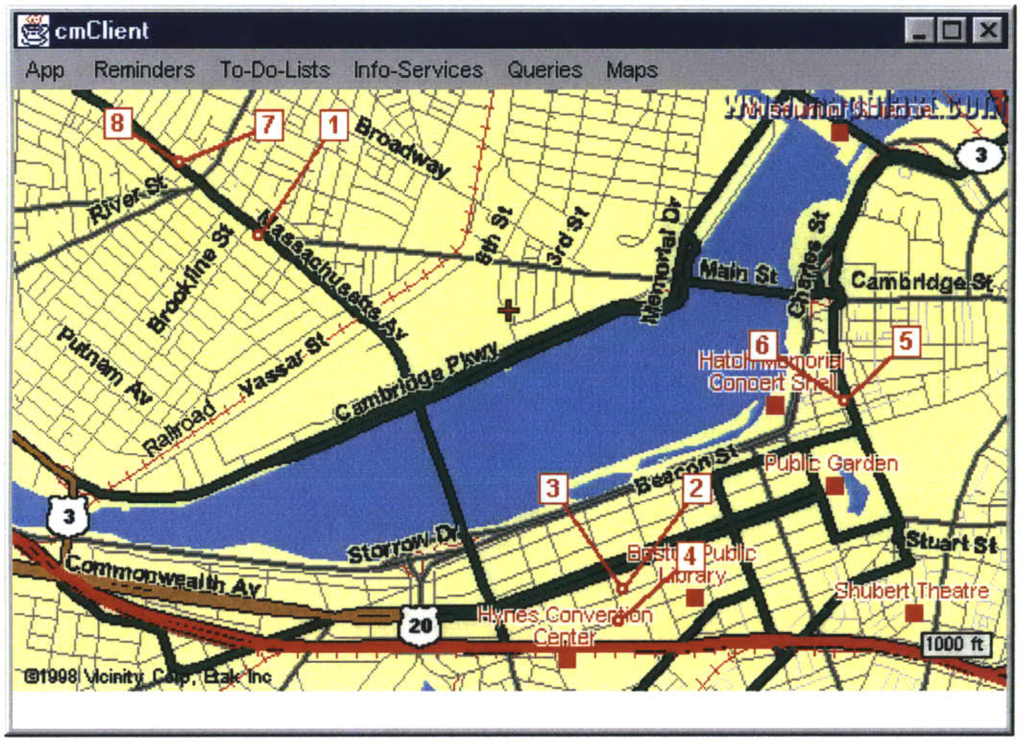


Figure 5.8: Map indicating current location and eight closest requested locales

References

[1] Maes, Pattie. "Agents that Reduce Work and Information Overload" *Communications of the ACM*, July 1994/Vol.34, No.7, pp. 31-41.

[2] Mark Weiser's Ubiquitous computing Web-page. http://www.ubiq.com/hypertext/weiser/UbiHome.html

[3] The MIT wearable computing Web-page. http://wearables.www.media.mit.edu/projects/wearables

[4] US Coast Guard Navigation Center http://www.navcen.uscg.mil

[5] Want, R., A. Hopper, V. Falcao, and J. Gibbons. "The active badge location system". *ACM Transactions on Information Systems*, 10(1): pp. 91-102, January 1992.

[6] Lamming, Mik and Mike Flynn. "Forget-me-not: Intimate computing in support of human memory". *FRIEND21: International Symposium on Next Generation Human Interface*, pp. 125-128, 1994

[7] Want R, W. Schilit, N. Adams, R. Gold, K. Peterson, D. Goldberg, J.R. Ellis, and M. Weiser. "An overview of the PARCTAB ubiquitous computing environment". IEEE *Personal Communications*, 2(6), pp. 28-43, 1995.

[8] Starner, Thad and Dana Kirsch. "The locust swarm: An environmentally-powered, networkless location and messaging system". *Proceedings of the International Symposium on Wearable Computing,* IEEE, pp. 169-170, October 1997.

[9] Poor, Rob. iRX 2.0. http://ttt.www.media.mit.edu/pia/Research/iRX2/index.html

[10] Sumi, Y., T. Etani, S. Fels, N. Simone, K. Kobayashi and K. Mase. "C-MAP: Building a Context-Aware Mobile Assistant for Exhibition Tours". Presented at the *First Kyoto Meeting on Social Interaction and Communityware*, Kyoto, Japan, June 1998.

