

# Context software

## A prototype platform for contextual mobile applications

Mika Raento

Basic Research Unit, Helsinki Institute for Information Technology  
Department of Computer Science, University of Helsinki  
`Mika.Raento@cs.Helsinki.FI`

## 1 Introduction

During the course of the past two years we in the Context project [1] have developed software for Nokia Series 60 Smartphones [2], for use in our project. What started out as data logger, grew later into a presence service for our user studies and now enables a range of different mobile, contextual applications to be put together.

As the functionality of the software has grown, so have its potential uses. To aid the reproducibility of our own results and get the maximum benefit from the work put into it, we have released the software under the Gnu General Public License GPL [3]. This article describes the features, architecture and possible applications of the software so that the research community can judge its suitability to their tasks.

## 2 Features

The prototype software can be used to sense, process, store and transfer contextual data as well as to use this data itself to attach to media or modifying its behaviour.

### 2.1 Sensing and mediating

Different kinds of contextual variables are available on a mobile phone relating to user behaviour, phone internal state, the GSM network and external sensors. The software in its basic form can sense:

- Current GSM Cell Id
- Bluetooth devices around it
- GPS data from a Bluetooth GPS receiver
- Phone profile
- Active phone application
- Phone idle/active time
- Charger status
- Incoming/Outgoing calls

- Incoming SMS
- User interaction with the Phonebook and recent call log
- Media captured with the device (photos, audio, video, text)
- Visualcodes [4] two-dimensional barcode recognition via software courtesy of Michael Rohs and Beat Gfeller

As the current Cell Id is not very meaningful for many applications, the Context software can mediate that data into:

- Selection of *important locations* implementing the method described in [5]
- Naming cells either via a network service or by asking the user

## 2.2 Communication with the outside world

Since the original purpose of the software was logging of contextual data and since the phone has quite limited storage space a robust and automatic upload of files created in the phone is provided. The upload can work in the background or triggered by the user, and it can either use the GPRS network or a local connection (Bluetooth or Infrared).

In addition to the 'as-is' uploading of files, uploads can be annotated with any and all of the contextual variables described above. This includes automatic or user-controlled upload of media captured on the phone, with contextual annotation.

In addition to file transfers the software can send and receive SMS and MMS, as well as reacting to incoming/outgoing calls. Different actions can be taken based on the sender/receiver of the SMS, MMS and calls as well as on the content of the SMS or MMS. For example any mobile service that can be used via text messages can be incorporated into the software (this is how the network based cell naming works).

To build the presence service we used the Jabber protocol [6]. The Context software can send and receive the jabber presence data. We use a custom XML-based presence description to encode different aspects (like location and phone profile) into the Jabber presence. This means that any Jabber-enabled software can be used to interface with the Context presence service. We can also receive Jabber instant messages and act on their content. There is a custom XML message that we use to alert the user that there is new information at a certain URL available.

## 2.3 Interfacing with other applications on the phone

Since we wanted to implement a presence service, we wanted to integrate the software with the call-making aspects of the phone. The Context software has a customizable Phonebook and Recent calls component that can override the phone built-in one.

To retrieve information from the internet, Context knows how to show an URL with a browser on the phone (both the 6600 built-in Opera browser, as well as the Doris browser installable on a 7650). The URL in the Jabber notification message can then be automatically shown in a browser.

### 3 Architecture

The software has grown fairly organically from a pure data logger with a few variables (2 KLOC) into the large collection of different features (40 KLOC) it is today. The original pure publish-subscribe architecture implementation isn't quite enough anymore, so we are in the process of reimplementing it in a Blackboard architecture [?].

Physically the software consists of four processes on the phone:

- Context\_log: which handles the sensing, storing, processing and acting upon of context data. This is always running in the background and handles also most communication with the outside world and the user.
- Contextbook: is the replacement Phonebook and Recent calls application. It uses the built-in Contact and Log engines, augmenting the user interface with presence information.
- ContextServer: handles contact with a Jabber server
- Starter: acts as a watchdog and error logger, starting and restarting Context\_log as necessary.

The whole implementation is done in C++ for the Symbian operating system [7] for access to the native phone facilities. The result is extremely non-portable, but Symbian provides maybe the only truly ubiquitous computing platform in the world.

### 4 Example applications

The features described above can be connected in many different ways. Here we describe some applications that have already been implemented by us on top of the basic framework.

