BeTelGeuse – A Tool for Bluetooth Data Gathering

Petteri Nurmi, Joonas Kukkonen, Eemil Lagerspetz,
Jukka Suomela, Patrik Floréen
Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki
P. O. Box 68, FI-00014 University of Helsinki, Finland
firstname.lastname@cs.helsinki.fi

ABSTRACT

In ubiquitous computing, activity-related data is typically gathered using customized sensing equipment that give physiological measurements. Unfortunately, such systems are often proprietary or expensive to obtain. Recently, the decrease in the prices of Bluetooth chips has made Bluetooth sensors a viable alternative. In previous research, various systems for gathering data from Bluetooth sensors have been proposed, but they are usually limited to a specific set of sensors or to a specific runtime platform. To address these shortcomings, we have developed BeTelGeuse, a tool for Bluetooth data gathering. BeTelGeuse turns a standard mobile device such as a cellular phone into a relay node which gathers data from a body area network over Bluetooth, and forwards it to a remote server over a mobile data service such as GPRS.

Categories and Subject Descriptors

J.7 [**Computer Applications**]: Computers in Other Systems; C.3 [**Computer Systems Organization**]: Special-purpose and Application-based Systems; C.2.4 [**Computer Systems Organization**]: Computer-communication Networks—*Distributed Systems*

General Terms

Design, Experimentation

Keywords

Data gathering, Body area networks, Sensor networks, Bluetooth, Java

1. INTRODUCTION

Many compelling applications of context-aware computing rely on information about human activity. For example, monitoring physical activities is fundamental in applications of pervasive health care (e.g., [18]). As another example, user activity is used in adaptive interfaces for inferring the user's current goals and information needs (e.g., [13]).

Activity information is seldom directly accessible, but must be inferred from physiological measurements such as galvanic skin response, heart rate, ECG, and acceleration (e.g., [20]). Currently, the dominant approach for gathering physiological sensor data is to build customized sensor boards, armbands, or intelligent fabrics equipped with sensors. Unfortunately, these systems are often proprietary or expensive to obtain.

The decrease in the prices of Bluetooth chips has made Bluetooth sensors an attractive alternative. Many systems that gather data from Bluetooth sensors have already been proposed, but the systems are usually limited to a specific set of sensors or to a specific platform. This clearly limits research as (i) the data gathering tool determines the sensors that can be used and (ii) user tests must be carried out using devices with specific platforms.

To facilitate research in mobile context-aware computing, we have developed *BeTelGeuse*, a tool for Bluetooth data gathering. BeTelGeuse runs on a mobile device (e.g., a cellular phone, a PDA or a laptop computer), and gathers data from a body area network over Bluetooth. With a transmitter plug-in, BeTelGeuse can transmit the gathered data to a server for further processing; see Figure 1. BeTelGeuse is available under the LGPL license [10] and it can be obtained free of charge from the project web site [19].

The main advantages of BeTelGeuse are that (i) it can be extended to use new sensors, (ii) it runs on several platforms, and (iii) it is freely available. Due to its modular design, BeTelGeuse can be integrated with tools that provide functionalities for data dissemination and analysis.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 gives a high-level overview of the system and describes the platform requirements. Section 4 presents the core functions and Section 5 discusses the plug-in components. Section 6 presents some applications where BeTelGeuse can be used. Section 7 concludes the paper and summarizes the main features of BeTelGeuse.

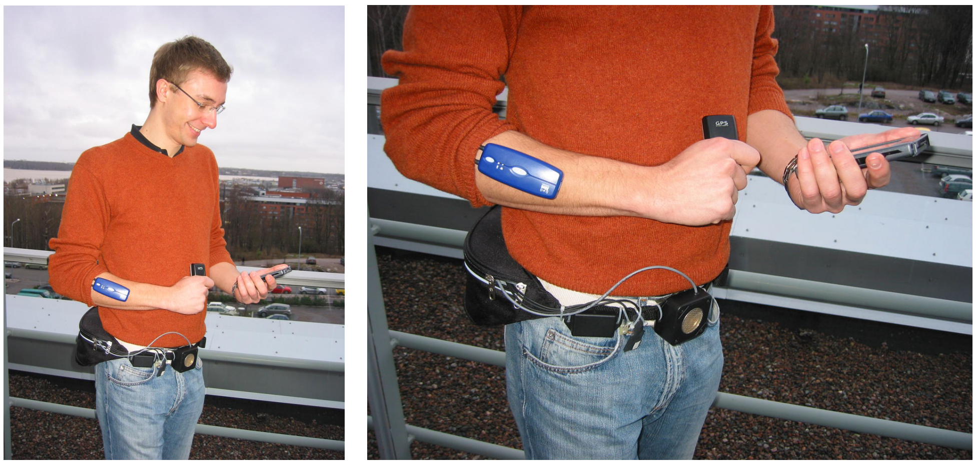
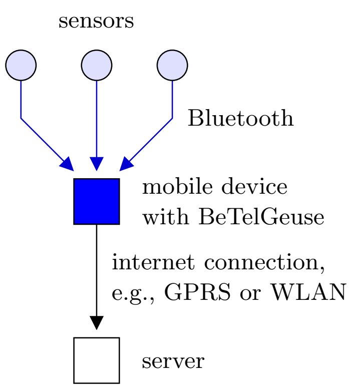



