Bluetooth Interaction Management in Smart Spaces

MIGUEL M. ALMEIDA
Universidade do Minho, Braga

Many applications available on the market, as well as projects related to smart spaces, are using Bluetooth as a way of interacting with users. However, their ad-hoc implementation come as a limitation to their deployment in a global interactive model. This survey focuses on the requirements of such applications and aims to be the first step in the creation of a Bluetooth hotspot - an open source software for situated interaction management, running on Linux routers.

Categories and Subject Descriptors: H.5.2 [**User Interfaces**]: Interaction Styles; I.3.6 [**Methodology and Techniques**]: Interaction Techniques

General Terms: Design, Management, Standardization

Additional Key Words and Phrases: smart spaces, bluetooth interaction, integration models, urban computing, ubiquitous computing, middleware infrastructures

1. INTRODUCTION

1.1 Motivation

Handheld devices actually take part of our lives. As Bluetooth is present in most of these devices and is an easy to use, short-range and no cost technology, it becomes a nice way of interaction with other applications available on the surrounding space. A lot of existing projects are taking advantage of such characteristics to interact with users. Despite many needs of these projects being similar, each one implements its own Bluetooth interface, which is deployed only for that purpose. Why not have a shared component, capable of serving multiple applications at the same time? This empowers and encourages the development of new applications based on the same situated interactions, freeing the developer from Bluetooth technical details and enabling a more sustainable development of the desired applications.

1.2 Objectives

The objective of this study is to justify the need of this component on the context of smart spaces, urban computing or similar environments. In this survey, the common characteristics of the existing systems/applications that uses Bluetooth situated interactions are identified (as well as some needs for future related projects). Based on this, a set of common requirements is presented, becoming the first approach to the description of the architecture of a Bluetooth hotspot - a Linux router running a software dedicated to the interaction with users and offering an API to the interested applications. The choice of this kind of devices is because of its ease of deployment and its low price and size. A simple Linux enabled router equipped with an USB port can be easily turned into a Bluetooth hotspot just by plugging an USB Bluetooth dongle and installing the software.

Miguel Craveiro Martins de Almeida, Departamento de Sistemas de Informação
Universidade do Minho Master Course on informatics - State of the Art Reports 2010

2. RELATED WORK

Despite any wide definition, we can define a "smart space" as a space where users can ubiquitously access services or interact with the environment, on one or multiple contexts [McCarthy et al. 2008]. Middleware infrastructures for smart spaces can be seen like an operating system where all the hardware and software components can fluently live and work together. After analysing the existent related projects (or somehow presenting some kind of interaction between the user and the environment) we mainly identify three different ways to share information between the system and its users: using the device name, sending or receiving a file, or establishing a permanent connection (e.g. socket) to share more complex data. Here we analyse each one of them, in the context of those existing applications.

2.1 Device Name

The simplest kind of interaction with an application available on a smart space is the use of the Bluetooth's device name, using a predefined well-known syntax. With this method, the user does not need to install any kind of application, turning on a easy and cross-platfrom method. Naturally, as the device name space is short, abbreviation becomes necessary [Davies et al. 2009]. In the context of smart spaces, the Bluetooth Extended Naming[1] is a good choice for tagging personal information or interests. It allows users to share small personal information like his name or mood, his Facebook[2] profile, tagging words to share interests about specific news (e.g: `tag.radiohead`) and even voting (e.g: `vote.braga`). Projects like the one described below are using Bluetooth scannings to get the names of nearby devices. All of them refer a "Bluetooth scanner" that can store the sightings (the list of discovered devices) or send them to a central server.

BluetunA [Stephan Baumann 2007] purposes a system for people to share their interests just using the Bluetooth name of their devices. On a public space with environment music playing, the present users can suggest artists or musics to be played just by changing the Bluetooth name of their devices to specific tags. *If there are more than one user in the Bluetooth proximity area, the system identifies the preferences shared by most of the users.* Other services can be based on such kind of information, like public displays showing news and advertisements or adjusting the temperature in the office to the user preference. With this method of using the device name, neither the user needs to be previously registered on the system nor need to install any kind of software on his device.

BluetoothTracking.org is a project for collecting Bluetooth sightings from geographically distributed measurement points. *The challenge of this project is the amount of data collected*, that can be later used for statistical or investigation ends. Embed devices with a Bluetooth scanner software developed for this purpose are being used, software which sends the sightings to a central server, to become available on the web.

