

xTolk - A Context-Aware Mobile Application on the Nokia N95 8GB Smartphone

Yanhua Tu

Kongens Lyngby 2008
IMM-Master-2008

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Abstract

A three-layer system module was designed to collect, process and represent the spatial, temporal, behavioral and personal context data of a mobile phone user. A context-aware application called xTolk was prototyped in Python on the Symbian S60 platform. XTolk collects the real-time location of the mobile phone user, assesses his physical behavior, estimates the travelling time for recorded routes, identifies the present Bluetooth devices and utilizes the Nokia Mobile Web Server to host a web page displaying the context data to an authorized public. According to performance and usability tests, xTolk successfully combines different context data and exploits time-correlations; improvements for more scalable data structure and easier user interface are outlined for the future development. A small survey evaluated users' interests on existing and potential functions of xTolk.

Preface

The thesis was prepared in the Department of Informatics and Mathematical Modelling at the Technical University of Denmark, in partial fulfillment of the requirements of acquiring the M.Sc. degree in engineering.

This M.Sc. thesis is the result of work carried out in the period from 9th of May 2008 to 24th of November 2008, with the amount of 30 ECTS points workload.

Lyngby, November 2008

Tu Yanhua

Acknowledgements

It's a pleasure to thank the many people who made the thesis possible.

I would like to thank my thesis supervisor, Michael Kai Peterson and Jakob Eg. Larsen. With their enthusiasm, inspiration and efforts to explain things clearly, they helped to make writing this thesis fun for me. I appreciate all those discussions which brought good ideas. I would have been lost without the help.

I am grateful to DTU for the pleasant education and training it provided.

I would like to thank all my friends, Shen Hao, Fu Huijin, Fan Hua, Shao Lei, Hu Qi, who have contributed brilliant ideas, kind encouragements and great help during the tough time.

I owe my thanks to Anna Grethe Rischel and Aida Riberholt Rischel for the love, care and support.

I give my thanks to Antonio Franco for love, understanding, encouragement, humor and help.

Lastly, and most importantly, I wish to thank my parents, Tu Jisong and Xu Zeping, for the love through all the years. To them I dedicate this thesis.

Contents

Abstract	i
Preface	iii
Acknowledgements	v
1 Introduction	1
1.1 Context and Context-Awareness	1
1.2 Contextual Information of Mobile Phone	2
1.3 Context-Aware Mobile Applications	3
1.4 Problem Definition	4
1.5 Thesis Outline	5
2 Analysis on Key Elements of Context-aware Applications	7
2.1 Location	7
2.2 Motion	8
2.3 Environment	9
2.4 Events	9
2.5 Combination	11
2.6 Key Elements	11
2.7 summary	13
3 The Mobile Device Environment	15
3.1 The Characteristics of Mobile Device	16
3.2 The Symbian S60 Platform	18
3.3 The Requirements of Mobile Applications	20
3.4 Summary	20

4	A Mobile Context-aware Application	21
4.1	Use case	21
4.2	The System Module	25
4.3	The Application Requirements	28
4.4	The Implementation	28
4.5	Summary	28
5	Context Data Collecting	31
5.1	Spatial Context	31
5.2	Temporal Context	34
5.3	Behavioral Context	35
5.4	Personal Context	36
5.5	Summary	37
6	Context Data Processing	39
6.1	Local Processing	39
6.2	Third Party Services	45
6.3	Summary	46
7	Context Data Representation	49
7.1	The Component	49
7.2	Local Engine	50
7.3	Web Content Generator	51
7.4	Summary	56
8	Test	59
8.1	Performance Test	59
8.2	Usability Test	63
8.3	Survey of user interest	64
8.4	Summary	64
9	Conclusion and Future Work	67
9.1	Future Work	68
A	The Survey on User Interest	69
B	Python script for Web Page Generation	71
C	Starting xTalk	75
D	Option Watching You	79
E	Route Tracking Option	81
F	Arrival Estimate Option	85

G BlueMark Option

89

Introduction

The terminology, 'Context-Aware', was proposed about a decade ago, referring to the ability of computing applications to sense and take advantage of the contextual information. Nowadays, with the fast development and spreading use of smart mobile phones, increasingly more mobile applications emerge, featuring 'context-aware'.

1.1 Context and Context-Awareness

The word *context* literally means 'the interrelated conditions in which something exists or occurs' according to Merriam-Webster's Collegiate Dictionary. This loose definition does not well explain the usage of the word under computing circumstances. A more specified definition was first given by Dey and Abowd[21]:

Context is any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context

is typically the location, identity and state of people, groups and computational and physical objects.

Context-aware as a computing terminology was first introduced by Schilit[21], emphasizing that:

Context-aware software adapts according to the location of the use, the collection of nearby people, hosts, and accessible devices, as well as to the changes to such things over time. A system with these capabilities can examine the computing environment and react to changes to the environment.

Proper analyzing of context and instant response by application, namely, context-awareness, could lead to:

Easier Adaptation enabling the device to adapt itself to the changes of either the environment or the user. For example the automatic changing of mobile profile to normal, meeting or outdoor according to the external environmental conditions and the activities of the user[16].

Reduced Manual Work, the manual input work of the individual user could be simplified, reduced or even avoided. For example the automatic logging in to computers by video recognition of user's facial characteristics instead of typing username and password.

Enhanced Personalization, the higher capability of the application to learn and remember the user preferences. For example, a context-aware application could reserve different user profiles for users who vary in working environment, preferred activities and behaviors. The emphasis is on the uniqueness of the user experience.

Easier adaptation, reduced manual work and enhanced personalization help to shape the application in a dynamic and flexible way to create a unique and natural experience for the user.

1.2 Contextual Information of Mobile Phone

Contemporary mobile phones are getting richer and more robust in the functionalities by integrating various innovative technologies. And a mobile phone

differs from PC, Laptop or other computing devices as it can be used almost any time and anywhere, continuously exposed to the changing environment and user behavior. Therefore the mobile phone has comparatively more accesses to the context information. A general categorization of the contextual information available from the mobile device includes:

1. The external environmental information, which refers to location, e.g., the longitude and latitude coordinates from GPS ;Accessible networks, including current registered telecommunication network, available local area network (LAN);And neighboring entities, including both people and devices presented;
2. The internal environmental information, which refers to System information (battery status, IMEI, memory usage, software version, Hardware configuration information, etc.)
3. The user-related information, which refers to general information, including phone contacts, calendar, SMS database; Multi-media file storage, including video clips, sound tracks and pictures; and furthermore user behavior, habits, history, social/cultural aspects.

As computing power,memory and battery volume have been largely increased over the past few years, researchers have been utilizing the context information of the mobile phone to develop context-aware applications.

1.3 Context-Aware Mobile Applications

This section presents a brief review of existing context-aware mobile applications. The applications are categorized by the type of context data they utilize:

- Location. Location-based service is becoming increasingly popular lately. A large number of applications and research work have been conducted focusing on utilizing the location information. According to the capability of individual mobile phone, the location data varied from registered mobile tower, available WLAN, which approximately specify the location; to GPS (Global Positioning System)coordinates and surrounding entities with Bluetooth identity as well as RFID (Radio Frequency Identification), which presents more accurate description of location. The objective of these applications is very diverse. Cyberguide, a typical context-aware

tour guide running on mobile device, focused on providing tourist information according to the current position of the user; another spreading use based on location information is the automatic change of phone profile, the phone chooses to ring, vibrate, adjust the ring volume, or mute, depending on position of the phone.

- **Activity.** With the embedded sensors, the mobile devices are able to sense the parameters of activities such as the orientation, speed and accelerations. Mobile applications that leverage the context of activities generally fall into three fields: real time monitoring, long data logging and motion-controlled entertainments. Nokia Sports Tracker is an example of real-time monitoring application, which automatically record speed, distance, time and user's physical conditions during the training process; long data logging is able to record daily activities of elderly or patients for clinic purpose. The motion-controlled entertainment takes the gestures of the user as a part of the user input, and responses accordingly. A mobile tilt-game is one simple example.
- **Combination of third party resource.** The popularity of web services such as Flickr and Facebook thrives the new use of mobile context. The Application Programming Interface (API) of Flickr allows the mobile user to upload the photos to their Flickr albums in real time. With context-aware photo management application, the tags, which describe the content of the photos, such as location, time and events, could be generated automatically or semi-automatically by the mobile. A more recent example is the 'My Nokia Phone Status' enabled by Nokia mobile web server and Facebook API. The application enables the user to change the description of their location and status, which appears in their Facebook account by the mobile phone.

There has been intensive focus on the use of location information, although the purposes varied. The applications based on activity sensing emerged as a result of the increasing complicity and capability of the embedded sensors. The novel applications which combine the mobile phone context and the third party services are gaining focus as they increase the availability of those services.

1.4 Problem Definition

While extensive research and work have focused on one specified resource of context, the combined usage of context is largely unexplored. Besides, most applications of the context are real-time based, such as the automatic functions of

adaptation and interaction; However, the needs of user could be time-correlated, which means the current user preference could be affected or decided by the previous behavior.

This thesis investigates how a light-weighted mobile application could utilize the contextual information including the location, user calendar, phone contacts and user's physical behavior in a combined manner. The usage of time-correlated context will be also explored. With these elements in consideration, a context-aware personal mobile application, called xTolk (contEXt Translator)¹ is designed and prototyped in this thesis. xTolk is supposed to capture and record the combined context of the user in a optimized way as well as provide the platform to share the information with people who are allowed to view.

1.5 Thesis Outline

Chapter 2 introduces a selection of popular context-aware applications which are designed for different purposes. The key elements of these applications are then analyzed and compared.

Chapter 3 discusses the capabilities and limitations of mobile devices, with a focus on the platform of mobile device utilized in this thesis, and presents the general requirements for the application prototyping

Chapter 4 presents the use case and system module of the context-aware mobile application, xTolk

Chapter 5, 6, 7, each chapter demonstrate the design and implementation of one layer of the whole system

Chapter 8 presents the test process and the evaluation of the application, solutions for optimization is also investigated

Chapter 10 presents the conclusion and the future work

¹The prototype of this thesis will be referred as xTolk throughout the thesis.

Analysis on Key Elements of Context-aware Applications

In the previous chapter, definitions of context and context-awareness was presented. In this chapter applications which utilize the context information are summarized. The applications are categorized in six groups according to the type of context they utilize: location, motion, environment, events, personal status and combined usage. The variation of platforms, upon which the applications run, is not taken into consideration. The focus is on the key features and elements of these applications. The analysis of these features and elements is conducted afterwards.

2.1 Location

2.1.1 Cyberguide

Input Data: Location, Orientation

Cyberguide[10] is a prototype of context-aware tour guide application based on mobile devices such as PDAs and smart mobile phones. It comprises four

components: The map, the information base, the positioning system and the communication system. Among them, the map is the view the tourist uses to navigate around a specified area; The information base provides the on-going events of important locations; The positioning system constantly updates the location and orientation of the tourist, so that the application knows where the tourist is and provides most relevant information consequently. The communication system provides connection between the tourist's handheld device and the internet.

The long term goals of the Cyberguide project, which have not been realized, include: to know where the tourist is and what he/she is looking for; to predict and answer the questions he/she might pose; to provide interaction with other users and environment. The aim of Cyberguide, besides providing tourist information, is to enable the application to follow the user's needs, not vice versa.

2.2 Motion

2.2.1 Nike+

Input Data: Motion data, time

Nike Plus (<http://nikeplus.nike.com/nikeplus/>), as a joint effort of Nike and Apple, is the technology that enables the communication between the runners' shoes and their iPod Nanos¹. This sporting kit includes the small-size sensor for the shoes and a transmitter for iPod Nano. Thus while the user is running, the motion data including distance, pace, calories and time is sensed by the sensor embedded in the shoes, received and tracked by iPod Nano and the user has these data reported and training advice given in real time.

As the motion data are collected and recorded, the user could look into these data after running; as well as share with other users the training diary or their favorite sporting music in the web community.

¹Portable media player by Apple Company. <http://www.apple.com/dk/ipodnano/>

2.3 Environment

2.3.1 Interactive Office

Input Data: Environmental context

Interactive Office is a context-aware application for optimizing the office environment. With a series of embedded sensors in the physical environment, the application is able to monitor the context of the user. The original drive of the application is to avoid unwelcome interruptions such as telephone calls and unexpected entries to the meeting room.

The environmental context includes the sound and light level in the room, the movement via door ways, the usage of furnitures and so on. The context information is collected and evaluated by software in a rule-based system which enables pre-defined actions to be triggered. One example is the handling of telephone call during the meeting time: when there is a meeting going on in the office, the telephone call to that room will only ring once while the call connection is maintained. Thus the people in the meeting are interrupted in the minimum level and are able to decide whether answer the telephone call or not.

The core part of the Interactive office is the rule-based system which analyze the situation and reacts to specified events, namely the changing context information.