#### 4.1 Presence

Some studies have indicated that as much as 50% of phone calls fail, in the sense that they do not reach the intended recipient or that the recipient did not want to be reached. Motivated by this, we have been conducting research into how the automatically sensible context of the recipient can be used to facilitate choosing the time and medium for communicating.

As one of the first steps, we have built a presence service that can transmit the values of some context variables to the recipients communication partners, 'Buddies'. Figure 1 shows the augmented Phonebook and the detailed view of a contact in that phonebook. The idea is that the potential caller can use their background knowledge of the recipient and human cognitive abilities to make inferences about the meaning of the contextual information shown beyond its face value.

At any time the user who is publishing presence information can look at their own information in exactly the same way it is shown to others, so they know

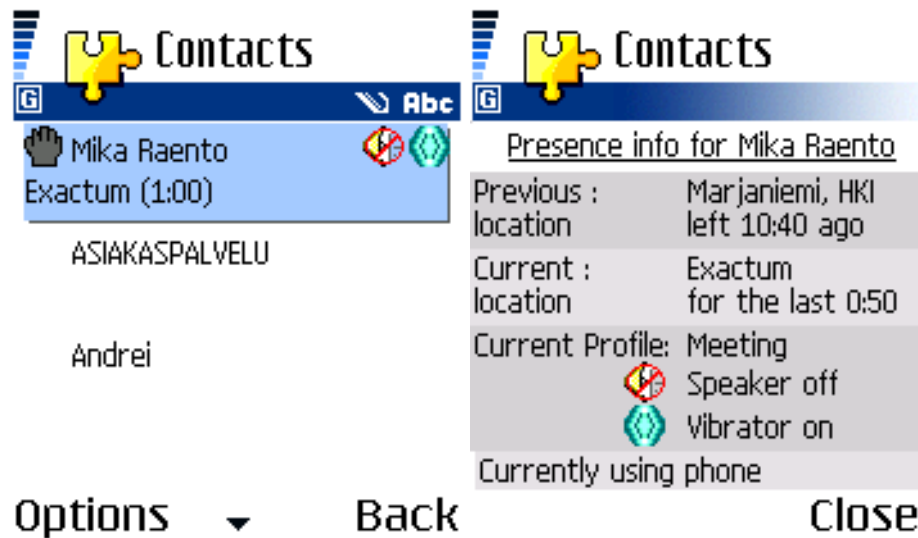


Fig. 1. Phonebook augmented with presence information

what information they are disclosing. They are also able to disable the service at any time. These facts do not mean that there are no privacy issues in the service, but that is one of the aspects we are actively studying.

#### 4.2 Sharing of contextual media

Figure 2 shows the interaction needed for uploading contextually (and manually) annotated media from the phone. Notable here is how simple the task is: after capturing media the upload prompt is displayed, and if the user is happy with the categorization the media can be uploaded with a single joystick click.

Of course the immediate upload and annotation is not unproblematic, as the user does not always have the time to do even that simple thing (for example, they might want to immediately take another photo), but it does facilitate a range of activities. Most picture-taking studies have shown that people are not very willing to categorize their pictures afterwards, so such an immediate and easy categorization (especially with the automatic categorization possibilities from e.g. the location and time) has value.

The photos thus uploaded can be immediately and automatically shared to an interest group. The Context software does not produce notifications itself, but the server receiving the media can push notifications to other phones running Context via Jabber. Aware [8] by John Evans is a server platform that can do this. Figure 5 shows how the notifications are shown to the user.

Of course the software can be used for pure MediaBlogging (like PhotoBlogging or Moblogging, but with any media). You can write your own server side scripts, or use Aware. Figure 4 shows an example MediaBlog from Aware.