---

[1] *Bluetooth Extended Naming is a technique where the Bluetooth device name is used for interaction purposes. The main concept is to use Bluetooth name as command line interface with a command language, a collection of natural-language-derived commands, built with the purpose of enabling simple interaction with near systems*, sit [2009b]

[2] `http://www.facebook.com/`

Bluetooth can also be useful in other completely different contexts, like vehicle traffic monitoring. One in twenty vehicles have at least one Bluetooth-enabled device inside it [Young 2008]: a cell phone, a GPS receiver, an headset or even the car itself. With multiple Bluetooth scanners distributed along a high-way or road (installed from 2 to 4 miles of each other) it becomes possible to calculate the average speed of a car on that road section, if it is observed by two scanners. The Center for Advanced Transportation Technology of University of Maryland developed a box that can be deployed on the street, that periodically searches for devices around it and stores the sightings internally, to be afterward downloaded and analyzed. The Traffax Inc.[3] is an enterprise that has an identical commercial product called BluFax.

On a public space - like a bar or hotel hall - it is possible to deploy a public display containing news, weather, adverts, quizzes/pools, photos or other kind of useful information. However, it becomes very interesting to choose or orient the context based on the user's preferences or wishes. The Instant Places is a system that explores how Bluetooth can be used for situated interaction in public displays [José et al. 2008]. First, the system shows the names of the present Bluetooth devices on the screen, which encourages the users to explicitly interact with the screen. With a specific syntax for the device name, they can share simple words like tags for the news selection (e.g. changing his name to "Ericsson tag:radiohead") or personalise their avatar with a photo of Flickr (e.g. using the name "John flk:johnsmith"). The article talks about a *"Bluetooth scanner [that] periodically collects information about nearby devices, which a situation data model then consumes"*. This scanner is a dumb component that only obeys orders from a central server, responsible for maintaining the state and all the context.

2.2 Send/receive Files Using OBEX

Probably the most common and useful kind of interaction for the user, is the ability of sharing files with "something" in the environment (either send or receive). This can be done in both ways: to the user (e.g. for Bluetooth marketing purposes) or from the user (e.g. to display a PowerPoint^{TM}presentation on a public display).

BlueMall is a Bluetooth-based advertisement system for marketing purposes on large commercial areas, *delivering information based on the clients' current location*, Sánchez et al. [2008]. The article describes the system architecture based on three software entities: client mobile devices, BlueMall access-points[4] and a central server. This server is responsible for managing the access-points (AP) behaviours as well as maintaining the state and context. The server configures the access-points using a XML file (with variables like the AP location, time elapsed to consider a client's visit like a new visit, server address and the amount of time to ignore a device until considered as a new visit). To avoid flooding the users with files (which can be very negative if it leads the user to turn off his device) BlueMall takes care

[3] `http://www.traffaxinc.com/`

[4] Beyond the BlueMall access-points, the article [Sánchez et al. 2008] also refers the possibility of using a Bluegiga [sit 2009a] Bluetooth acces-point - a commercial device capable of sending and receiving files by the Object Exchange protocol (OBEX Object Push service) and sharing an Internet connection to the user's Bluetooth devices.

of "what" information is sent to "whom" and "when". Each time a device is seen (identified by his MAC address) the server's database is updated, also registering if the OBEX Object Push service is available. Knowing such information speeds up further scans since there is no need to ask the device for the service availability again. The system also supports a white list of devices not interested in receiving files, intended to be filled with the MAC addresses of the employees. *Experimental results show that this system provides a viable solution for permission-based mobile advertising.* Commercial products like BlueMarket, BlueMagnet and BlueSender implements systems like BlueMall, using a standalone computer.

2.3 Establishing a permanent connection

LectComm is an open-source software that can be used to support the lector during his classes. Running a software on the student's computers (or accessing a web site), they may answer questions and quizzes made by the professor, who uses the server version of this software. As the students needed to access the Internet to use the application (and only 4 of 72 were willing to [Bär et al. 2007]), an extended version of the client software application supporting Bluetooth was later developed. The article also talks about a generic access-point, which maintains a connection with the clients[5].

Home automation systems available on the market are using protocols like X10 protocol[6] to transmit signals between the system and the controllers (e.g. lights, doors, windows and temperature or light sensors) using the power line. The article "Design and Implementation of a Bluetooth based General Purpose Controlling Module" [S.P. et al. 2008] describes how Bluetooth can be used in the domotic context, replacing the power line with wireless connections. The proposed system consists on a main application which controls the multiple sensors and actuators spreaded around the house.