2.4 Events

2.4.1 Epocware's Handy Calendar

Input Data: Calendar Events

The application, Handy Calendar by Epocware Software Group is an enhanced time management tool for smart phones with Symbian Operating System (Symbian OS). It fully covers the functions of a normal calendar application: displaying in different view mode as monthly, weekly and daily; adding entries such as appointments, meeting, memo, notes and repeat them per user's request; reserving alarms or other kinds of reminder. The Handy Calendar enables the estimation of how busy the day could be according to the calendar entries and the synchronization to other popular calendar applications.

Compared to the functions of other calendar applications, the estimation function somehow makes Handy Calendar 'intelligent'. Aside from considering the complicity of realizing this function, the estimation is based on analyzing the existing context of the user, and makes them more meaningful. Utilized properly, there could be more out of the list of calendar events.

2.4.2 Twitter

Input Data: user events, personal contacts

Twitter (<http://twitter.com/>) is a micro-blogging service which allows the users to upload short description (140 characters) about the moments of their lives. There are multiple ways of uploading: The user is able to update the status on his/her profile page on the Twitter website; it is also possible to send the update by SMS to one of the Twitter gateways; Besides, Email is another option for updating the status, this service is provided by some third party services.

The micro-blogs are displayed on the user's profile page and also delivered to the subscriber of these information, or namely, the 'followers' of the user. To protect the privacy, one can choose to authorize certain group of people to view the updates, or block the people who are not supposed to read.

Twitter fulfills the demand for updating the user status in real-time, and provides simple, secure and fast ways of sharing the updates. On the other hand, these frequent updating relies heavily on the manual input of the user.

2.4.3 Facebook Status

Input Data: User Events

Launched in 2004, Facebook as a social networking website has attracted more than 10 million registered users up to now. In the Facebook community, one could join or create social networks by city, region, graduate school, workplace or personal interests to connect with others. There are a large number of Facebook applications available for communication and interaction, either developed originally by Facebook or third party developers with Facebook API, among them the Facebook Status is a basic and popular function.

The status is a short description of what the user is currently doing or even thinking, which appears in the profile page of the user. Other users who are

from the same group or categorized as friends, could comment on the status. Thus one can view the live status of friends along with the related comments. This function requires manually input from the user and a internet connection for uploading it, which makes the updating not really handy. But this function is still very popular, as many people are frequently connected to internet and curious to see the status of the others.

2.5 Combination

2.5.1 ContextWatcher

Input Data: User Events, Location, Motion data , Visual data, etc.

ContextWatcher is a mobile application for Symbian S60 phones. As a context-aware lifelogging application, it automatically connects to the context resource, logs the data and generate daily summary of user's locations, activities, environmental conditions and moods.

As a mobile application, ContextWatcher collects to as many context resources from the mobile device as possible, including the embedded and extended sensors, network connections, calendar events, images and some user inputs. Collecting, summarizing and sharing with other people are the key issues. With ContextWatcher, the user is able to have a automatically-generated daily context collection. However, further comprehension of these context is not taken into consideration.

2.6 Key Elements

The applications listed above are typical among the context-aware applications. The individual application utilize different types of context data and functions in different levels of context-awareness. A comparison of these applications is shown in Table2.1 and Table2.2.

Application	Manual Input	Location	Environ-ment	Motion	Events	People
Cyberguide	-	x	-	-	x	-
Nike+	-	-	-	x	-	-
InteractiveOffice	-	-	x	-	x	-
Handy Calendar	x	x	-	-	x	x
Twitter	x	-	-	-	-	-
FaceBook Status	x	-	-	-	-	-
Context Watches	x	x	x	x	x	x

Table 2.1: A comparison of context variation

2.6.1 Comparison on Context Resources

Table 2.1 shows the variation of the context resources adopted by individual applications. 'Manual Input', as shown an important context resource of most of the applications, is listed parallel to the auto-detected context such as location, motion, environment, events and people. While the manual input could be precise, personalized and rich in context, the auto-detected context are less accurate and more difficult to detect.

Among the auto-detected context information, location and events information are widely utilized by context-aware applications, either used in a combined way as Cyberguide or respectively as Context Watcher. Compared to other resources of context listed in the table, location and events information suggest more about the on-going situation and are relatively easier to retrieve.

The other context resource, environment parameters, motion and people also contributes to produce a full picture of the context. The usage of these information varies according to the objective of the applications.

2.6.2 Comparison on Automation

Table 2.2 shows the level of automation of each application. The term automation refers to three aspect:

Context Analysis The automatic analysis is performed by the application based on the collected context data. The context data becomes meaningful to the application user only after been analyzed and comprehended

by the application. The analysis could be further calculating or reconstructing of the collected context data, for example, the analysis on the data collected by Nike+ sensors provides the user readable figures and optimization suggestions as well; The analysis could also be estimation of following events or situation based on the current context data, for example, the estimation based on the calendar events of Epocware's Handy Calendar suggests more about the user's daily life than just a series of events.

Automatic Updating The automatic update of the context data reduces the manual work of the user. The user context, especially those of the hand-held device holder, could change from time to time. It is important for the application to monitor the context and follow status of the user. Therefore, automatic detection and updating is also a key issue.

Automatic Adaption This refers to the ability of the application to automatically adapt to the external environment or to the behavior of the user. The adaptation could be the decision-making process based on previous defined rules. Thus, certain response from the system are triggered according to the context data, as the Interactive office, which is able to mute the telephone call during the meeting time.

Application	Context Analysis	Automatic Update	Semi-automatic Update	Automatic Adaption
Cyberguide	x	x	-	x
Nike+	x	x		
InteractiveOffice	x	x	-	x
Handy Calendar	x	-	-	-
Twitter	-	-	-	-
FaceBook Status	-	-	-	-
Context Watches	-	x	-	-

Table 2.2: A comparison of automation variation

2.7 summary

The Cyberguide utilizes the location information of the user to provide relevant touristic information. The interactive office responds according to the physical conditions of the environment to avoid the interruptions during the office time. Nike+ uses the motion data, to collect, report, analyze and give advise to optimize the sports training of the user. Twitter and Facebook Status feature

real-time update of the user status. They provides the newest information to the people who want to know. Epocware's Handy Calendar enhances the functions of a normal calendar by providing the estimation of how busy the day could be based on the comprehension of the calendar events. ContextWatcher uses the mobile device to collect as much context information as possible and to automatically generate the daily summary.

Key elements of these applications are: a handy way of updating the status in a real-time manner, a further comprehension of the user inputs such as calendar events, a combined use of the embedded sensors of the device and proper adaption to the changing context. These elements were considered for the prototype and implementation of xTolk.

The Mobile Device Environment

The spreading use of mobile computing has changed many people's life style by providing ubiquitous information access, entertainment, organizing the daily schedule and helping people stay connected to their social network. The most popular mobile device, the mobile phone, has been adopted faster than any other technology in human history[7]. A survey conducted in Britain [23] by Vodafone on a large and representative sample of the British population showed that:

- 38% of the population say they could not live without their mobile phone
- 51% in the 15 - 24 year old range use their mobile phones to organize their social life
- 63% in the 15 -24 year old range use their mobile phone frequently
- More than twice as many people are reliant on their mobile phones than internet

The statistics show that mobilephones have the largest potential for context-aware applications in terms of number of users and time availability. This chapter focuses on the capabilities and limitations of the mobile device environment

as well as the requirements for the applications running on the mobile devices. As the prototype xTolk of this thesis is based on Nokia N95 8GB smartphone, an introduction to the software platform of this specific device is presented in the end of this chapter.

3.1 The Characteristics of Mobile Device

The two most important features of mobilephone are *mobility* and *ubiquity*.

Mobility is the nature of high portability.

Ubiquity is the potential of mobile phones to be used anywhere, anytime with any type of access. Note that this is different from the definition of ubiquitous computing introduced by Xerox PARC [24], which emphasized the invisibility of the computing environment.

Given the nature of mobile device, some physical limitations are inevitable. The user experiences of the mobile devices and services, consequently, differ from those of desktop computers. The following sections[3] discuss the limitations the mobile devices.

3.1.1 Limits of the Mobile Device

Table 3.1 shows the key limitations of the mobile device applications and services compared to those running on the desktop computers. In the past few decades, the computing power and memory of mobilephone has largely increased. Many contemporary mobilephones are capable of running multiple tasks simultaneously. However, the computing power and memory is still limited, for applications, such as those involving graphical rendering. The power consumption problem hinders the mobilephone from fully utilizing all its functionalities. As many functions of the mobilephone are battery-consuming, (e.g. initiating and running location services), these applications are turned on only when necessary to save battery power.

The high latency compared to desktop applications is yet another important constraint. It is the combined result of relatively low data transfer rate and limited bandwidth. Connection condition also varies according to the external environment of the mobilephone user. Loss of connection or weak connectivity

Limitation	Mobile Devices	Desktop Computers
Computing Power	Y	N
Memory	Y	N
Power Consumption	Y	N
Latency	Y	N
Data Transfer Rate	Y	N
Bandwidth	Y	N
Connection	Y	N
Cost of Use	Depends	Depends

Table 3.1: Key limitations of mobile services and applications (Y: limitation N: no limitation)

to the mobile network may slow down the applications running on the mobile phones.

3.1.2 Limits of Usability

The usage of mobile device is different from the typical usage of desktop computers[18]. The user of a desktop computer sits at a comfortable distance from the monitor in a very controlled environment. The lighting condition is good. The user keeps constant attention on the applications running on the screen. The usage of mobile device happens both indoors and outdoors. The environmental conditions vary. In many cases the usage of mobile device is not the primary activity. The user often focuses simultaneously on multiple tasks[11], such as driving or walking, and often has very limited attention to the operation on the mobile device.

The input possibilities are also limited for the mobile device user. The most typical and widely-adopted input method for mobile devices is the numerical keypad. As a trade-off for the market demand of smaller device size, the size of the keypad is limited. Some of the mobile devices adopted the QWERTY keyboard, however the size of the key still limits the usability of the keyboard. Another notable limitation of the mobile devices is the smaller screen. While the desktop computers provide the resolution at 800 * 600 pixels or higher level, mobile devices employ significantly smaller displays. The operations on applications which provide rich features or information, such as web browsing, may become inconvenient for mobile device users because of excessive scrolling and typing demands.

Another important characteristic of mobile device user experience is the rich-

ness of data. The mobile device stores large amount of personal information, including the calendar, phone contacts, photos, E-mails, instant messages and so on. Loss of the device causes unwanted exposure of private data.

3.2 The Symbian S60 Platform

All of the applications and services of Nokia N95 8GB run over S60 platform. The S60 platform is a software platform developed by Nokia for smart mobile phones that runs on Symbian Operating System. It is multi-tasking, meaning there could be multiple application running simultaneously, RAM permitting. Figure 3.1 shows the architecture of S60 platform[6].

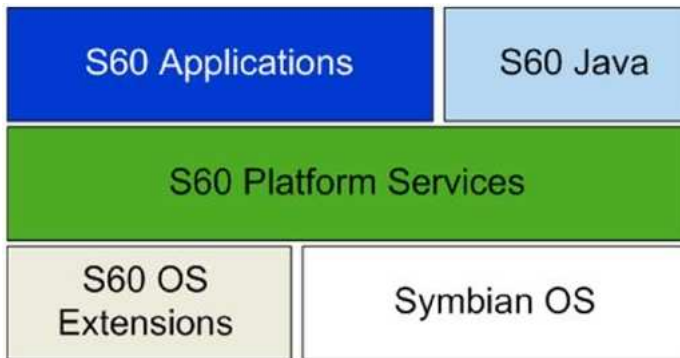


Figure 3.1: The architecture of S60 Platform

The Symbian OS(www.symbian.com) is an open, robust, multi-tasking operating system running on many mobile devices. It provides basic functionality to the S60 platform and supports the open communication standards including TCP/IP, POP3, IMP4, SMTP, SMS, MMS, Bluetooth, OBEX, WAP, Java and SyncML. Built upon Symbian OS, S60 platform uses, enhances or replaces the functionalities provided by Symbian OS.

S60 extensions provides APIs to connect applications and the device hardware such as vibrator, keypad lights, charging and power status and device profile.

S60 platform services include the main services provided by the mobile phone:

Application Framework services covers multiple components provided by Symbian OS as well as S60 platform to make common functions available

to applications. Among these functions are, for example, the application launching, event handling and internationalization and localization.

Graphic Services provides a graphics engine for the creation and drawing of graphics on the screen.

Location Services provides APIs to the developer to leverage the user's physical location in the application. With these APIs it is possible for the mobile device to scan the close-by devices, retrieve the location of the device and so on.