Figure 1: Using BeTelGeuse. Our subject carries a PDA device which runs BeTelGeuse. BeTelGeuse gathers data from the following Bluetooth devices: a GPS device (in the right hand), Alive Heart Monitor (attached to the right arm) and I-CubeX sensor system (on the waist, from left to right: the controller and an orientation sensor in the bag; a 3D acceleration sensor, an ambient light sensor, a temperature and humidity sensor, and an ultrasound distance sensor attached to the belt). See Table 1 for more information on the sensors.

2. RELATED WORK

Data gathering is the basis of research in ubiquitous computing and other closely related fields. As a consequence, much work has been conducted on building different kinds of sensing devices and tools for data gathering. For example, the Muffin terminal [31] integrates a number of sensors, including an air temperature sensor, a humidity sensor, an alcohol gas sensor, a pulse sensor, a compass, and a linear 3-axis accelerometer. Custom sensor boxes have been used, among others, at Intel Research Seattle [20] and at Carnegie Mellon University [26].

Tools that gather data from Bluetooth sensors have also been proposed. Contrary to BeTelGeuse, these tools are usually confined to a specific platform. For example, ContextPhone [24] can read GPS information and it provides information about nearby Bluetooth devices. Another similar tool is Context Watcher [17], which can read data from a GPS and from a proprietary device that provides heart rate, distance and speed information. Both tools run only on Nokia S60 2nd edition mobile phones, whereas our tool works on several platforms.

Another closely related field is wearable computing. Especially in the field of pervasive health care, various systems for health monitoring have been proposed; see, for example, the Body Media Sensewear armband PRO [29]. A problem with these systems is that they seldom allow online data gathering, which reduces their applicability to mobile computing.

Closest to our approach is the Personal Mobile Hub [14]. However, the implementations of the Personal Mobile Hub focus on specific devices (custom-built hardware and Sony Ericsson P800 mobile phones) while BeTelGeuse runs on several standard mobile devices. Intel Place Lab [27] provides a platform-independent tool for data gathering, but the focus of Place Lab is on location data, whereas we focus on physiological data.

3. OVERVIEW

The high-level system structure of BeTelGeuse is inspired by the microkernel architecture pattern [6, §2.5]. We have a separate core, which offers minimal functionalities needed to run the tool. The core functions are introduced in Section 4.

The core also defines interfaces that components offering extended functionalities must follow. This allows us to have a single implementation of the main functionalities, while having custom extensions for different runtime platforms. For example, we have implemented separate graphical user interfaces, one for PCs (see Figure 2) and one for mobile phones (see Figure 3). We have also implemented a plug-in component for data dissemination (see Section 5).

3.1 Runtime Platform

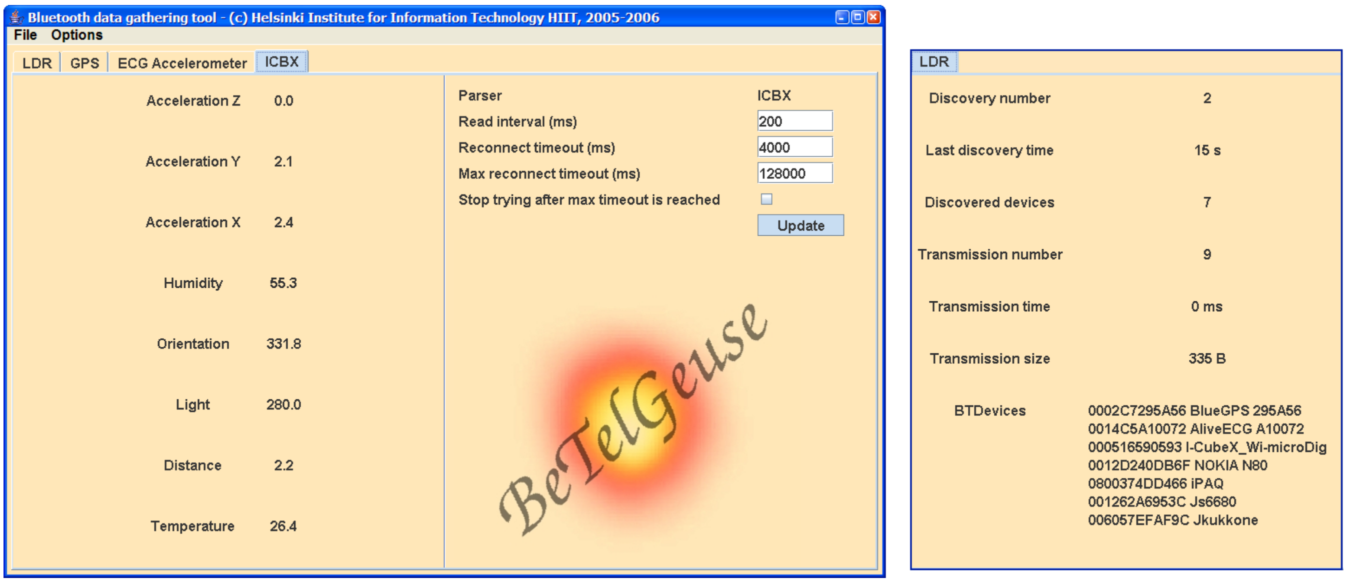
In order to achieve interoperability, we have selected Java as our implementation language, and we only use features that are commonly available in the Java implementations of most mobile and desktop devices.