[11] Tufty Steven. "Watcher". Bachelor's thesis, Massachusetts Institute of Technology, 1990.

[12] Marmasse, Natalia, John Holmes, Nitin Sawhney, and Chris Schmandt. "RadioSpace: physical and virtual presence". http://www.media.mit.edu/~nmarmas/radioSpace.html

[13] Rhodes, Bradley. "The Wearable Remembrance Agent: a system for augmented memory." *Proceedings of the International Symposium on Wearable Computing,* IEEE, pp. 123-129, 1997.

[14] Jebara, Tony, Bernt Schiele, Nuria Oliver and Alex Pentland. "DyPERS: A Dynamic and Personal Enhanced Reality System". *Proceedings of Intl. Conference on Vision Systems, ICVS99.* Gran Canaria, Spain. January 1999.

[15] Long, Sue, Dietmar Aust, Gregory Abowd, and Chris Atkeson. "Cyberguide: Prototyping Context-Aware Mobile Applications". *Proceedings of the Conference on Human Factors in Computing Systems, CHI'96.*

[16] Abowd, Gregory, Anind Dey, Robert Orr, and Jason Brotherton. "Context-awareness in wearable and ubiquitous computing". *GVU Technical Report* GIT-GVU-97-11.

[17] Smailagic, Asim and Richard Martin. "Metronaut: A Wearable Computer with Sensing and Global Communication Capabilities". *Proceedings of the International Symposium on Wearable Computing,* IEEE, pp. 116-122, October 1997.

[18] Pascoe, Jason and Nick Ryan. Stick-e Notes. http://www.cs.ukc.ac.uk/research/infosys/mobicomp/Fieldwork/Sticke/

[19] Dreilinger, Daniel. FolksFinder. http://www.media.mit.edu/~daniel/research.html

[20] Kreller, B., D. Carrega, J. Shankar, P. Salmon, S. Bottger, and T. Kassing. "A Mobile-Aware City Guide Application". *ACTS Mobile Communication Summit*, Rhodos, Greece, 1998.

[21] Rekimoto, J, Y. Ayatsuka and K. Hayashi. "Augmenta-able Reality: Situated Communication through Physical and Digitial Spaces". *Proceedings of the International Symposium on Wearable Computing,* IEEE, pp. 68-75, October 1998.

[22] Marx, Matthew and Chris Schmandt. "CLUES: Dynamic Personalized Message Filtering". *Proceedings of CSCW '96*, pp. 113-121, November 1996.

[23] Schmandt, Chris. Phoneshell: The Telephone as a Computer Terminal. *ACM Multimedia '93*, pp. 373-382, August 1993

[24] Sawhney, Nitin and Chris Schmandt. "Nomadic Radio: A Spatialized Audio Environment for Wearable Computing". *Proceedings of the International Symposium on Wearable Computing,* IEEE, pp. 171-172, October 1997.

[25] Marti, Stefan and Chris Schmandt. Knothole. http://www.media.mit.edu/~stefanm/pager/

[26] Marti, Stefan and Chris Schmandt. Active Messenger. http://www.media.mit.edu/speech/sig_projects.html

[27] http://www.attws.com/personal/pocketnet/

[28] http://www.umail.com/democdpd.html

[29] http://www.nokia.com/com9000/n9000.html

[30] http://www.gpss.co.uk/chase.html

[31] http://www.garmin.com/navTalk.html

[32] Allport, D., B. Antonis and P. Reynolds. "On the Division of Attention: a Disproof of the Single Channel Hypothesis". *Quarterly Journal of Experimental Psychology*, 24:225-235, 1972.

[33] AT&T Watson SDK. http://www.att.com/business/watson/index.html

[34] http://www.mapblast.com/mapblast/start.hm

[35] Ly, E. and Chris Schmandt. "Chatter: A Conversational Learning Speech Interface". *Proceedings of AAAI Spring Symposium on Intelligent Multi-Media Multi-Modal Systems,* March 1994.

[36] http://www.portico.net

[37] http://www.webley.com

[38] Manandhar Sanjay. "Activity Server: A Model for Everyday Office Activities". Master's thesis, Massachusetts Institute of Technology, 1991

[39] Luce, P. A., T. C. Feustel, and D. B. Pisoni. "Capacity Demands in Short-Term Memory for Synthetic and Natural Speech". *Human Factor*, 25(1):17-32, 1983.

7 [illegible] – 4(