Web Based Services provides the HTTP utilities API and Web Services APIs. The HTTP utilities set of APIs enable the developer to build applications with downloading functionalities. Downloading can take place in the background, with no interruption to the browsing operation. It can also be suspended and resumed. The Web services APIs provide the basic services including service invocation and discovery, message security, integrity and confidentiality, XML parsing and messaging, authentication; service management and service policy support.

Multimedia services provide audio APIs, which enables the creation of application for audio recording streaming and playback; Digital Rights Management (DRM) APIs which allow the application to open, play and process DRM protected content if authorized; Exchangable Image File Format (Exif) API and Media Fetch APIs allows the creating, reading and editing of Exif metadata tags; Content Listing Framework API enable the applications to get relevant information of the media from the device.

Communication services provide networking utilities APIs which allow the application to get information of active data connections; telephony utilities APIs enables the check and validation of phone numbers; Bluetooth APIs enables the device to discover the bluetooth device and create communication in between.

S60 applications services are the features of the platform available for the developers to utilize when creating applications.

PIM applications service The Personal Information Management (PIM) provides APIs to the calendar, phonebook and notepad of the mobile phone[?]

Messaging applications service The S60 platform provides the messaging utilities APIs, which allow the user to use messaging services. The typical usage of the messaging services include Short Message Service (SMS), Multimedia Messaging Service (MMS), Email and Instant Message. Messaging services on the S60 platform are provided by the messaging framework, which uses Message Type Modules (MTMs)

Browsing and Downloading service The S60 provides APIs which allow the user to browse the web content, create plug-ins for the browser and open the browser in other applications. APIs are also provided to enable the user to create and to manage a session, in which files are downloaded over HTTP.

3.3 The Requirements of Mobile Applications

Upon the above discussion on the characteristics of mobile phone and the detailed investigation on S60 platform, a mobile application should in general:

- require as little text input as possible and provide selective options instead. Popped-out notification is another option.
- have high tolerance for interruptions such as incoming calls and SMS. It should start, stop, and resume with little effort.
- be designed for short-term and split attention of the user. This is because the user could also focus on other tasks while using the mobile phone.
- present information through multilevel or hierarchical mechanisms[5], which avoids a lot of scrolling and unnecessary reading.
- be designed for multiple and dynamic contexts, as the usability of application could change based on different context factors like the occasion, environmental conditions and network connection.
- securely handle the data flowing through the application, as the data in the mobilephone is usually sensitive and private. Unprotected exposure of these information should be avoided.
- be easy to use. Not all mobile phone users can use a computer application. Required knowledge of computing devices should be minimized.

3.4 Summary

Considering of the general requirements of mobile applications and the capabilities as well as limitations of S60 platform, a successful target application should be non-intrusive, handy, demanding only partial attention, context-dependent, secure and easy. Next chapter outlines the context-aware mobile application by describing the use cases and building the hierarchical system module afterwards.

A Mobile Context-aware Application

This chapter outlines the context-aware application, xTolk, which is based on the smart phone, Nokia N95 8GB. A multi-layered system module was designed for xTolk. Requirements for xTolk based on the system module are drawn out in the end of this chapter. The programming language for prototyping the application is PyS60 (Python for S60 platform), the advantage of the language is demonstrated.

4.1 Use case

Based on the S60 platform, the application aims to collect the available context information of the mobile phone user, analyze on them, adapt the behavior of the application accordingly and represent the context data to the user as well as other interested groups. The following use cases focus on the process that a smart mobile phone collect - analyze - represent the available context data. As shown in Figure 4.1, the system consists of three sub domains: The Data Collection, Data Processing and Data Representation domain. The Collection domain collects the available raw context data, the data processing domain

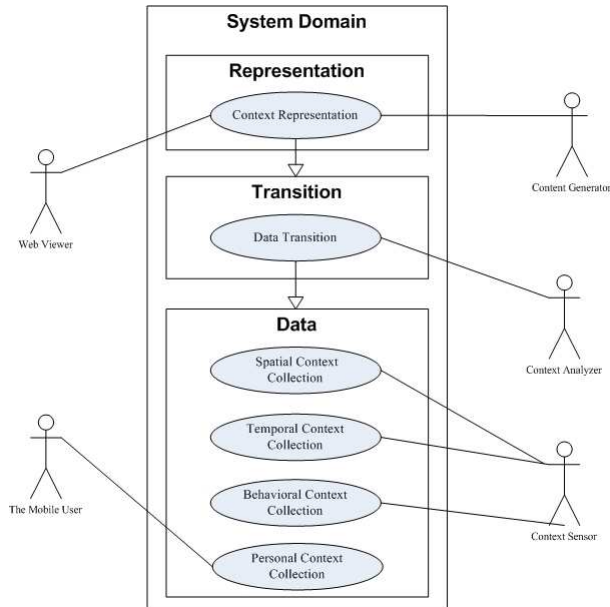


Figure 4.1: use case diagram

interprets the context, the representation domain formats the interpreted data in a readable way and represent them to different related groups.

4.1.1 Context Collection Domain

The context collection domain performs the collection of the available context data of the mobile user. As shown in Figure 4.1, the context data is collected by the context sensor. The sensors refer to both hardware sensors, which are capable of sensing light, sound, acceleration, location, and to the software sensors, which sense the current applications, the mobile phone database etc.[14]. The context data is divided into four categories.

4.1.1.1 Spatial Context Collection

The spatial context refers to the location and location-related information of the mobile user. It can be the precise geographical point,(e.g. the latitude/longitude coordinates, the street address), or a personalized reference of the location, (e.g.

the home, the office, a friend's place, or even the favorite sport club). The location-related information refers to the environmental status of the specified location. The indicators could be light and noise level, temperature, humidity, present objects etc.

Goal To collect the spatial context of the mobile user

Actors Sensors

Description The locations of the mobile user are learned and recorded by the application. Thus, when the mobile user is in a location which cannot be recognized by the mobile phone, he is requested to input the personal description of the new location. The personal description is then mapped to the absolute geographical address, which is detected by the sensor and recorded by the mobile device.

Trigger The collection of spatial data starts when the mobile user's location changes. For example, upon the detection that the user is moving from place A to place B.

4.1.1.2 Temporal Context Collection

Many of the mobile user's daily events are time-correlated. Daily events mostly follow a repetitive temporal pattern . Therefore, the time and date information are collected as the supplement to the other context data to better characterize the present user context.

Goal To collect the temporal context of the mobile user

Actors Sensors

Description The temporal context includes the time references of events that happen, the locations that the mobile user visits and the persons the mobile user encounters. It is the timeline (time, date, season) of different context data.

Trigger Temporal context collection starts when spatical, behavioral and personal context is detected.

4.1.1.3 Behavioral Context Collection

The behavioral context refers to the physical movements of the mobile user. The movements can be described by parameters such as speed, acceleration, orientation and frequency. These parameters could be collected based on the monitoring of the joints of body, such as elbow, arm, knee, ankle and wrist, which are most likely used during certain movements.

Goal To collect the behavioral context of the mobile user

Actors Sensors

Summary The behaviors, such as walking, running, staying quietly, suggest the status of the user. This movement information is collected by sensors.

Trigger When certain movements of the mobile user lasts longer than a given duration (e.g. over 2 minutes) this behavior context should be collected

4.1.1.4 Personal Context Collection

The personal context include the data inputted by the mobile user in the real-time (e.g. new photos taken, updated calendar events and new dialed calls) and the data stored in the mobile phone memory (e.g. the phone contact lists, the images and other files). Both categories are manually or semi-automatically created by the mobile user.

Goal To collect the personal context of the mobile user

Actors The mobile user

Description The mobile user inputs manually additional personal context data

Trigger It is collected when the user manually input the context.

4.1.2 Context Data Processing

Raw context data collected in the Data Collection domain are scattered and hardly give a full picture of the mobile user without further processing and interpretation. The context data processing domain combines the context data, and processes them to generate a complete picture of the mobile user's status.

Goal To process the context data and make further use of them

Actors The analyzer

Description The analyzer is a subsystem which investigate the useful information carried by the raw context. For example, being in a bus but without location change could suggest the traffic congestion. In this case, the traffic congestion is the interpretation of the spatial and temporal context.

Trigger When the data collection is updated

4.1.3 Context Representation

The context representation domain provides a meaningful representation of the context data to people who are interested.

Goal To present the interpreted context in a readable way

Actors Content generator

Description The content generator organizes and represents the interpreted context in a way that is easy to access and read. The representation could be formatted in text, tables or images, shown to the mobile user and viewed by the web browser.

Trigger Per request by the web browser for reading the current status of the mobile user

The use cases show how the application leverage the varied context information to present the status of the mobile user. The context data is passed from bottom to top through collection domain - processing domain - representation domain. The following section describes a system module (adopted in the implementation of xTalk) to perform the listed use cases. The module is constructed according to the three domains listed above.

4.2 The System Module

Figure 4.2 illustrates the system module of the application. Corresponding to the user case diagram (Fig 4.1), the system module consists of three layers: the data collection layer, the data processing layer and the data representation

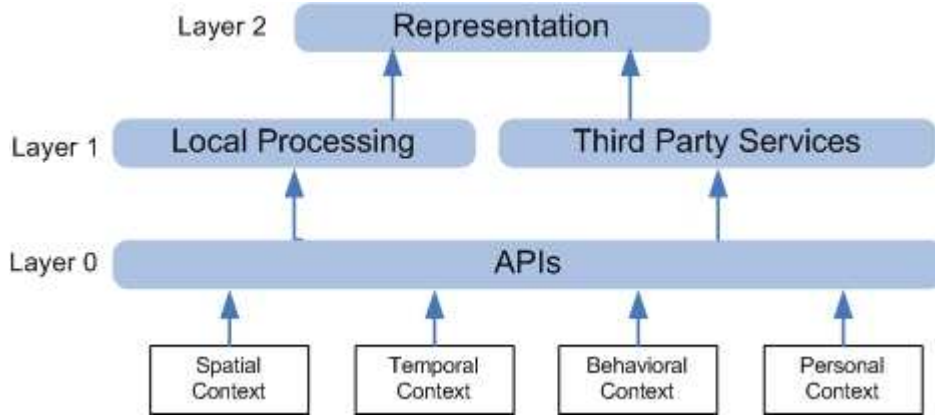


Figure 4.2: The system module

Context data group	APIs
Spatial Context	Networking Utilities Bluetooth
Temporal Context	Time
Behavioral Context	OS Extensions
Personal Context	Application Framework

Table 4.1: APIs utilized in Layer0

layer. The communication between individual layers is uni-directional and occurs only between neighbor layers: the lower layer transfers the data upwards per request by the upper layer.

4.2.1 Data Collection Layer

Layer0 is the data collection layer, the lowest layer of the module. All the context data are collected through APIs provided by the S60 platform in this layer. The S60 Platform provides APIs to different applications, services and hardware utilities. APIs are selected to provide the spatial, temporal, behavioral and personal context, as shown in Table4.1. The spatial context includes the active data connection and bluetooth information; the temporal context includes the time information; the behavioral context utilizes hardware functions; the personal context applies APIs in the Application Framework.

4.2.2 Data Processing Layer

Layer 1 is the data processing layer. It is the layer where analysis of context data is conducted. The layer includes the local processing component and the third party service component. The local processing component analyzes the context data passed from Layer 0. The analysis of the context data characterizes the mobile user's status with three parameters: the environment conditions, the ongoing events and the mobile user's behavior. While other functionalities based on the collection of corresponding context data from the lower layer could be derived, xTolk only exploits the three parameters listed above.

The third party services are not provided by the mobile device, but hosted by other devices. Examples include the Google Map service, which allows the user to upload geographical parameters to the service server and retrieve the Google Map Object. The third party services are used as a supplement to the local processing of the context data because it could save computing resources and mobile device memory. Utilizing the third party services usually requires network connection and APIs.

4.2.3 Data Representation Layer

The representation of context takes place in Layer 2. Three categories of information are passed by Layer 1 to Layer 2: the presence of people, the relevant route information and the activity of the mobile user. This information is not yet formatted for reading. Based on the processed context data there are a set of actions Layer 2 takes[14]:

- **Create New Process:** to launch a new process or an external application.
- **Notification:** to display an alert to the mobile user, with a customized user interface, vibration, sound, etc.
- **Send SMS:** to send a SMS to related groups of people with the confirmation of the mobile user.
- **Generate New Readable Content:** to translate the processed context information from Layer1 to readable format, and display it per request.

The actions of creating new process, displaying notification and sending SMS are mobile user-centered. The generation of new readable content serves other interested visitors.

4.3 The Application Requirements

Based on the discussion of the system module above, xTolk should feature:

- a interface between the application and the embedded sensors.
- the ability to adapt to the capabilities and limitations of the mobile platform.
- a semi-automatic update of context data, allowing manual input to enrich the content.
- minimal interruptions to the mobile users.
- a proper way of retrieving and reading the context data to different user groups.
- user-controlled authorization to protect the privacy.
- a secure transfer of personal data to other users

4.4 The Implementation

The xTolk is implemented with Python for S60 platform(PyS60). It is chosen for the purpose of rapid prototyping. The S60 platform provides APIs for Python to access to different resources. Furthermore, as the Nokia Mobile Web Server has ported mod_python which enables the development by Python, it is fast and handy to generate web content by Python script.