More specifically, the core of BeTelGeuse is compatible with mobile systems that conform to the MIDP 2.0 [16] and CLDC 1.1 [28] specifications, and it is also compatible with desktop systems with Java 1.3 or later. Additionally, a JSR-82 [21] compliant Java Bluetooth stack is required.

In practice, the platform requirements can be satisfied on common devices as follows:

1. Most *mobile phones* which support Java and Bluetooth support the above-mentioned specifications. BeTelGeuse has been tested with Nokia 6680, Nokia N80, and Sony Ericsson W800i devices.

2. On *Linux PCs* we have used a standard Java installation with *AvetanaBluetooth* [2] as the Java Bluetooth stack. AvetanaBluetooth requires the *BlueZ* [12] Blue-

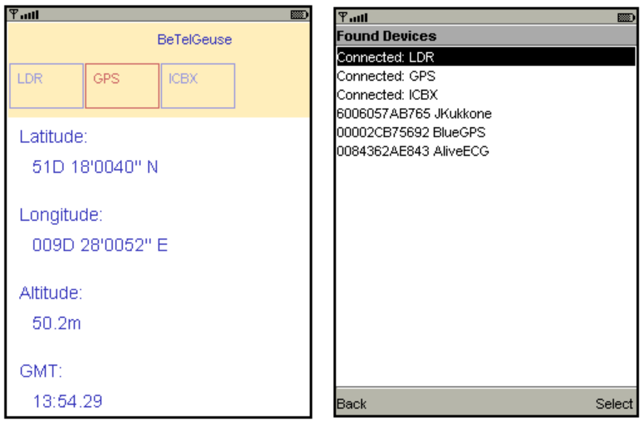

Bluetooth data gathering tool - (c) Helsinki Institute for Information Technology HIIT, 2005-2006
File Options
LDR GPS ECG Accelerometer ICBX
Update
BeTelGeuse
LDR

Acceleration Z	0.0
Acceleration Y	2.1
Acceleration X	2.4
Humidity	55.3
Orientation	331.8
Light	280.0
Distance	2.2
Temperature	26.4

Parser	ICBX
Read interval (ms)	200
Reconnect timeout (ms)	4000
Max reconnect timeout (ms)	128000
Stop trying after max timeout is reached	☐

Discovery number	2
Last discovery time	15 s
Discovered devices	7
Transmission number	9
Transmission time	0 ms
Transmission size	335 B
BTDevices	0002C7295A56 BlueGPS 295A56 0014C5A10072 AliveECG A10072 000516590593 I-CubeX_Wi-microDig 0012D240DB6F NOKIA N80 0800374DD466 iPAQ 001262A6953C Js6680 006057EFAF9C Jkukkone

Figure 2: The PC version of BeTelGeuse



Latitude:
51D 18'0040" N
Longitude:
009D 28'0052" E
Altitude:
50.2m
GMT:
13:54.29

Connected: LDR
Connected: GPS
Connected: ICBX
6006057AB765 JKukkone
00002CB75692 BlueGPS
0084362AE843 AliveECG

Figure 3: The mobile phone version of BeTelGeuse

tooth stack on Linux. Both BlueZ and the Linux version of AvetanaBluetooth are freely available.

3. On *Windows PCs* we have used a standard Java installation with *Bluesock* [4] Bluetooth stack. Bluesock additionally requires that the operating system has the *Microsoft Bluetooth stack* installed and it works only with Bluetooth beacons that support the Microsoft Bluetooth stack. Bluesock is freely available, and the Microsoft Bluetooth stack is part of, e.g., Windows XP Service Pack 2.