Nintendo's Wii remote controller (Wiimote) is equipped with Bluetooth, which makes it possible to use it as a way of interaction with a smart space, for example in a museum where users aim the Wiimote to a piece of art and an audio record about it starts playing or a display shows information. *An accelerometer inside the Wiimote measures force values acting on the different axes and in turn, these values provide the basis for computer gesture input and recognition*, Fitz-Walter et al. [2008]. After being paired with a system, the Wiimote can send information about its movement on the X, Y and Z coordinates, using the accelerometer. Using an infrared (IR) receptor and two IR emitters in front of the user, the Wiimote can also known his location in the museum's space[Fitz-Walter et al. 2008] [7]. The remote control also informs the remote system when each of the 12 buttons available is pressed, but can also receive orders, like starting or stopping its internal vibrator.

[5]Despite LectComm works over an RFCOMM connection, a simplest solution could be just using the device name to answer to the questions and quizzes proposed by the lector [Davies et al. 2009].
[6]X10 is a communication standard developed by Pico Electronics in 1975, but still dominant on the market, that uses the power line to transmit signals.
[7]Using an infrared bar (or multiple bars [Sko and y Gardner 2009]) it is possible to improve the movement detection.

3. FUTURE CHALLENGES

Almost all of the described systems assume the existence of an independent component that acts as a Bluetooth bridge with the user. Usually this component is just a single part of the deployed application, but when we think in a ubiquity way it makes sense to have a piece of hardware that is only focused on Bluetooth generic interaction tasks. It will act as a context/situation middleware component, and it must be capable of serving multiple applications at the same time. We will call it a "Bluetooth hotspot".

3.1 Context management

The hotspot is an autonomous component of the system, but probably it will never be deployed alone without a central server. The server must configure the hotspot to act like it should but must also receive data from it to dynamically maintain the context. Imagine a situation where BlueMall successfully sends a file to a user: the server must log this event and reconfigure the router not to send the file again. This kind of "conversation" between the hotspot and the system immediately raises a problem: on situations like this one, if the server is not reachable, the hotspot probably needs to suspend all its automated tasks to avoid decontextualization.

Some scenarios require multiple hotspots to cover all the user space. Although, if we want to maintain a context and a state of the situation, we cannot simply deploy more than one without some integration mechanism. Imagine a user that enters a museum, hotspot A detects him and successfully send him a file. Then, the user moves across the museum and hotspot B also detects his presence. Despite the file not been sent to the user through this hotspot, it does not make sense to send it again because the user already have it. The server must be able to deal with this situation.

In some situations it may not be necessary to inform the server about all the events, giving the hotspot the ability to decide and make "intelligent decisions". For example, a space where the user handles a Wiimote to interact with a display, but we only want certain actions to be triggered if a specific gesture is detected (imagine shaking from left to right). It does not make sense to inform the server about all the movements made by the user. It should be possible to somehow describe this gesture and the triggered action.

After researching and analysing the described characteristics, we are leaded to the idea that an hotspot mainly needs to deal with two things: **events** and **actions**. Even a more complex task - like interpreting Wiimote gestures or establishing a PAN connection[8] - is triggered by an event.

3.2 The communication between the hotspot and the system

The functional logic described above fits on three kinds of server vs. hotspot communications:

Sightings. This is the list of seen devices on each Bluetooth scan. This list must be sent to the server, allowing the interested applications (like BluetunA,

[8] A Bluetooth PAN connection is a TCP/IP connection between two Bluetooth devices, commonly used to share an existing Internet connection with the connected devices.

InstantPlaces or BluetoothTracking.org) to consume this data.

Internal configurations. The list of variables and values needed to configure the hotspot (e.g: scan interval and server's URI).

Behaviour rules. As described above, a list of events and actions should be adequate to describe hotspot behaviour, each one identified by a unique string.

We do not want to have a permanent socket connection opened between the hotspot and the server, but think on a RESTful way (Representational State Transfer). This can be peaceful for cases like the "sightings" list being sent to the server, but can be a problem for the "behaviour rules" list as it can be quickly changing or near real-time, on some situations. Imagine the BluetoonA scenario, where the hotspot sends a file to a user: after being successfully sent, the server needs to inform the hotspot (or a group of them, if there are more than one) that it should not send any file to that device again. But how, if the client is a client and not reachable? On the other hand, we cannot have an intensive pooling from the hotspot to the server, asking for new rules. All this situations needs to be balanced in order to find a solution capable of serving all the possible scenarios.