The Implementation is done in each layer of the system module. The output by Python coding of each layer is described in the following chapters. A complete copy of Python for xTolk is provided along with thesis.

4.5 Summary

Based on the use cases, the three-layered system module is built with corresponding requirements in this chapter.

In the following chapters, the layers will be discussed, focusing on design. Firstly Layer0, the data collection layer, with introduction to the available context resources provided by the mobile device; And Layer1, the data processing layer, with the design of local processing functionality and usage of third party services; Then Layer2, the data representation layer, with the design of proper representation to different groups. As the three layers are highly independent, they are discussed respectively in different chapters.

Context Data Collecting

Layer 0 collects raw spatial, temporal, behavioral and personal context data. This chapter presents the available resources of these context provided by the S60 platform. With the rapid development and maturity of location-based services (LBS)[22] of mobile devices, the resources for location context are richer than those for the three other types.

5.1 Spatial Context

The S60 platform provides two basic approaches for locating the mobile device: satellite-based system and terrestrial radio[20]. The satellite-based approach relies on the information supplied by a network of satellites, known as the Global Positioning System (GPS); the terrestrial radio approach, on the other hand, relies on the collection of information about terrestrial-radio networks, such as the mobile network base station, Wireless Local Area Network (WLAN) access point and radio resources. This section presents for resources of location information: GPS, GSM Cell ID, WLAN SSID and Bluetooth Device. The characteristics of these resources are discussed for their potential to fit with xTolk.

5.1.1 Global Positioning System

GPS ¹ is a location system based on 24 satellites orbiting the earth at altitudes of about 17,000 kilometers. The operation of the satellites are monitored by the ground stations. GPS has its root in the military-purpose project by the United States Department of Defense, but has proven to be a useful tool in non-military mapping applications during the past few years.

GPS is based on satellite ranging. Each GPS satellite broadcast a continuous stream of data including:

- Status of the transmitting satellite
- Clock/Time data
- The location of the transmitting satellite
- The orbits of the transmitting satellite

The distance between the GPS receiver and the position of three satellites are then applied to triangulation. Mathmatically three satellistes are enough to determine the location correctly. A fourth satelliste is optional in order to provide extra accuracy. Accuracy of positions determined by GPS satellites signals ranges from 50 to 100 meters. Using a technique called differential correction, the accuracy could be up to 5 meters or less[?].

As discussed above, the GPS-based location determination is accurate and cost-less. However, it has some notable drawbacks[20]:

- It may not work in all locations. GPS receiver cannot be used to determine location inside buildings, in built-up areas with a high density of tall buildings, or in some natural locations such as canyons or steep sided valleys.
- Its accuracy is decreased under, bad weather conditons.
- It does not response instantly. It could take up to a couple of minutes to locate satellite signals when first requested for a location.
- Integrated GPS hardware considerably increases battery consumption, which makes it unsuitable for an application running constantly.

¹<http://www.gps.gov/>

Although provided by N95 device, GPS was not chosen as one of the constantly updated context data resources. It is only used per user request.

5.1.2 GSM Cell ID

Global system for mobile communication (GSM) is the globally accepted standard for the second generation of mobile phones. The GSM network consists of three major systems: the switching system, the base station system (BSS), and the operation and support system (OSS)[4]. The switching system (SS) performs the call-processing and subscriber-related functions. BSS is responsible for all the radio-related functions and the OSS operates and maintains all equipments within the GSM network.

Among the three components, BSS provides useful information of the location. The base station system consists of base station controllers (BSCs) and base transceiver stations (BTS). Controlled by BSC, the BTS is the radio equipment that serves each cell of the GSM network. A cell is the area given radio coverage by one BTS, ranging from 2km to 20km, according to the density of radio communication. The operator assigns a unique numeric ID to each cell, namely the CellID. Therefore a CellID corresponds to a certain geographical area.

As a mobile phone is almost always connected to a BTS, the CellID is available at the mobile terminal and can be used as a reference for the location. Compared to GPS, CellID indication of location is not as precise, but the retrieving of CellID does not depend on weather or external environment. It consumes less battery to constantly collect the context data. Therefore the CellID was therefore adopted as a context resource.

5.1.3 Wireless LAN ID

Wireless technology has enabled two or more computers to simultaneously share resources without additional wiring. IEEE² has developed the 802.11 specification for WLAN (Wireless Local Area Network), which provides four standards based on CSMA/CA (Carrier Sense Multiple Access With Collision Avoidance) Ethernet protocol for path sharing. The coverage of WLAN is up to 50m indoor and 100m outdoor; the bandwidth is up to 54M[?], and decreased to different standards. WLAN is particularly suitable for home, office and other public use because of its coverage and bandwidth. With the spreading use of WLAN, it

²<http://www.ieee.org/>

is possible to find the location of the mobile user by indentifying the Service Set Identifier (SSID) of WLAN. A SSID is the name of a WLAN, all wireless devices within a WLAN must employ the same SSID in order to communicate with each other. It is possible for the mobile device to scan the SSID being broadcast by the WLAN. With the detected SSID, it is possible to track the location such as home, office, airport, bus and so on.

5.1.4 Bluetooth Device

Designed to replace cables for interconnecting consumer electronic devices such as printers and keyboards, the Bluetooth technology has been applied in many other applications. Bluetooth is a low-power, short-distance radio technology which operates in the unregulated Industrial, Scientific, and Medical (ISM) band at 2.4 GHZ spectrum. The coverage of Bluetooth varies from 10 meters to 100 meters. It is omni-directional and does not require line-of-sight positioning of connected devices. Both synchronous and asynchronous applications are supported, making it easy to implement on a variety of devices and for a wide range of services.

Different application based on Bluetooth connection may run over different protocol stacks[2], as shown in Figure??. For example, the business card exchange application provided by N95 runs over the following protocol stack (top - down):vCard - OBEX (OBject EXchange)- RFCOMM (Radio Frequency COMMunication)- L2CAP (Logic Link Control Adaptation Layer Protocol)-Baseband. Nevertheless, different protocol stacks share the common data link (L2CAP) and physical layer. Furthermore, each of the bluetooth device has a unique Bluetooth address, such as 00:d3:d1:24:00:e4. N95 provides the Bluetooth scanning function, which finds the surrounding active Bluetooth devices and their addresses. The Bluetooth adress of surrounding devices can be part of the context information and can be collected by the mobile phone. The device behind the Bluetooth address usually suggests the location or the occassion while the user behind the Bluetooth address suggests the events.

5.2 Temporal Context

The temporal context in this application is used as reference for other context data.The time information is available in two different ways:

- The Date-Time format, such as October, 1st of 2008, 19:30.

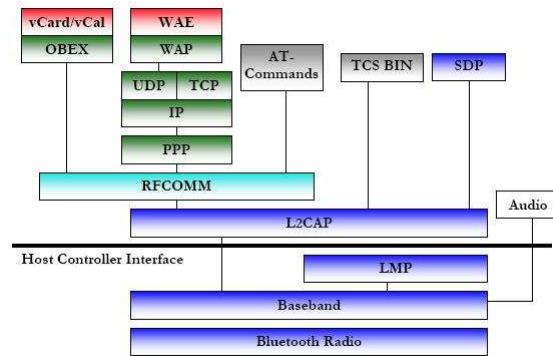


Figure 5.1: The Bluetooth Protocol Stack

- The absolute time duration, from the epoch to current time. The epoch is where the time starts for computing system, January 1st of 1970.

The date-time format is suitable for referring the occurrences of events while the absolute time can be used to calculate the duration between two events. Thus, both formats are utilized by xTalk.

5.3 Behavioral Context

5.3.1 The motion sensor

There are two embedded motion sensors available for the N95 device: the orientation sensor and the accelerometer. The orientation sensor provides data indicating how the mobilephone is placed: it could be placed on one of the six dimensions of the device. The embedded accelerometer indicates how the mobilephone is moving. It provides the acceleration with coordinates in 3 different axes: Top-bottom, left-right and front-back, as shown in Figure 5.2. The sensor data retrieved from the mobile phone are presented as 3D coordinates (x,y,z). The acceleration of the mobile device is

$$\vec{a} = \vec{x} + \vec{y} + \vec{z} \tag{5.1}$$

It is not interesting determine on which side the mobilephone is placed, as this does not enrich the user context.

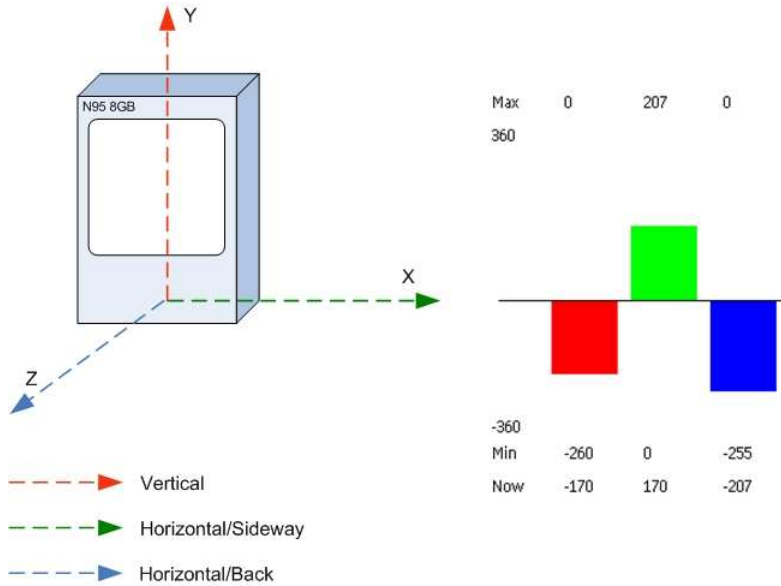


Figure 5.2: The acceleration sensor

With the accelerometer, and the changing pattern of coordinates, it is possible to represent the movement of the mobile phone, and possibly of the user. The accelerometer was therefore chosen as a resource of the behavioral context data.

5.4 Personal Context

5.4.1 Required User Input

Manual user input is requested when extra information is needed to perform the analysis in the upper layer of the system. These inputs are the personal descriptions complementing context (locations and events) or the confirmation by the user to allow certain actions the system is going to take. These requests by the application could be frequent and intrusive at the early stage of usage. As the system is supposed to learn and remember such information, requests will decrease with time.

5.4.2 Calendar Database

The original calendar database in Nokia N95 provides four kinds of calendar entry: appointment, event, anniversary and todo list, each with a unique ID in the database.

Appointment	Event	Anniversary	To-do
title	title	title	title
place	start date	start date	priority
start time	end date	end date	
end time	repeat	repeat	
start date			
end date			
repeat			

Table 5.1: calendar entry properties

All these four entries can be used as the resource of context information. The properties of calendar entries are listed in Table 5.1. Appointment entries are precisely defined in time (start and end time of the meeting), thus providing precise information to Layer 1.

5.4.3 Contacts

The contacts information is an additional resource of mobile user context. The mobile phone provides API to address book services allowing access and operation, such as searching, creating, editing and deleting, to the contacts information database. The properties of a phone contact is listed in the following. Names appearing in the calendar database can be associated with the contacts database.

5.5 Summary

The Layer 0 collects the context data and passes them to Layer 1. The following context data were selected based on their availability and usefulness for xTolk:

- The surrounding Bluetooth device MAC addresses and corresponding device names.

item city	company name	country
date	dtmf string	email address
extended address	fax number	first name
last name	job title	mobile number
note	pager number	phone number
po box	postal address	postal code
street address	url	picture
second name	voip	sip id
personal ringtone	share view	prefix
suffix	push to talk	licationid indication

- The present WLAN SSID.
- The Cell ID of the GSM network service tower to which the mobilephone user is currently registered.
- The calendar event, including the information of related people, place, start and end time.
- The acceleration parameters of the mobilephone.
- The time information in standard and absolute format.
- The contact information of the certain people per request.

Context Data Processing

The processing of context data takes place in the Layer 1 of the system. As the context data collected by Layer 0 is scattered, Layer 1 re-arranges, analyzes and formats them for the usage of Layer 2. This chapter describes how context data are processed, formatted and passed to the upper layer. The processing involves two components as shown in Figure 6.1: the local processing functionalities and the third party services. The mapping database is generated by the local processing component and utilized both by the third party service and by the upper layer. The two components are described respectively, followed by the description of the layer as an entity in Unified Modeling Language (UML).