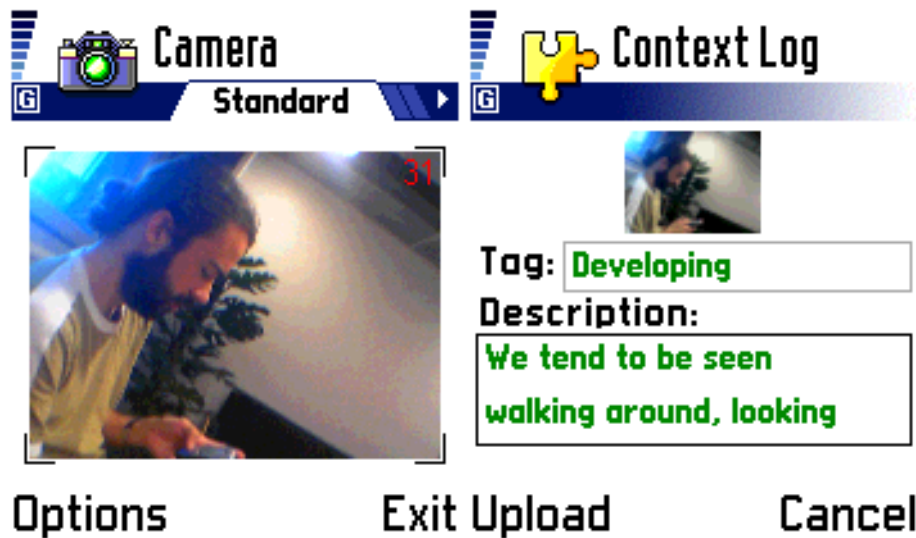


Fig. 2. Easy upload of annotated media

#### 4.3 Information retrieval: Mobile Journey Planner

The Helsinki Metropolitan area Public Transport Journey Planner at <http://aikataulut.ytv.fi/> is one of the most successful Finnish web-based information services. It works on a mobile web browser as well, but the interaction has not been designed for small-screen, low-bandwidth devices. The interaction on e.g. the Nokia 6600 goes (best case):

1. Switch to menu, Open browser (8 sec)
2. Open bookmark (opens GPRS connection, fetches page, 45 sec)
3. Key in start and endpoints (30-60 sec)
4. Do search (15 sec)

for a total of about 2 minutes for a single search. We've implemented an alternative interface, shown in figure ?? with locally stored endpoints, which enables the following interaction sequence (best case):

1. Switch to our app, select Reittipas (5 sec, opens GPRS in the background)
2. Select endpoints from list (10 sec)
3. Fetch results (opens browser with URL, fetches page, 10 sec)

for less than half a minute, with much less waiting time.

The current implementation just proposes endpoints by frequency. It could be extended to use the location instead, learning which endpoints correspond to which locations. Other location-based or contextual information retrieval strategies could easily be tested as well.

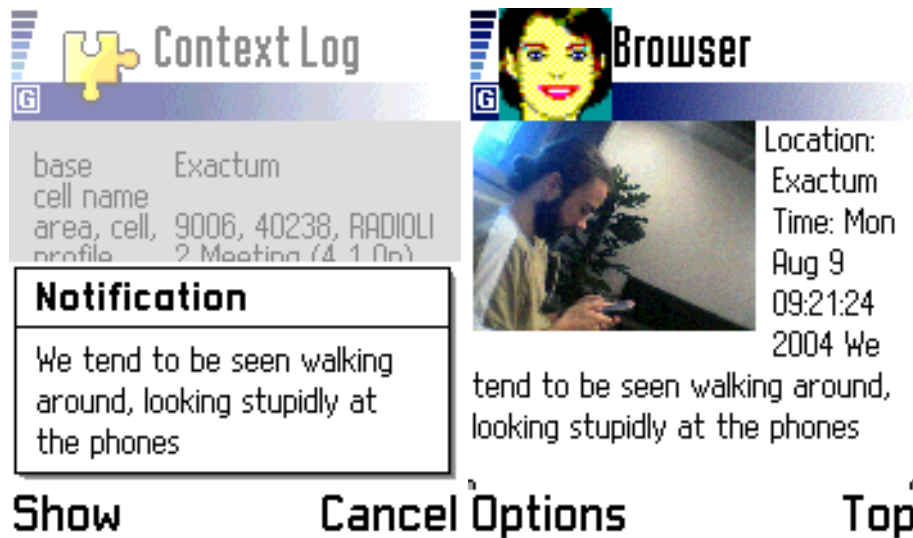


Fig. 3. Notification of new media

## 5 Other users

The Context software is already in use in several research projects in Finland and in the US:

- The Context project for research in using context in improving mobile communication
- Aware - Experimentation and research into collective publication and syndication of mobile media at Media Lab - University of Art and Design, Helsinki
- at MIT Media Lab for a study on organizational modeling, social network analysis, and complex social systems
- by the Garage Cinema Research at the University of California at Berkeley School of Information Management and Systems for research in automatic annotation and sharing of mobile media

## 6 Using Context for your research

Many aspects of the software are run-time customizable, and you may be able to use the provided binaries for your research directly. This is especially true if you want to just use the features presented in a different social or interactional context. You can get a ready-made build with installation instructions from <http://www.cs.helsinki.fi/group/context/latest/>.