4. On *PDAs with Microsoft Windows Mobile* we have used Java Personal Profile environment and a commercial version of AvetanaBluetooth. BeTelGeuse has been tested with a Hewlett Packard hx4700 PDA which runs Microsoft Windows Mobile 2003 Second Edition.

For more information on system requirements, set up, and usage of BeTelGeuse, see the user manual [19].

3.2 Technical Testing

We have conducted technical tests related to the power consumption of BeTelGeuse in some typical use cases. The tests described in this section were performed by a single user with a Nokia 6680 mobile phone and BeTelGeuse version 1.0. The tests focused on continuous data gathering, where Bluetooth sensors stream new data constantly. The testing situations included commuting, walking in areas with several active Bluetooth devices, and spending time at the office. In the tests, we used a GPS device and an Alive Heart Monitor device (see Table 1). A local device reader (see Section 4.5) was used to provide Bluetooth proximity and GSM cell data. The gathered data was transmitted to a server over GPRS.

In our tests, the battery of the Nokia 6680 mobile phone lasted between 3.5 and 4.5 hours. The battery lifetime depends on the details of the configuration (the data rate from BeTelGeuse to the server; the number of active sensors; whether device discovery is used; whether the phone runs with a screen saver on), but none of these factors affected the battery lifetime considerably. We concluded that experiments using continuous data gathering can be performed with a duration of a few hours; however, the battery lifetime hinders longer-running experiments.

In addition to the battery lifetime, we studied the effect of Bluetooth device discovery on continuous Bluetooth data gathering. We focused on urban environments with several moving Bluetooth devices. During the tests, device discovery time varied from 10 seconds to over 4 minutes. Device discoveries slowed down the Bluetooth communication or even blocked BeTelGeuse from accessing active Bluetooth connections for the duration of the discovery. Long-lasting discovery scans also slowed down other components, e.g., the transmitter plug-in. Generally, this was not a problem, since Java Bluetooth implementations buffer the received messages. However, long-lasting device discoveries can cause the Bluetooth buffer to overflow on some devices.

Sensor	Examples of measured data
GPS[a]	latitude, longitude, altitude, time, number of satellites
Alive Heart Monitor[b]	ECG, 3-axis acceleration
I-CubeX[c]	distance (ultrasound), 3-axis acceleration, temperature, humidity, orientation, background light
Place Lab[d]	GSM cell information: identifier, area, network and country codes, network name, signal strength
local device	Bluetooth proximity information (when periodic device discovery is enabled)

[a]Bluetooth GPS sensors which use the NMEA 0183 protocol and produce at least GGA messages (see, e.g., [9, §8.4]).
[b]Product website: `http://www.alivetec.com/`
[c]Product website: `http://infusionsystems.com/`
[d]Product website: `http://placelab.org/`. Place Lab GSM server works only on Nokia S60 2nd edition mobile phones.

Table 1: List of currently supported sensors

4. CORE FUNCTIONS

This section introduce the core functions of BeTelGeuse. For technical details we refer to the BeTelGeuse website [19].

4.1 Device Discovery

Bluetooth device discovery is used to detect which Bluetooth sensors are currently available. The list of reachable Bluetooth devices may also provide useful information on the environment; for example, Bluetooth proximity information can be used for behavioral analysis (see Section 6).

Device discovery in BeTelGeuse can run in one of two modes:

1. The device discovery is performed periodically; the user can configure the time between subsequent scans. This mode makes it possible to gather Bluetooth proximity information. In this mode it is also possible to discover and use new sensors automatically; for example, the user may have switched on a new sensor device, or there may be a sensor device installed in the building that the user has just entered.
2. The device discovery is performed on system startup; the user can also manually trigger device discovery. This mode is useful in experiments where the set of sensors is fixed and Bluetooth proximity information is not needed. This mode can be used to conserve energy and to avoid conflicts between Bluetooth data gathering and Bluetooth device discovery (see Section 3.2).

The device discovery returns a list of Bluetooth addresses and, if requested, their friendly names. The discovery also returns statistics regarding the latest device discovery: the execution time, the number of devices found, and the number of new devices. External components can subscribe to device discovery information. For example, the local device reader (see Section 4.5) uses the device discovery information to provide a list of nearby Bluetooth devices.

4.2 Connection Management

BeTelGeuse manages Bluetooth connections to the discovered sensors automatically. We use device mappings to map Bluetooth devices to device types, and we attempt to connect to a device only if its device type is supported (note that the Bluetooth specification limits the number of active Bluetooth connections to seven [5, §B.1]). A detailed explanation of the device mappings is given in Section 4.4.