6.1 Local Processing

The local processing refers to the processing of context data which relies only on the local resources of the mobile phone, especially the APIs provided by the S60 platform. The processing is divided into two stages: the basic processing and further processing. The basic processing re-arranges the raw context data in a proper format while the further processing interprets them.

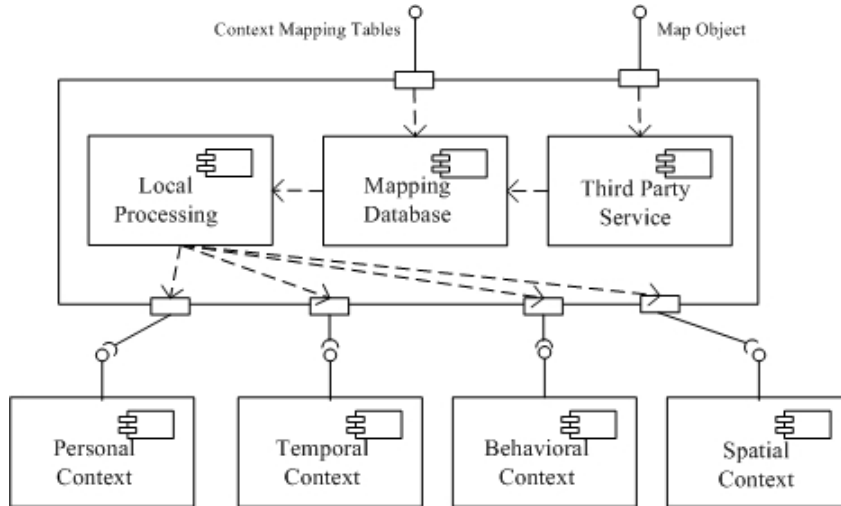


Figure 6.1: The component diagram of Layer 1

6.1.1 Basic Processing

The basic processing includes the learning of the mobilephone user's location, the surrounding people, the user's activity and the upcoming calendar event.

6.1.1.1 Location learning

The learning of location is based on the mapping between location context data and the corresponding descriptions of these locations. The location context utilized are GSM, Cell ID and the WLAN SSID. While the diameter of a GSM Cell ranges from 2 kilometers to 20 kilometers, the coverage of WLAN is relatively small, ranging from 10 to 100 meters. Therefore, the identification of the Cell ID can locate approximately the mobile phone user, and WLAN SSID can identify the location more precisely.

Upon the detection of Cell ID and WLAN SSID, the mobile user is required to input a description of the current location. The description is then saved by the application, so that the same location is recognized when visited again. Figure 6.2 shows how the mapping is structured. For the Cell ID the user is required to input the location and a personal description. For the WLAN SSID the user is required to input personal description only, because the SSID usually suggests the location of the WLAN AP (e.g. 'DTU Wireless').

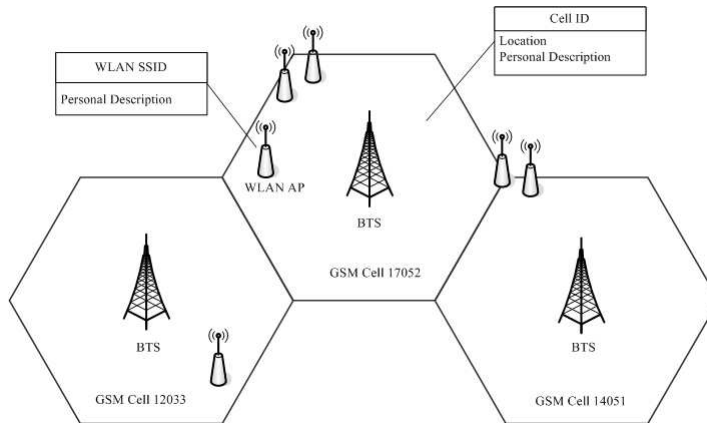


Figure 6.2: The location mapping structure

The third location context data resource, GPS, is retrieved per user’s request. For example, the mobile user uses the application to specify the longitude and latitude of the most relevant places in daily life, such as school, home or office. The GPS data is associate with the CellID of the place for further use.

Examples of these three location data are shown in Table 6.1. In the example, Cell ID and WLAN SSID are the context collected by the application, and the location and descriptions are input by the mobilephone user. The location learning process learns all the Cell ID and WLAN SSID the mobile user encounters, and save them in two tables for further use: the Cell ID - Location mappig table and the WLAN SSID - Location mapping table.The Cell ID - GPS mapping is also saved in a table, but only per request.

Cell-Loc Mapping	SSID - Loc Mapping	Cell-GPS Mapping
Cell ID = 17052 Location: <i>DTU Campus</i> Personal Description: <i>Writing my thesis here</i>	WLAN SSID = MiLab Personal Description: <i>Office of Michael and Jakob</i>	Cell ID = 17052 Personal Description: <i>My home</i>

Table 6.1: Examples of location mapping data structure

6.1.1.2 People presence learning

The learning of the presence of other people is based on mapping Bluetooth device MAC address to the device holder. The Bluetooth is installed in mobile-

phones, laptops, desktop computers, PDAs and home or office facilities such as printers, cameras and scanners. The surrounding people can be identified by the Bluetooth MAC address.

A practical problem is the time taken to scan the bluetooth devices. On average the N95 mobilephone takes up to one minute to detect the present Bluetooth devices. This time depends on the number of Bluetooth devices to scan. Besides, scanning Bluetooth devices is battery-consuming. Although the mobilephone user could encounter many Bluetooth device holders daily, most of them are unknown and irrelevant users. Therefore, unlike the mapping of CellID and WLAN ID, the Bluetooth device mapping does not take place constantly, but only per user's request. The mapping is done only when the user wants to map a certain device to a person. An example of mapping between Bluetooth MAC address and the person is:

00:12:FF:45:34:1A = (The laptop of) Matia

The mapping is then saved in Bluetooth MAC - People mapping table. Upon a subsequent detection of bluetooth MAC address, the application identifies those recorded MAC addresses, and ignore the others.

6.1.1.3 Activity learning

The learning of mobilephone user's activity is based on the analysis of the acceleration parameter passed from the embedded accelerometer of N95 mobilephone. In practice, however, the activity recognition is hardly accurate because the activity typically involves multiple parts of the body such as the neck, arms, wrists, hands, thighs, knees, crus, ankles and feet. The set of combined movements identify activity of a human. The activity is hard to recognize using a single accelerometer placed near the pelvic region[17]. The mobile phone does not necessarily reflect the movements of the user as it is often left on the table or in the bag.

The activity recognized by Layer 1 is kept to a low level. The assesses whether the user is moving around or standing. Figure 6.3 shows the periodical sampling of acceleration data.

The acceleration parameters are constant. Upon the amplitude of the sampling data, Layer 1 decides whether the mobile phone user, or rather the mobile phone is moving or standing. To avoid misunderstanding of short-term movements (e.g. the phone is accidentally touched), Layer 1 considers both the current and

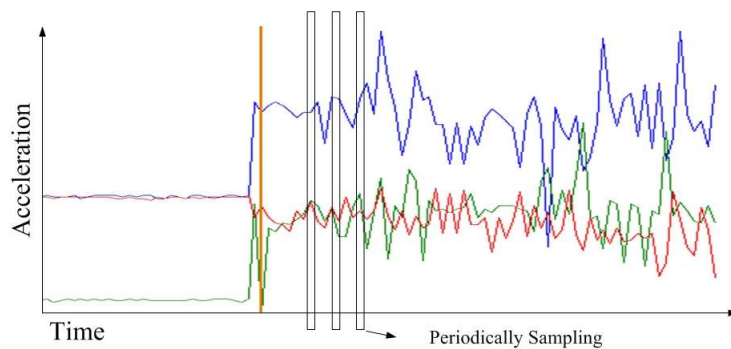


Figure 6.3: The periodically sampling of acceleration data

previous sampled data to assess the activity. Only long-term and stable status is recognized while the sporadic activities are ignored. The activity learning information is not saved in a table as for location or people information. It is however accessible for further processing.

6.1.1.4 Event and contact learning

The local processing associates the participants of calendar events to phone contact which are saved in two individual databases in S60 platform.

6.1.2 Further Processing

Based on the outputs of basic processing, the further processing provides useful and handy services to the mobile user. Two functions implemented in xTolk are: the CellID based time estimation and the Status window update.

6.1.2.1 CellID based time estimation

The CellID based time estimation is a function based on the repetition of frequent-taken routes in the daily life. This function remembers routes and estimates travelling time over a previously recorded route. Figure 6.4 demonstrates the process of remembering and estimating.

The function was tested on the route from Klampenborgvej to Taarbeck Havn.

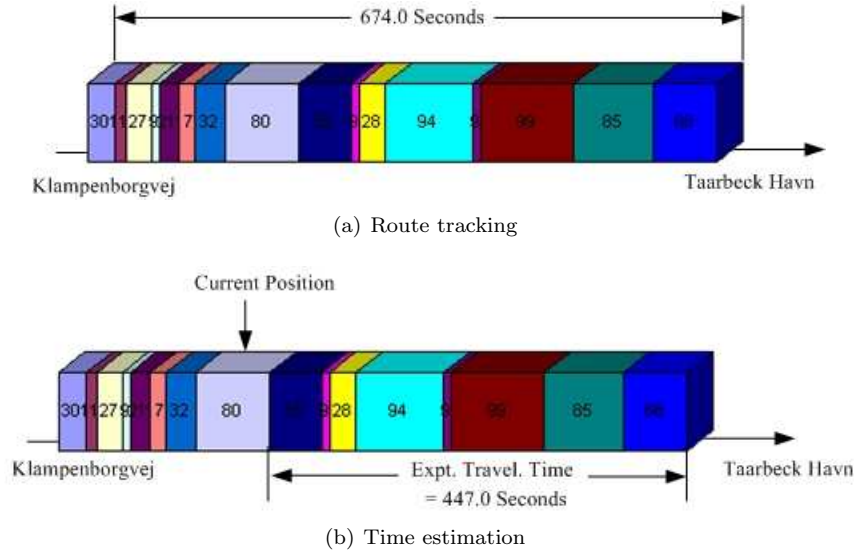


Figure 6.4: Example of route tracking and travelling time estimation

The route took 674 seconds in total. Figure 6.4(a) shows how the route was remembered. During the route, the application records sequentially every CellID detected and the duration of the CellID signal. The route is mapped to a list of CellIDs and the corresponding durations. The duration varies for different CellIDs. On arrival, the list is saved by the application. The description of the journey (i.e. the location of departure and arrival) is required to input manually by the user.

The estimation of travelling function is available for all previously recorded routes. When the application is requested to perform time estimation at certain time during the travel, it locates the CellID of the current position in the recorded list of the route (Figure ??). It then calculates the sum of the durations of the remaining CellIDs. In the example, 447 seconds is the expected travelling time to the destination. Since one CellID could appear in multiple routes, the application asks the mobile user to correctly specify the current itinerary. This is done using a dialog window. Besides, the detection of CellIDs could vary for the same travel depending on networking traffic situation. Unknown CellIDs may appear while the application estimates the time. In this case, the application looks back at the CellID appeared before the unknown one to track the current route. The previously appeared CellIDs are recorded in the status window, which is discussed in the following section.

Table 6.2: The structure of status window

Content	Format	Resource
Time	MMDDYY	The time of mobilephone
Location	CellID	Cell - Location Mapping Table
Location	WLAN	WLAN SSID - Location Mapping Table
People	Names	Bluetooth - People Mapping Table
Activity	Move/Still	Activity learning
Event	Database ID	Event and contact Learning

6.1.2.2 Status window update

The status window is a constantly updated table and records the status of last two hours. It is a sliding window along the timeline of the user's status. The status of the mobile user includes the location, the surrounding people, the user's activity, the current or upcoming events and a time stamp. These information are provided by the basic processing part. Table 6.2 shows the data structure of the status window and the resource of the content.

The status window provides a view of the current and previous status of the mobilephone user. The length of the status window could be adjusted, depending on the memory limitation of the mobile phone and on the needs of previous status records. The status window is created in order to optimize the time estimation function. The status history of the mobilephone user is available to both the mobilephone user and other interested people. It could be used for further statistic, such as finding out the most visited places and people.

6.2 Third Party Services

The third party services refer to the available APIs provided by the third party hosted web services. Examples include popular web services such as Google Map¹, Flickr², Facebook³ etc.

One of the feature built with these APIs in xTolk is Mashup, a method to integrate multiple resources available on internet. Mashups are becoming increasingly popular lately as many web sites which host applications for social networking, photo-sharing, searching and mapping provide APIs for software devel-

¹www.maps.google.com Free online map services

²www.flickr.com Popular online photo-sharing community

³www.facebook.com Social networking website.

opers to access the original data and integrate them in their own applications[8]. Mashups can provide useful local information such as events, photos and tourist information on the map. Mashup application can be client-side and server-side. The former integrates data at the client end (usually by Javascript in the browser) and provides the user with interactive user experience; the latter combines several already exposed services or data sources into a single service at the server end.