3.3 Modular system

It is not possible to foresee all the possible scenarios and design a perfect system, so a modular architecture is a must. Each module is responsible for its tasks but capable of comunicating with all the others, as they probably need to be coordinated. So, despite the core system, we identify the next modules for the studied systems and applications: the "configuration and rules manager", a "dispatcher that analyses the rules and triggers the actions", the "device scanner", the "OBEX manager", the "RFCOMM manager" and the "wiimote gesture identifier".

3.4 Fundamental implementation issues

There are some unavoidable problems that can occur during the deployment. If analysed during the development, they can be minimized. Probably the hardest to deal with is the intersection of our system with others existing in the neighbourhood: When the hotspot tries to pair with a new device seen we cannot make sure that the user is interested in the connection of that spot.

Another problem inherent to the usage of the device name is the limited space available for it. Despite of the Bluetooth specification limit of 248 bytes with `UTF-8` encoding for the device name [Bluetooth Special Interest Group 2004], most of the brands strict to his (very low) own limit - sometimes above 32 characters. This makes abbreviation and tags an obligation (see [Davies et al. 2009] and [sit 2009b]).

The device scanning and the device name retrieval can introduce unforeseen performance issues if not done with the appropriate duration [Peterson et al. 2006]. If scanning is made too many times in a given time period, the majority of these scans will not find a new device. However, if we take too much time to scan, the user can give up and leave the space. Furthermore, when retrieving the device name from the sighted devices, even the number of devices must be taken into account [Fresnedo et al. 2007]. If there are too many devices on the space, this process can take too long. So, probably scanning frequency configuration should not be completely flexible for the hotspot administrator.

4. CONCLUSIONS

This survey showed the state-of-the-art for the systems and applications based on Bluetooth situated interactions and leaded to a set of requisites and issues for the development of this middleware component. It also becomes evident that a software like this Bluetooth hotspot - easy deployable and easy to use - is a must on this area. As a system like this one will be implemented shortly, this became a very important study in order to fully understand the actual needs from the point of view of Bluetooth interaction, and also easing the design of a well-grounded architecture of this component.

REFERENCES

2009a. Bluegiga: Bluetooth access-points `http://www.bluegiga.com/`.

2009b. Bluetooth extended naming `http://ubicomp.algoritmi.uminho.pt/ben/`.

Bär, H., Häussge, G., and Rössling, G. 2007. An integrated system for interaction support in lectures. *ACM, SIGCSE conference on Innovation and technology in computer science education*.

Bluetooth Special Interest Group, B. S. 2004. Bluetooth core specification v2.0. technical report.

Davies, N., Friday, A., Newman, P., Rutlidge, S., and Storz, O. 2009. Using bluetooth device names to support interaction in smart environments. *ACM, International conference on mobile systems, applications, and services*.

Fitz-Walter, Z., Jones, S., and Tjondronegoro, D. 2008. Detecting gesture force peaks for intuitive interaction. *ACM, 5th Australasian Conference on Interactive Entertainment*.

Fresnedo, O., Iglesia, D. I., and Escudero, C. J. 2007. Bluetooth inquiry procedure: optimization and influence of the number of devices. In *CSN '07: Proceedings of the Sixth IASTED International Conference on Communication Systems and Networks*. ACTA Press, Anaheim, CA, USA, 205–209.

José, R., Otero, N., Izadi, S., and Harper, R. 2008. Instant places: Using bluetooth for situated interaction in public displays. *Pervasive Computing, IEEE*.

McCarthy, J. F., Congleton, B., and Harper, F. M. 2008. The context, content & community collage: Sharing personal digital media in the physical workplace. *ACM*.

Peterson, B. S., Baldwin, R. O., and Kharoufeh, J. P. 2006. Bluetooth inquiry time characterization and selection. *IEEE Transactions on Mobile Computing 5,* 9, 1173–1187.

Sánchez, J.-M., Cano, J.-C., Calafate, C. T., and Manzoni, P. 2008. Bluemall: A bluetooth-based advertisement system for commercial areas. *ACM*.

Sko, T. and y Gardner, H. 2009. The wiimote with multiple sensor bars: Creating an affordable, virtual reality controller. *ACM, International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*.

S.P., W., U.S., W., G.R.V., P., C.S, A., and A.T.L.K., S. 2008. Design and implementation of a bluetooth based general purpose controlling module. *IEEE, Information and Automation for Sustainability*.

Stephan Baumann, Björn Jung, A. B. M. W. 2007. Bluetuna: let your neighbour know what music you like. *ACM*.

Young, S. 2008. Bluetooth traffic monitoring technology. *Center for Advanced Transportation Technology, University of Maryland*.