Connections to sensor devices may be lost for various reasons, either temporarily or permanently: radio communication may be blocked due to interference or obstacles; sensor devices or their batteries may fail; sensors installed in the environment may become unreachable as the user moves.

If a connection is lost, BeTelGeuse tries to automatically re-establish the connection. We use a backoff algorithm that exponentially increases the time between subsequent attempts. The algorithm uses three user-configurable parameters. The first parameter determines the time to the first attempt. The second parameter limits the maximum time between the attempts. The third parameter determines what to do when the time limit is reached: we can either close the reader or keep retrying at the specified maximum intervals.

4.3 Readers

Individual sensors are handled by *Readers*. There is one reader for each sensor type. Each reader has a unique device type identifier, which is used in the device mappings (see Section 4.4). The readers that we have implemented are listed in Table 1.

The core of BeTelGeuse is responsible for opening a Bluetooth connection to the sensor device. The reader uses a socket to communicate with the sensor device. The reader (i) handles the device-specific protocol that is used over the Bluetooth socket, (ii) parses the device-specific messages, and (iii) maps sensor data into key-value pairs.

External components can obtain the sensor data by using a publish-subscribe approach or by pulling data directly from the reader.

In order to add support for a new sensor, the developer implements a new reader, assigns a unique device type identifier to the reader, and registers it with BeTelGeuse; see the developers' guide [19] for detailed instructions.

4.4 Mappings

To be able to automatically instantiate the appropriate readers for discovered Bluetooth devices, we use device mappings. The mappings are rules that specify how Bluetooth devices are associated with device types. Each device type is assigned to one reader. Currently we support address-based and friendly name-based mappings. The former maps an individual Bluetooth address to a specific device type. The latter allows specifying a string that the friendly name of a device must contain for it to be mapped to a specific device type.

A simple example of a friendly name-based mapping is that when the friendly name contains the string “GPS” the sensor is identified as a GPS device and a GPS reader is automatically instantiated for the sensor. If necessary, address-based mappings can be used to override friendly name-based mappings.

The mappings are stored locally on the device. Because the APIs for storing information differ across platforms, the storage of the mappings is outside the core of BeTelGeuse; instead the core offers a specific interface for obtaining information about the mappings.

4.5 Local Device Reader

In order to obtain information from the device that runs BeTelGeuse, we have implemented a *Local Device Reader*. The local device reader provides Bluetooth proximity information, and it is able to read data from localhost socket connections. As an example of the latter case, on Nokia S60 2nd edition mobile phones we use the Place Lab [27] GSM server and read GSM cell information from a local socket.

5. PLUG-INS

The modular architecture of BeTelGeuse offers several possibilities for extending BeTelGeuse. In the previous section, we have seen two ways to extend BeTelGeuse: implementing new readers in order to support new Bluetooth sensor devices (Section 4.3), and using the local device reader to acquire information from the local device (Section 4.5). In this section, we present another possibility: *plug-ins*. The data transmitter serves as an example of a plug-in.

5.1 Data Transmitter Plug-in

Data dissemination is outside the core of BeTelGeuse. This makes it possible to integrate BeTelGeuse with existing tools (e.g., JCAF [3] or Context Toolkit [25]) or to implement new tools. As part of our GUIs, we have implemented a data transmitter plug-in that sends all data to a server that stores the data in a MySQL database.

The data is transmitted using the internet communication capabilities of the device, for example, a wireless LAN or a mobile data service such as GPRS. The data transmitter uses a custom protocol on top of a TCP connection. The protocol borrows ideas from lightweight sensor communication protocols, especially from the Simple Sensor Interface (SSI) protocol [15]. The specification of the protocol is available from the BeTelGeuse website [19].

Currently the transmitter plug-in does not support server authentication or privacy-based filtering of the data, but these functionalities can be added by extending the current implementation or by implementing a new plug-in for transmitting data.

6. USAGE SCENARIOS

To demonstrate the usefulness of BeTelGeuse, this section presents various applications where BeTelGeuse can be used to facilitate research.

Activity recognition: An intuitive use case for BeTelGeuse is to gather data for activity recognition. Contrary to most previous work on activity recognition, BeTelGeuse makes it possible to (i) use off-the-shelf sensors and (ii) test activity recognition algorithms in an online setting.

Behavioral analysis: Recently, Bluetooth-enabled mobile phones have been used to analyze daily patterns of individuals and their social relationships [8]. Our tool is also suited for carrying out this kind of studies.

Context-dependent user modeling: In this domain, the goal is to build models that can be used to personalize mobile applications by taking into account the situation of the user and her personal preferences [23]. BeTelGeuse supports context-dependent user modeling by facilitating the gathering of situational information.