6.2.1 Google Map Service

The Google Maps API is implemented xTolk. Google Maps API is created by Google to facilitate developers integrating Google Maps into their own applications. The developers are required to request an API authentication key before retrieving the data from Google Map Service. Map data can be retrieved by sending the geographical coordinate or the address.

Sending the coordinate is a straight-forward and accurate way of locating a point on the earth. However, it requires the ability of the local application to retrieve them, usually by an internal or external GPS receiver. The basic processing of xTolk retrieves the geographic coordinate per request of the user. Thus, some of the locations are associated with the GPS coordinate and can be shown on a Google Map.

An alternative way of retrieving map data is by sending the address. The address refers to the combination of street name and number, city, zip code and country. Google Map service provides the Geocoding functionality to map these addresses to geographic coordinates and to locate the specified point on the map.

6.3 Summary

The functionalities of the basic processing are primarily generating and formatting the tables containing different context data mapping. The basic processing also assesses the physical status of the device (moving or stationary), links the participants of the calendar events to the phone contact database, performs the time estimation task and constantly updates the status window. The Google Map service generates a map object upon the geographical parameters sent by the local application. The data generated by the context data processing layer are:

- CellID - Location Mapping table
- WLAN SSID - Location Mapping table
- Bluetooth MAC address - People/Device Mapping table
- CellID - GPS coordinates Mapping table per request
- Assessment of the mobilephone user's behavior
- Estimation of mobilephone user's travelling time on recorded routes
- Constantly updated status window
- Google Map of the current position

Context Data Representation

The Layer 2 of xTolk, the context data representation layer, displays the context data processed by Layer 1 and interacts with both the mobile phone user and other people who want to read the context data . It responds upon the user's request and input, delivers feedback from the system to the user and provides a authorization function to allow external users to access the context data.

7.1 The Component

The data representation layer consists of the local engine and the web content generator (Figure 7.1). The local engine meet the needs of the mobilephone user. Its input data include the processed context data from the local processing part of Layer1 as well as the input and request of the mobilephone user. Its output data is the feedback or the notification to the mobilephone user. A graphical user interface (GUI) is built between the mobile phone user and the local engine for the user access to these functions. The web content generator gegerates readable content and hosts it for other interested users. Its input data come from Layer 1 including the Google Map Service. Its output is a web page hosted by the mobile phone. Other possible formats are discussed. This chapter describes the design of local engine and the web content generator.

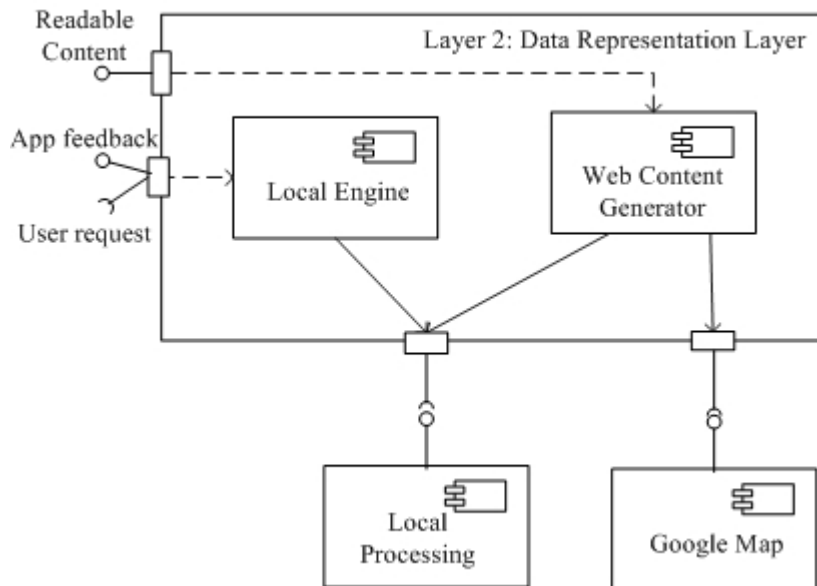


Figure 7.1: The component diagram of context data representation layer

7.2 Local Engine

7.2.1 User Interface and Functionalities

The User Interface (UI) provides a drop-down menu including a selection of different functions (7.2). In the xTolk prototype seven options are available: *Watching you*, *Shut up and Sleep*, *Start and Stop routetracking*, *Arrival Estimate*, *BlueMark* and *GPS retrieval*.

Watching You switches xTolk to monitor the context of the mobile phone user. The following context data are searched: the location of the mobile phone user, the available network, the surrounding Bluetooth devices, the motion and the calendar events. In the prototype, the interaction takes place when a new GSM Cell ID or WLAN SSID is detected. In this case, the user is required to input the relevant information in a pop up window. Upon the detection of Bluetooth devices, xTolk only displays those recorded before. All context information is automatically collected except when new locations and local networks are detected. However, this interruption is expected by the user who selects the *Watching You* option.

Shutup and Sleep stops all the running functionalities of xTolk.

Start and Stop Route Tracking triggers the function to learn the route the mobile phone user is currently taking. xTolk detects the GSM Cell ID the user passes by and records the time duration for each of the cells. No interaction takes place during the tracking process, even if the detected Cell ID is unknown. This is because the duration of a Cell ID could be as short as a few seconds, but the specification of the location by the user takes up to half a minute. This could cause a mismatch between the Cell ID and the user's description. Therefore, learning the location is only included by *Watching You* option. When the mobile user selects *Stop Route Tracking*, he is asked to input the location for the starting and end points of the route.

Arrival Estimate provides the estimation of arrival time for a recorded route. xTolk searches in the route database the recorded routes containing the current GSM Cell ID, asks the user to confirm the selected route and finally calculates the time remaining to complete the route. The xTolk checks the calendar database, if there is any appointment scheduled before the estimated arrival time. In this case xTolk alerts the mobile phone user and proposes to send a reminder SMS to inform contact people who appear in the calendar event about the estimated arrival time.

BlueMark triggers the scanning of surrounding Bluetooth devices. It detects all the present MAC addresses and corresponding device names, and asks the mobile phone user to name the relevant devices so that the system can recognize them upon later detection.

GPS Retrieval enables xTolk to retrieve the geographical coordinates of the device. GPS coordinates are retrieved only per user request, and only for those locations that the user wants to display on a map.

7.3 Web Content Generator

7.3.1 The content

The content provided to allowed users describes the context of the mobilephone user. In xTolk, the content is formatted in combination of texts and graphics. The textual content appears as in the following example:

- I'm now at *DTU*, which I think is *full of students*

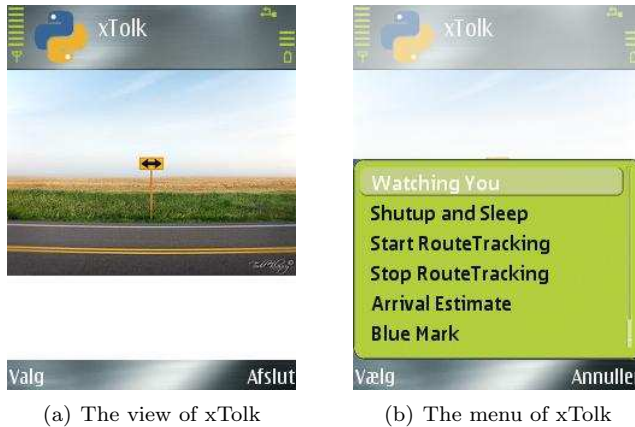


Figure 7.2: The view and menu of xTolk

- I am in *MiLab*
- I have been here for *2 hours*
- I am surrounded by *Maciej, Morten*
- I am *staying quietly*
- According to the schedule of my calendar, I am going to meet *Michael* at *15:00 Mon, Nov 17,2008* at *Milab*

The content in italic form is generated in real time based on the data provided by the processing layer. Table 7.1 shows the resources of displayed content.

Displayed Content	Context Resource
<i>DTU</i> <i>full of students</i>	GSM CellID Mapping Table
<i>Milab</i>	WLAN SSID Mapping Table
<i>Maciej Morten</i>	Bluetooth MAC address Mapping Table
<i>staying quietly</i>	Learn Activity Function
<i>meet Michael</i> <i>15:00 Mon, Nov 17,2008</i> <i>at Milab</i>	Calendar database

Table 7.1: The corresponding context resources provided by Layer 1

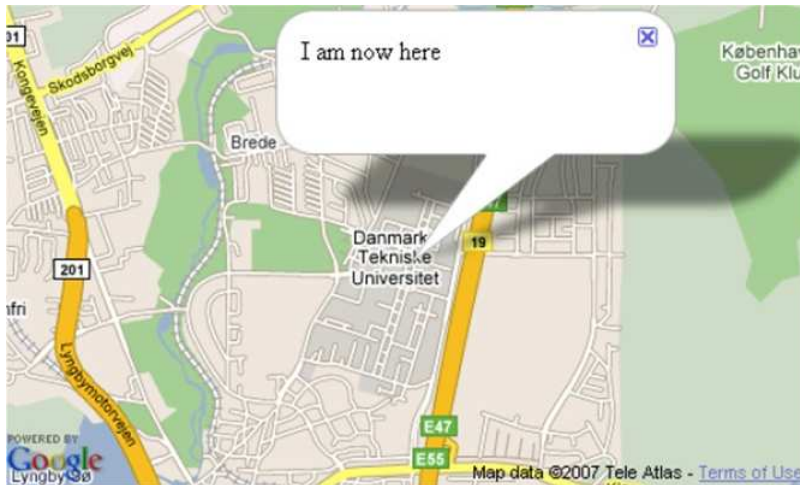


Figure 7.3: Google Map service embedded in the web content

A Google Map of the mobile phone user's position (Figure 7.3) is optional. When the web content generator finds the geographical coordinates or address associated to the current location, it displays the map retrieved from the Google Map server; if this information is not found, no action is taken.

The content generated includes detailed information of the mobile phone user's status. Therefore measures to prevent unintended exposure of this information are essential [9].

7.3.2 The Viewing approach

The viewing approach refers to the method utilized by the end users to view the context data of the mobile phone user. Desktop computers and laptops are the most widely adopted devices to connect to internet. The user could view the web content with a web browser such as IE¹, Mozilla Firefox², Opera³ etc. Standard web pages meet the needs of reading the data.

Another important category of potential viewers are the users of handheld devices to connect to internet, such as PDAs and mobile phones. Since these devices have small screens, the web pages for desktop computer and laptops

¹www.microsoft.com/windows/products/winfamily/ie Free web browser by Microsoft

²www.mozilla.org An open-source XML/HTML browser.

³www.opera.com Web browser and internet suit by Opera company

are no longer the best option. The Web Widget is a good alternative for these devices because it avoids extensive scrolling.

Only standard web page is implemented in prototype of xTalk.

7.3.3 The Host

The most widely-adopted way of publishing personal data is to upload the relevant data on a personal web page. A web server provided by a third party hosts the content and allow a authorized public to browse. However,uploading, hosting, delivering and retrieving the data of xTalk should be secured because many of them are too detailed and/or private; Secondly, as the context information of the mobilephone user is constantly updated, the data stored by the host should also be continuously updated. This differentiate a web page providing context data from a normal personal web page. Furthermore, frequent uploading requires computing power, memory and battery power.

The Nokia Mobile Web Server (MWS) is a better alternative than the traditional web server used for hosting personal web pages. In the following, an introduction to MWS is presented, followed by a discussion on how MWS addresses problems about the security, frequent updating and power consumption.

7.3.3.1 Nokia Mobile Web Server

The Nokia MWS is not the first web server implemented on the mobile devices. For instance, Nicoloudis and Prastitha presented an architecture for a web server running on Microsoft Pocket PC[19]; McAleey provided SmallServ, a simple HTTP web server for Symbian OS[15]. The MWS ports Apache httpd to the Symbian platform and provides multiple functionalities based on the server[1]. visitors can reach the mobilephone by the URL of the registered mobsite (https://mobilephoneusername.mymobilesite.net). MWS consists of :

S60 port of the Apache HTTP server Apache HTTP server is one of the most popular web servers deployed with large websites. It is ported to the platform of the mobilephone because of its robustness, its richness of features and its modular structure. Unneeded features can be excluded to save the limited memory. Apache is implemented upon Apache Portable Runtime (APR), a software library providing predictable and consistent interface to an underlying platform-specific implementation[13]. The Apache modules are written in C or C++, which enables the access to

all functionalities of Symbian and S60. Mod_Python is ported to integrate Python with Apache to accelerate development and prototyping.