As the software is licensed under GPL, you can of course freely make changes to it to suit your needs. The sources with building instructions are at <http://www.cs.helsinki.fi/group/context/latest/>.

# aware

1234567891011121314>>


Create feed from current view


Packet [8996]

Location  
+ GSM Finland 9000-62125 Casa

Proximity  
+ Nameless person 0002ee9f901f

Keywords  
+ Diary




sent by  Mika  
Thu +23+Sep 2004 17:04 GMT


Packet [8994]

Location  
+ GSM Finland 9007-39495

Proximity  
+ Nameless person 0002ee9f901f

Keywords  
+ Diary




Jarmo  
sent by  Mika  
Tue +21+Sep 2004 15:37 GMT


Packet [8993]

Location  
+ GSM Finland 9111-40257

Proximity  
+ Nameless person 0002ee9f901f

Keywords  
+ Diary

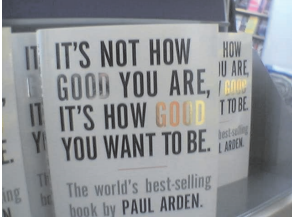



Mushrooms  
sent by  Mika  
Sun +19+Sep 2004 09:19 GMT

Packet [8836]

Location  
+ GSM United Kingdom 52-59201

Keywords  
+ MobileHCI




Who picked that emphasis  
sent by  Mika  
Fri +17+Sep 2004 04:05 GMT


Packet [8821]

Location  
+ GSM United Kingdom 63-53163 Glasgow center west

Proximity  
+ Nokia 6310i [person]  
+ Nokia 8910i [person]  
+ Nokia3650 [person]

Keywords  
+ MobileHCI




Real food  
sent by  Mika  
Thu +16+Sep 2004 16:59 GMT


Packet [8820]


Location  
+ GSM United Kingdom 63-53162

Proximity  
+ Nameless person 0002ee9f901f

Keywords  
+ MobileHCI



Usability  
sent by  Mika  
Thu +16+Sep 2004 15:54 GMT

RSS feed of current view 


XML feed of current view 

Fig. 4. MediaBlog



Fig. 5. The Mobile Journey Planner

[//www.cs.helsinki.fi/group/context/latest/source/](http://www.cs.helsinki.fi/group/context/latest/source/). As we are currently rewriting the structure of the code, we'd like to hear from you if you are going to be modifying the source so that we can coordinate the changes. Realistically, to make changes in the system you will need to know basic Symbian C++ programming, but you can ignore a lot of the hard-to-use APIs and build on top of the framework we have.

The aware server platform is also available for external use. Contact the aware group via <http://aware.uiah.fi/>.

## 7 Conclusions

As a Symbian Smartphone provides an off-the-self, ubiquitous (as users in many countries *always* carry a phone with them) computing platform it presents researches an unprecedented opportunity to cheap and wide scale experiments with contextual and/or mobile applications. The Context prototype software has a number of features needed in such applications, and provides a coherent framework for extensions, is available as source code and is already widely used.

## References

1. Context project: Context recognition by user situation data analysis (Context). Website <http://www.cs.helsinki.fi/group/context/> (2004)
2. Nokia: Series 60 platform. Website [http://www.forum.nokia.com/main/1,6566,010\\_40,00.html](http://www.forum.nokia.com/main/1,6566,010_40,00.html) (2003)



3. Free Software Foundation: Gnu general public license, version 2. Website <http://www.gnu.org/copyleft/gpl.html> (1991)
4. Rohs, M.: Real-world interaction with camera-phones. In: 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004), Tokyo, Japan (2004)
5. Laasonen, K., Raento, M., Toivonen, H.: Adaptive on-device location recognition. In: Proceedings of the 2nd International Conference on Pervasive Computing (Pervasive 2004). Lecture Notes in Computer Science, Springer-Verlag (2004)
6. : Rfc 3920. extensible messaging and presence protocol (xmpp): Core (2004)
7. Symbian Ltd.: Symbian developer library. Website <http://www.symbian.com/developer/techlib/sdl.html> (2004)
8. John Evans and Andrew Paterson and Aki-Ville Pöykiö and Marcus Ort: aware. Website <http://aware.uiah.fi/> (2004)