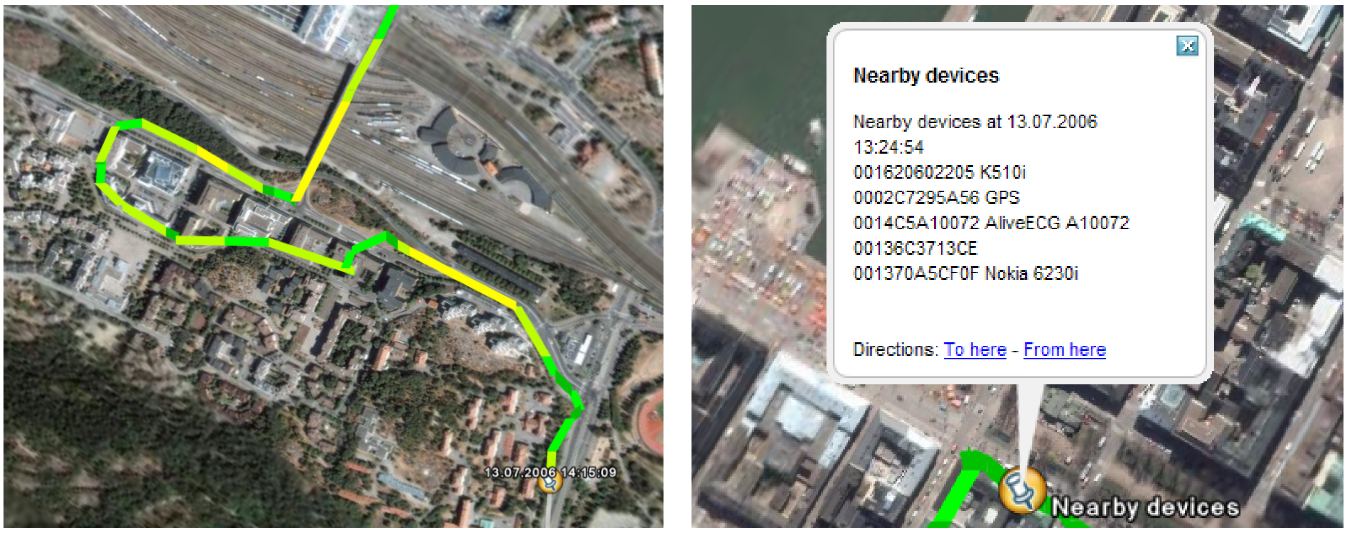
Experience sampling: Experience sampling [7] is an approach for evaluating ubiquitous computing applications. In a study, participants are occasionally alerted to fill in a brief questionnaire that may involve simple multiple-choice questions or open questions. Central to the method is that the context of the user can be used to select when the questionnaires are shown to the user, which facilitates gathering a representative sample of responses. As BeTelGeuse gathers situational data, it is well suited as a background application for experience sampling studies. We are currently implementing an experience sampling plug-in for BeTelGeuse.

GSM positioning: Most GSM positioning algorithms are based on GSM identifiers and signal strength information [30]. In addition, the algorithms typically require a separate training phase where also the GPS coordinates of the user are gathered. Since BeTelGeuse supports gathering both GPS and GSM information, it can be used to gather positioning data and to test positioning algorithms. Unfortunately though, most mobile phones are unsuited for positioning as the application programming interfaces do not allow accessing detailed enough GSM signal information.

Location clustering: Another application related to location data is to identify meaningful places from raw location measurements [1, 22]. BeTelGeuse can gather both GSM and GPS data and thus it can be used also for location clustering.

A proxy for testing context-aware applications: Our tool can be used as a data source for external applications and thus it can be used to test, e.g., novel context-aware applications and interaction techniques.

The above use cases have been used to guide the specification and design of the BeTelGeuse system. In the following, we present a simple, concrete use case for BeTelGeuse.



Nearby devices

Nearby devices at 13.07.2006
13:24:54
001620602205 K510i
0002C7295A56 GPS
0014C5A10072 AliveECG A10072
00136C3713CE
001370A5CF0F Nokia 6230i

Directions: To here - From here

Figure 4: Using Google Earth to visualize (a) movements of the user and (b) Bluetooth proximity information

6.1 Use Case: Context Visualization

A useful way to convey the gathered information to the user is to visualize it. This can be used, e.g., in applications for sharing information among users, and in user studies for measuring the acceptance of users to expose private information. In addition, data visualization is usually the first step in exploratory data analysis.

Although BeTelGeuse can be used to gather data in various kinds of environments, our main focus is on mobile environments. In addition to the activity of a user, her location is an important and intuitive source of context information in mobile environments. Therefore, we decided to use Google Earth[1] for visualization. While the geographic features of Google Earth are not as good as in commercial geographic information systems, the fact that Google Earth is freely available is a major advantage.