Connector As the ported Apache web server only handles the issues related to content services, a connector creates and manages the connectivity between the MWS and the gateway. A gateway is a service running on a computer which connects the mobilephone and the browser. MWS is usually connected to the gateway through the cellular network or WLAN. In this case, the visitors can reach the server by the URL provided(<https://mobilephoneUserName.mymobilesite.net>). This URL is mapped to the IP address of the gateway and the gateway is then connected to the specified mobilephone according to the *mobilephoneUserName* appeared in the URL. Thus visitors perceive a direct connection to the mobile phone. The content of web server is also accessible locally by the mobilephone user without external connections. In this case the IP address of the web server is the standard 127.0.0.1.

User Interface MWS provides one user interface for the functionalities run on the S60 platform of the mobile phone and one appearing on the web page.

Web Content The web content includes some functionalities providing access to the native applications of the mobilephone such as SMS and Calendar information. It is possible for the mobilephone user to create his/her own applications running on MWS.

Considering the limited processing power and memory of mobile phone, data flowing and access is protected by the Gateway. The firewall on the gateway handles the Distributed Denial of Service (DDoS) [12] and other attacks before the connection reaches the mobile phone. The gateway provides secure connectivity to both the browser and the MWS: the connection between gateway and the web browser is in Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) while the connection between the gateway and the MWS is in Secure Socket Layer (SSL). The gateway filters Malware to protect MWS. In addition, the mobile user could specify the accessible web content and provide usernames and passwords to visitors.

MWS well meets the requirement of frequent updating of web content. As the MWS and the updated content are in the same location (the mobile phone) , no continuous remote transferring and uploading is needed.

The battery consumption is reduced by utilizing the gateway to handle security and access control functionalities. An offline web page is stored in the gateway, and activated when the mobile phone is switched off. Another way of saving the battery is to define a threshold of the battery power for running the MWS. For example, the MWS could be configured so that it is accessible only when the battery power is more than 15% or when the charger is connected.

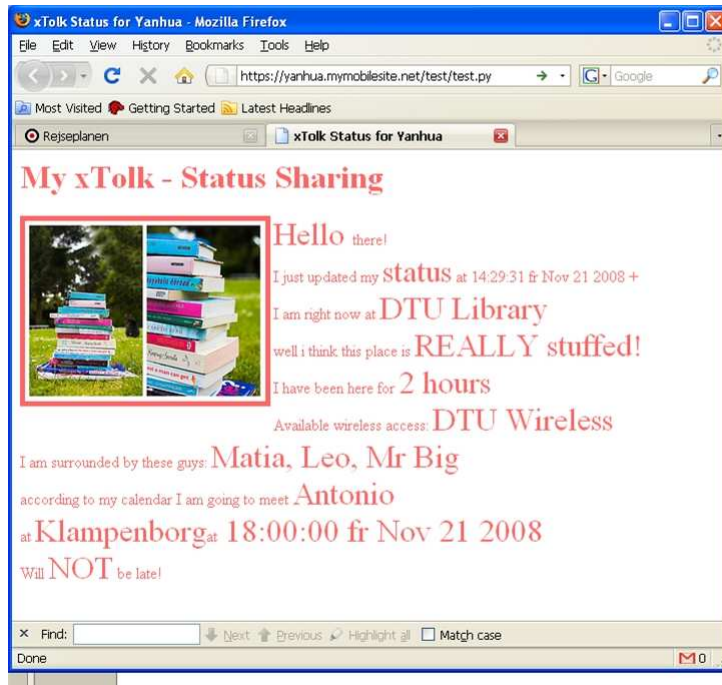


Figure 7.4: The personal webpage for context data sharing

7.3.3.2 The personal webpage

The personal web page which hosted by MWS provides the context information of the mobile user to the authorized visitors. A sample web page the url <https://yanhua.mymobilesite.net/test/test.py>. is shown in Figure 7.4. The web page is created by python script per request of the visitor (source code available in Appendix B). The web page shows the real-time status of the mobile phone user.

7.4 Summary

The data representation layer provides a local user interface to the mobile phone user and generates the personal web page which displays the mobile user's context to the authorized visitor. To provide a secure connectivity between the mobile user and the visitor, support the constant update of context data and

reduce the batter power consumption, Nokia Mobile Web Server is chosen to host the web content.

This chapter presents the outcome of preliminary testing of the prototype. xTolk was first tested internally to test the performance; then, feedback was collected based on one usability test by a potential user briefly introduced to it and on a small survey on 11 students with different knowledge background and experience. The survey evaluated the interest of existing and potential functions of xTolk.

8.1 Performance Test

The current prototype of xTolk includes the functionalities outlined in Chapter 4. Less focus is put on the non functional area of the application. The Performance Test lasted one week to expose the prototype to possible problems of battery consumption or database performance.

8.1.1 Battery Consumption

The device Nokia N95 8GB has a Nokia Battery (BL-6F) 1200mAh. According to the N95 official datasheet[35] the standby time of the phone is up to 280 hours

when the phone is registered either to 3G or GSM network. The Nokia N95 8GB smart phone tested has been used for approximately 12 months, thus with notably shorter standby time. To evaluate the power consumption, the phone was used for one week (week A) without switching the application on and for one week (week B) with the application running with full functionalities, providing exactly the same external environmental conditions and user behaviors. The external conditions and usage pattern are shown in Table 8.1.

Table 8.1: The external conditions and usage pattern of week A and week B

Phone Status	Mobile phone is always on through the week Registered to GSM network only
Daily Usage (Average Statistics)	20 minutes calling time 10 SMS sending and 10 SMS receiving 20 minutes music playing 3 Photos taken without Flash light 5 minutes gaming No internet browsing or messaging

With the same external conditions and basic usage pattern, in week B the Nokia Mobile Server is switched on from 13 PM to 21 PM and the following processes of xTolk run every day:

- Recording 3 routes, which last from 1 minute to 15 minutes.
- Detecting and recording the surrounding Bluetooth devices per requests for five times in different scenarios.
- Five estimations of travelling time
- Continuous scanning and updating of the user context

While the standby time in week A is up to 3.5 days, it is less than one day in week B. The battery consumption of week B is remarkably higher because of:

- Constant scanning the GSM Cell ID and wireless networks.
- Scanning of Bluetooth devices.
- Calculation of travelling time.
- Constant running of Nokia Mobile Web Server. The power consumption of the Nokia Mobile Server increases with the number of visitors.

To reduce the power-consumption, Continuous update of full picture of user's context could be replaced by updates based on change of location. Only the GSM CellID would be constantly scanned, while the scanning of wireless LAN and Bluetooth devices are triggered by a change of location. Besides, the mobile user should be able to switch off certain functions (e.g. switch off Bluetooth scanning). Finally, limited access to the Nokia Mobile Web Server also reduces the consumption of power.

8.1.2 Database Performance

The five mapping tables produced by the processing layer are the core part of xTolk: the Cell ID - Location mapping table, the WLAN mapping table, the Blue Mark mapping table, the Status Window and the Route Tracking table. The response time of different functions was investigated during week B.

Web Page Response The generation of the HTML personal context data web page by PyS60 takes approximately eight seconds. The production of the HTML web page requires operations on all the mapping tables except the Route Tracking table. This result is valid for a single visitor request.

Route Checking Response The time estimation function of xTolk requires user to confirm the current taken route. During week B 20 routes were recorded by the application and many of them cover identical locations. On average the user had to deny once before the application found the right route from the Route Tracking table. It took less than one second to display the next possible route.

At the end of week B every table was extended with computer-generated data. For example, the Cell ID - Location mapping table had 50 GSM Cell ID - Location matches; 200 extra matches were then input to the table. The same is applied to the other tables except the Status Window, whose length is fixed. The response time of the applications were:

Web Page Response seven to twelve seconds with the same network connection.

Route Checking Response less than one second to find the next possible route; however the response was not satisfactory as it took up to 5 denials (about 30 seconds) before the correct route is found.

The performance was acceptable except for the Route Tracking function, whose response time increase as the corresponding table scales. The data in the route tracking table is structured as:

```
Route_Tracking_Match = {
  'Route_0': [{cellid_0:duration_0}, ... {cellid_n:duration_n}]
  'Route_1': [{cellid_0:duration_0}, ... {cellid_n:duration_n}]
  ...
  'Route_m': [{cellid_0:duration_0}, ... {cellid_n:duration_n}]
}
```

The routes are recorded as matches between the route name and the sequence of Cell ID durations. When the application tries to specify a route by a given Cell ID, it searches the Cell ID in all recorded routes until the user confirms. This becomes time consuming as more routes are recorded. An optimization of the data structure could be:

```
Route_Tracking_Match = {CellID_0, CellID_1, ..., CellID_n}
```

While CellID_n could be built as an object:

CellID: 17251		
Location: DTU		
Description: Campus		
Next CellID	Appeared Times	Route Name
17250	8	A to B
13293	5	A to C
14033	3	M to N
12011	2	M to K
18743	1	X to A

← Most likely

← Least likely

Figure 8.1: The Optimized data structure of Route Tracking Table

In this structure, each CellID is labeled with the number of confirmed match to specified routes. In the example of Figure 8.1, the CellID 17251 was confirmed for route A to B eight out of nineteen times. As this is the most frequently taken route, it should be the first match attempt.

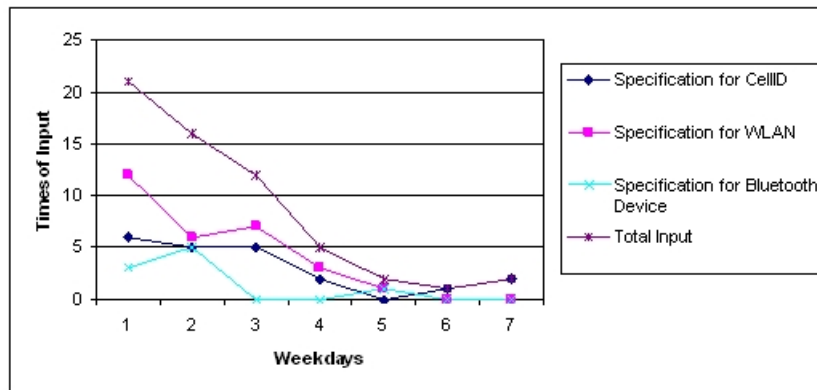


Figure 8.2: The daily input of xTalk

8.1.3 Required Input

During week B, xTalk was used under normal daily context:

Monday to Friday Working at school (DTU Library) in the day and traveling back home (Taarbeck Havn, Klampenborg).

Saturday to Sunday Staying home

Figure 8.2 shows the number of daily input by the user. The required input of the user decreased, meeting the requirements set for the design of the application.

8.2 Usability Test

The usability test evaluates the application user interface. The user¹ was given a brief introduction to xTalk and an overview of functionalities. The user could not understand the menu without guidance. Specific comments are hereby listed:

1. The *Watching You* is acceptable, although the user does not always want to make a comment on the location. Besides, prior knowledge of GSM Cell dimension is necessary to describe the detected location.

¹26-year-old male without prior knowledge of context-aware applications

2. The user found annoying to specify multiple WLAN detected.
3. The user was interested in the *Route Tracking* function.
4. The user was curious about the *Time Estimate* function, and found it useful.
5. The user is aware of the *Blue Mark* function, but would not use it. He showed concern about privacy violation by scanning Bluetooth Devices.

8.3 Survey of user interest

A survey on user interest was conducted among 11 DTU students². The interviewed had to rank from one to six a list of context-sharing functions according to their interest. Implemented and potential functionalities of xTolk were included in the list.

Among twelve options(listed in Appendix A), the two least-ranked functions were 'Displaying how long have you been in your current place' and 'Displaying the people you are with now', both ranked 1.54 in average; the two highest-ranked were 'Displaying the closest bar(or other entertainment) to you' and 'Displaying available wireless LAN', ranked 4.36 and 3.36. 'The physical behavior such as running, walking, biking' was judged fairly interesting, ranked 2.5.

People expressed concern about security, privacy and several were worried about improper commercial use.

The survey suggests that new features (e.g., 'closest entertainment to current location') could be implemented to xTolk to better satisfy the user. The detection of physical behavior, although technology-demanding, is welcomed by the users. Users are unwilling to share some personal information, so there should be options in xTolk allowing user to limit the content of the personal web page.

8.4 Summary

xTolk functions successfully combines the available context data of the mobile phone. It performs well with small scale of context data; the performance is acceptable with large scale of context data except for the route-checking function,

²5 Female and 6 Male, age ranging from 18 - 26.

which requires improved data structure. The usability test showed that the user interface can be improved and the survey identified most interesting features.

Conclusion and Future Work

Available context data of a mobile user were collected from the mobile phone, processed and represented by the prototype application xTolk. A three-layer hierarchal system was built to collect context data of a mobile phone user. Data of location, events, activities and present devices were collected in real-time. The temporal sequence of these context data was used for a travel time estimation function. Local processing combined with third party service turned raw context data to well-formatted context information of the mobilephone user. Context data are available to the user and can be shared to authorized visitors via the Nokia Mobile Web Server.

The proposed system module successfully utilized the context data collected by the mobile phone, combined usage of location, events and contacts of the mobile phone user and provided handy services. The personal mobile web page hosts context data in a secure way. Future improvements of the prototype xTolk may include optimizing the database of the context data, improving the user-interface and developing novel functionalities such as the indication of closest services or entertainment

9.1 Future Work

Based on the performance and usability test and on the survey on user interest, the following improvements are outlined for the future developmet of xTolk:

- *Data Structure* - As the amount of context data increases in the long term, a robust and scalable database for the different context data is important. The response time of xTolk and the personal web page hosted by Nokia Mobile Web Server could be shortened (e.g. by utilizing improved data structure).
- *User-friendly Interface*- An improved user interface should enable the user to easily understand and select the functionalities and to switch off unwanted functions.
- *Enhanced Features* - More detailed description of mobile phone user's physical behavior and on the near on-going events are potential interestig features to be added considering different target user groups according to age, user experience and cultural interests.

APPENDIX A

The Survey on User Interest

Your background: Please make a cross over the corresponding option(s).

Age	Under 18	18 - 25	25 - 40			
Sex	F	M				
You use mobile phone purposes of	SMS More	Calling	Music	Gaming	Navigation	
Do you download mobile applications from internet	Y	N				

if your phone could provide the following info to your friends (who are allowed by you to know these info) how much could these options interest you? Please rank from 1 - 6. While 1 means you don't care or you are not willing to expose it, 6 means you would very much like to let your friends know.

	1	2	3	4	5	6
Your current location						
Your comments of the location	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Where you have been	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Where you are going	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
How long you have been here	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The people you were with 1 hr ago	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The people you are with now	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The people you are going to meet soon	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Your physical behavior (fx.running.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Your speed for the behavior	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The available wireless network	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The closest bar (or other entertainment) to you	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Other wish, please comment:

APPENDIX B

Python script for Web Page Generation

```
Cell_Map_Pfile = u'e:\\Images\\cellMap.pkl'

Status_window_Pfile = u'e:\\Images\\statusWindow.pkl'

import pickle

def handler(req):
    from mod_python import apache
    req.content_type = 'text/html'
    nowSta = getStatus()
    req.write("<html>")
    req.write("<title>xTolk Status for Yanhua</title>")
    req.write("<H1 > My xTolk</H1>")
    req.write("Hello there!<BR>
    I just updated my status at "
    + nowSta[2] + "<BR>" )
    req.write("I am right now at
    + nowSta[0] )
    req.write(" well i think this place is "
    + str(nowSta[1]) )
    req.write("I have been here " + nowSta[7] )
```

```

req.write("Available wireless access: "
+ nowSta[6] + "<BR>")
req.write("I am surrounded by these guys: "
+ nowSta[3]+"<BR>" )
#req.write("It seems she is " + str(nowSta[4]) +"<BR>" )
if nowSta[5] == 0:
    eventDescript = 'do not have any appointment in 24 hours.'
else:
    event = nowSta[5]
    eventDescript = ' am going to meet '
    + str(event[0]) + ' at '
    + str(event[1]) + ' at<' + str(event[2])
req.write("according to my calendar I " + eventDescript)
req.write("</body></html>")
return apache.OK

def getLoc(loc):
    loc = str(loc)
    CellMapFile = open(Cell_Map_Pfile,'rb')
    cellMap = pickle.load(CellMapFile)
    CellMapFile.close()

    if cellMap.has_key(loc) :
        locdesp = cellMap[loc]
    else:
        locdesp = ['somewhere on this planet','nothing special']

    return locdesp

def getPeople(people):
    thePeople = ''
    if len(people) == 0 :
        thePeople == 'None'
    else:
        for i in people:
            thePeople = thePeople + str(i) + ', '
    return thePeople

def getDuration(loc):
    StatusWindowFile = open(Status_window_Pfile,'rb')
    statusWindow = pickle.load(StatusWindowFile)
    StatusWindowFile.close()

```

```
statusWindow.reverse()

duration = ' for more than 2 hours'

for i in statusWindow:
    location = i.keys()
    location = location[0]
    if location != loc:
        status = i.values()
        status = status[0]
        dura = status[0]
        duration = ' since ' + str(dura)
        break

return str(duration)

def getStatus():
    StatusWindowFile = open(Status_window_Pfile,'rb')
    statusWindow = pickle.load(StatusWindowFile)
    StatusWindowFile.close()

    print statusWindow

    statusWindow.reverse()

    print 'reversed'
    print statusWindow

    nowStatus = statusWindow[0]

    print nowStatus

    loc = nowStatus.keys()          # return a list
    loc = loc[0]
    nowSta = nowStatus.values()     # return a list
    nowSta = nowSta[0]

    locdesp = getLoc(loc)
    statusLoc = locdesp[0]
    statusDesp = locdesp[1]

    statusTime = nowSta[0]          # a string
    statusPeople = getPeople(nowSta[1]) # a string
```

```
statusAct = nowSta[2]           # a string
statusEvent = nowSta[3]        # a list
statusWlan = nowSta[4]
statusDuration = getDuration(loc)

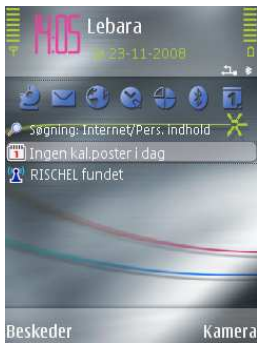
statusDisp = [statusLoc,
              statusDesp,
              statusTime,
              statusPeople,
              statusAct,
              statusEvent,
              statusWlan,
              statusDuration]

return statusDisp
```

APPENDIX C

Starting xTolk

The following describes how the xTolk application is started.



(a) N95 8GB mobile phone main screen



(b) Select the program option



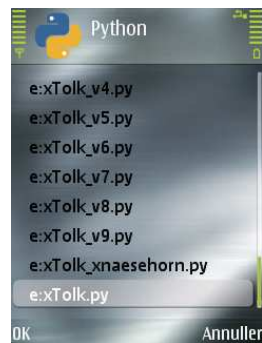
(c) Select Python



(d) The Python interface



(e) Select 'Run Script' option



(f) Choose xTolk.py



Vælg Afslut

(g) The user interface of xTolk



Vælg Annuller

(h) The menu of xTolk

APPENDIX D

Option Watching You

Functions provided by *Watching You* option



(i) The note popped out when function selected



(j) User is required to specify the WLAN location



(k) User is required to specify the detected GSM CellID

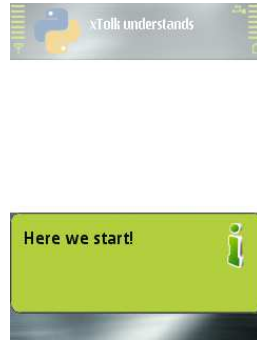
APPENDIX E

Route Tracking Option

Functions provided by *Start/Stop Route Tracking* option



(l) Select the Start Route Tracking option



(m) A note popped out



(n) Select the Stop Route Tracking option



(o) User is required to input the start and stop location

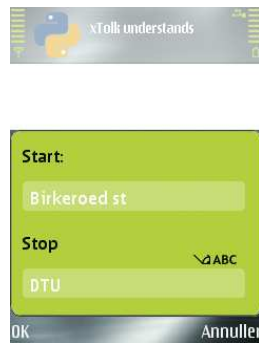
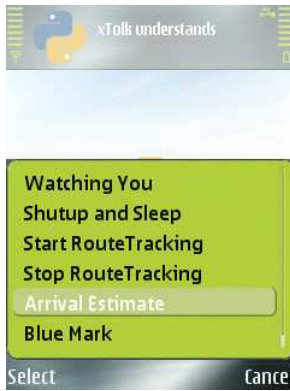


Figure E.1: The start and stop locations are specified

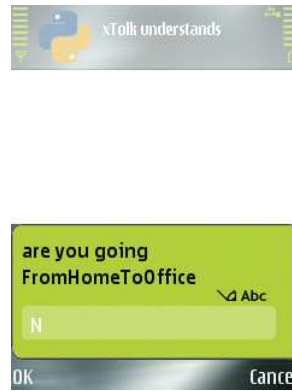
APPENDIX F

Arrival Estimate Option

Functions provided by *Arrival Estimate* option



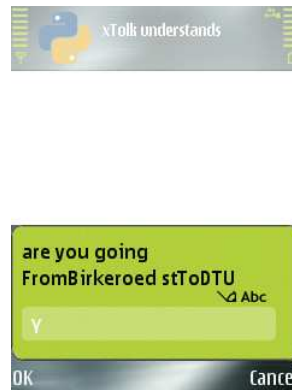
(a) Select the Arrival Estimate function



(b) User is required to confirm the route



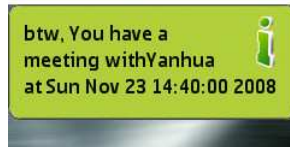
(c) Alert popped out when no route is found



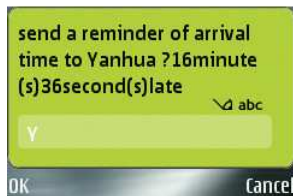
(d) Correct route is confirmed



(e) Estimated arrival time is shown



(f) xTolk provides a reminder of upcoming event



(g) xTolk proposes to send reminder to contact person



(h) The content of SMS reminder

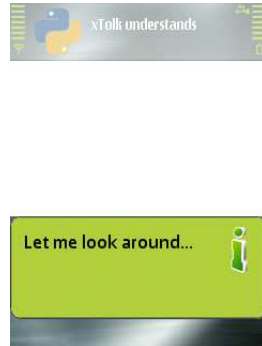
APPENDIX G

BlueMark Option

Functions provided by *BlueMark* option



(i) Select the BlueMark function



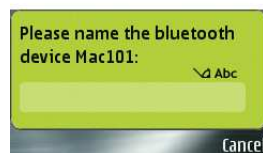
(j) A note popped out



(k) Note popped out when no devices found



(l) Note popped out when device recognized



(m) User required to name the detected device

Bibliography

- [1] Apache server project. <http://httpd.apache.org/>. Accessed 1:43PM Fr 21 Nov, 2008.
- [2] *Bluetooth Protocol Architecture*.
- [3] *DEVELOPING SUCCESSFUL MOBILE APPLICATIONS*.
- [4] *Global System for Mobile Communication*.
- [5] S. Brewster. Overcoming the lack of screen spaces on mobile computers. *Personal and Ubiquitous Computing*, 2002.
- [6] Nokia Corporation. S60 platform: Basics. Technical report, Nokia Corporation, 2006.
- [7] N. Eagle and A. Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 04(2):28 – 34, April - June 2005.
- [8] Mark Frydenberg. Earthquake maps and spinning photos: What is a mashup? Technical report.
- [9] Akihiro FUJII. Trends and issues in research on context awareness technologies for a ubiquitous network society. *Science and Technology Trends*, 2007.
- [10] Atkeson Jason Hong Sue Long Rob Kooper Mike Pinkerton Gregory D Abowd, Cristopher G. Cyberguide: a mobile context-aware tour guide. *Wireless Networks*, 3:421 – 433, October 1997.
- [11] P Tarasewich J Gong. Guidelines for handheld mobile device interface design.

-
- [12] Peter Reiher Jelena Mirkovic. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 2004.
- [13] Ferenc Dosa Johan Wikman and Mikko Tarkiainen. Personal website on a mobile phone. Nokia Corporation, May 2006.
- [14] Sunny Consolvo Beverly Harrison James A. Landay Jon Froehlich, Mike Y. Chen. My experience: A system for in situ tracing and capturing of user feedback on mobile phones. 2007.
- [15] John McAleely. Smallserv. <http://mysymbian.com/main/index2.php?m=200311cat=2>. Accessed 5:32PM Fr 14 Nov, 2008.
- [16] Giles John Nelson. *Context-Aware and Location Systems*. PhD thesis, Clare College, University of Cambridge, 1995.
- [17] Preetham Mysore Michael L. Littman Nishkam Ravi, Nikhil Dandekar.
- [18] Lucas Pettinati. Challenges of interface design for mobile devices. *Yahoo! User Experience Design*, 2004.
- [19] Nicholas Nicoloudis & Dennis Pratistha. .net compact framework mobile web server architecture. *.NET Compact Framework*, 2003.
- [20] Petro Soinenen Richard Bloor. Creating location-aware applications. 2008.
- [21] R. Schilit B., Adams N. Want. Context-aware computing applications. *1st International Workshop on Mobile Computing Systems and Applications*, pages 85 – 90, 1994.
- [22] Joe Barrett Tomi T. Ahonen. *Services for UMTS: Creating Killer Applications in 3G*. John Wiley & Sons, 2002.
- [23] Vodafone. The british mobile communications survey, January 2003.
- [24] M. Weiser. Some computer science problems in ubiquitous computing. *Communications of the ACM*, July 1993.