Google Earth provides an API that makes it possible to show location-based information on top of satellite images. The API uses an XML format called KML [11]. Google Earth can visualize information that is provided by third party servers in KML format. To integrate BeTelGeuse with Google Earth, we have built a server that transforms the data gathered by BeTelGeuse into KML format. Two examples of how the data visualization looks like are given in Figure 4. The example in Figure 4 (a) shows movements of the user. The paths are colored by the speed of the user. Figure 4 (b) shows also nearby Bluetooth devices.

7. CONCLUSIONS

In this paper we have presented BeTelGeuse, a tool for Bluetooth data gathering in body area networks and similar applications. One of the goals of BeTelGeuse is to make data gathering as easy as possible. BeTelGeuse has a modular architecture, which makes it possible to extend the tool with additional features. Contrary to previous work, our tool is not tied to a specific mobile device or to a specific set of sensors.

[1]Product website: `http://earth.google.com/`

We conclude by summarizing the main features of BeTelGeuse:

- automatic discovery of new devices
- automatic instantiation of readers once the mappings between Bluetooth addresses and device types have been specified
- automatic connection management; the tool attempts to re-establish lost connections automatically
- possibility to read data from the local device using localhost sockets
- possibility to access Bluetooth proximity information
- plug-in architecture: possibility to extend the tool with new components
- not tied to any particular context representation
- runs on many modern mobile phones and other mobile devices
- facilitates research in various applications of context-aware computing.

8. ACKNOWLEDGMENTS

This work has been performed in the framework of the IST project IST-2004-511607 MobiLife, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues, although the views expressed are those of the authors and do not necessarily represent the project. This work was supported in part by Helsinki Graduate School in Computer Science and Engineering (Hecse).

The authors thank Greger Lindén for interesting discussions and for commenting earlier versions of the paper; Johan Koolwaaij, Fabiola Lopez, Jan Storgårds, Dario Melpignano, and Luca Galli for their support and for providing access to sensors; and Fredrik Boström for testing.

9. REFERENCES

[1] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[2] Avetana GmbH, Ettlingen, Germany. *AvetanaBluetooth – JSR-82 implementation*, Apr. 2006. Available: `http://www.avetana-gmbh.de/avetana-gmbh/produkte/jsr82.eng.xml` and `http://sourceforge.net/projects/avetanabt/` [2007-01-07].

[3] J. E. Bardram. The Java Context Awareness Framework (JCAF) – a service infrastructure and programming framework for context-aware applications. In *Proc. 3rd International Conference on Pervasive Computing (PERVASIVE, Munich, Germany, May 2005)*, volume 3468 of *Lecture Notes in Computer Science*, pages 98–115, Berlin, Germany, 2005. Springer-Verlag.

[4] Bluesock developers. *Bluesock – Bluetooth for Java*, Aug. 2005. Available: `https://bluesock.dev.java.net/` [2007-01-07].

[5] Bluetooth SIG, Inc. *Specification of the Bluetooth system, Version 2.0, Specification Volume 2: Core System Package [Controller Volume]*, Nov. 2004. Available: `http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/` [2007-01-07].

[6] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *Pattern-oriented Software Architecture: A System of Patterns.* John Wiley & Sons Ltd, Chichester, UK, 1996.

[7] S. Consolvo and M. Walker. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing*, 2(2):24–31, 2003.

[8] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[9] A. El-Rabbany. *Introduction to GPS: the Global Positioning System.* Artech House, Norwood, MA, USA, 2002.

[10] Free Software Foundation, Inc., Boston, MA, USA. *GNU Lesser General Public License, Version 2.1*, Feb. 1999. Available: `http://www.gnu.org/licenses/lgpl.html` [2007-01-07].

[11] Google Inc., Mountain View, CA, USA. *Google Earth KML 2.0 Specification, Document Version 1.001*, 2007. Available: `http://earth.google.com/kml/` [2007-01-07].

[12] M. Holtmann, M. Krasnyansky, et al. *BlueZ – Official Linux Bluetooth protocol stack.* BlueZ Project, Dec. 2006. Available: `http://www.bluez.org/` [2007-01-07].

[13] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proc. 14th Conference on Uncertainty in Artificial Intelligence (UAI, Madison, WI, USA, July 1998)*, pages 256–265, San Francisco, CA, USA, 1998. Morgan Kaufmann.

[14] D. Husemann, C. Narayanaswa, and M. Nidd. Personal mobile hub. In *Proc. 8th International Symposium on Wearable Computers (ISWC, Arlington, VA, USA, October–November 2004)*, pages 85–91, Washington, DC, USA, 2004. IEEE Computer Society.

[15] J. Hyyryläinen, I. Jantunen, et al. *SSI Protocol Specification, Version 1.0.* Nokia Research Center, Helsinki, Finland, Apr. 2005. Available: `http://www.ssi-protocol.net/` [2007-01-07].

[16] JSR 118 Expert Group. *Mobile Information Device Profile for Java 2 Micro Edition, Version 2.0*, Nov. 2002. Available: `http://jcp.org/aboutJava/communityprocess/final/jsr118/` [2007-01-15].

[17] J. Koolwaaij, A. Tarlano, M. Luther, P. Nurmi, B. Mrohs, A. Battestini, and R. Vaidya. Context Watcher – sharing context information in everyday life. In *Proc. 2nd IASTED conference on Web Technologies, Applications and Services (WTAS, Calgary, Alberta, Canada, July 2006)*, pages 12–21, Calgary, Alberta, Canada, 2006. IASTED.

[18] I. Korhonen, J. Pärkkä, and M. van Gils. Health monitoring in the home of the future. *IEEE Engineering in Medicine and Biology Magazine*, 22(3):66–73, 2003.

[19] J. Kukkonen, E. Lagerspetz, and P. Nurmi. *BeTelGeuse website*, Jan. 2007. Available: `http://www.cs.helsinki.fi/group/acs/betelgeuse/` [2007-01-07].

[20] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. In *Proc. 4th International Conference on Pervasive Computing (PERVASIVE, Dublin, Ireland, May 2006)*, volume 3968 of *Lecture Notes in Computer Science*, pages 1–16, Berlin, Germany, 2006. Springer-Verlag.

[21] Motorola, Austin, TX, USA. *Java APIs for Bluetooth Wireless Technology (JSR 82), Specification Version 1.1*, Sept. 2005. Available: `http://jcp.org/aboutJava/communityprocess/mrel/jsr082/` [2007-01-15].

[22] P. Nurmi and J. Koolwaaij. Identifying meaningful locations. In *Proc. 3rd Annual International Conference on Mobile and Ubiquitous Computing (Mobiquitous, Sun Jose, CA, USA, July 2006).* IEEE Computer Society, 2006.

[23] P. Nurmi, A. Salden, S. L. Lau, J. Suomela, M. Sutterer, J. Millerat, M. Martin, E. Lagerspetz, and R. Poortinga. A system for context-dependent user modeling. In *Proc. OTM Federated Workshops (Montpellier, France, October–November 2006)*, volume 4278 of *Lecture Notes in Computer Science*, pages 1894–1903, Berlin, Germany, 2006. Springer-Verlag.

[24] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. ContextPhone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2):51–59, 2005.

[25] D. Salber, A. K. Dey, and G. D. Abowd. The Context Toolkit: Aiding the development of context-enabled applications. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI, Pittsburgh, PA, USA, May, 1999)*, pages 434–441, New York, NY, USA, 1999. ACM Press.

[26] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong. SenSay: A context-aware mobile phone. In *Proc. 7th IEEE International Symposium on Wearable Computers (ISWC, White Plains, NY, USA, October 2003)*, pages 248–249, Washington, DC, USA, 2003. IEEE Computer Society.

[27] T. Sohn, W. G. Griswold, J. Scott, A. LaMarca, Y. Chawathe, I. Smith, and M. Y. Chen. Experiences with Place Lab: An open source toolkit for location-aware computing. In *Proc. 28th International Conference on Software Engineering (ICSE, Shanghai, China, May 2006)*, pages 462–471, New York, NY, USA, 2006. ACM Press.

[28] Sun Microsystems, Inc., Santa Clara, CA, USA. *Connected Limited Device Configuration Specification, Version 1.1*, Mar. 2003. Available: `http://jcp.org/aboutJava/communityprocess/final/jsr139/` [2007-01-15].

[29] A. Teller and J. Stivoric. The BodyMedia platform: continuous body intelligence. In *Proc. 1st ACM workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE, New York, NY, USA, October 2004)*, pages 114–115, New York, NY, USA, 2004. ACM Press.

[30] A. Varshavsky, M. Y. Chen, E. de Lara, J. Froehlich, D. Haehnel, J. Hightower, A. LaMarca, F. Potter, T. Sohn, K. Tang, and I. Smith. Are GSM phones THE solution for localization? In *Proc. 7th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA, Semiahmoo Resort, WA, USA, April 2006)*, pages 20–28, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[31] T. Yamabe, A. Takagi, and T. Nakajima. Citron: A context information acquisition framework for personal devices. In *Proc. 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA, Hong Kong, China, August 2005)*, pages 489–495, Washington, DC, USA, 2005. IEEE Computer Society.