Bluetooth Hotspots for Smart Spaces Interaction

Miguel M. Almeida, Helena Rodrigues, and Rui José

Departamento de Sistemas de Informação
Universidade do Minho

Abstract. We may find in the market many smart space applications and projects which are using Bluetooth services as mechanism for user-interaction. 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, systematizing their interaction models and presents the main key design of a middleware component for Bluetooth interaction-based pervasive applications, running on Linux routers.

1 Introduction

Bluetooth is present in most of the actual handheld devices. Being easy to use, a short-range and low cost technology, many applications are using it to interact with users. Despite many needs of these projects being similar, each one implements its own Bluetooth interface, deployed only for that purpose. So, why not have a shared component, capable of serving multiple applications simultaneously? This could empower and encourage 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. The objective of this study is to justify the need of such component - a software running on Linux-enabled routers, offering Bluetooth services to applications. In this paper we survey a set of Bluetooth interaction-based projects leading to a set of reference scenarios, which we use as a starting point for the discussion of the main key design issues of a Bluetooth Hotspot, a middleware component for Bluetooth interaction-based pervasive applications.

2 Related Work

Our analysis of the related work is based on the study of representative systems based on most common Bluetooth services: device and service discovery, Bluetooth OBEX and RFCOMM connection. On the whole set, we mainly identify three different ways

INForum 2010 - II Simpósio de Informática, Luís S. Barbosa, Miguel P. Correia (eds), 9-10 Setembro, 2010, pp. 439–442

to share information between the system and its users: 1) using the device name, 2) sending or receiving a file, or 3) establishing a permanent connection (e.g. socket) to share more complex data.

The simplest form of interacting with an application is using Bluetooth's device name following a predefined syntax. With this method, the user does not need to install a specific application, promoting a cross-platform method. The BluetunA project [1] proposes a system for people to share their music interests using the Bluetooth name on their devices. Users on public spaces may suggest music and artists to be played just changing their device names to specific tags. Bluetooth device scanning based systems also include car traffic monitoring. For example, in [2], Bluetooth scanners, geographically distributed, collect device's addresses and names, which are then analysed for statistical purposes.

One of the most common Bluetooth interactions are based on the Bluetooth OBEX profile for sending and receiving files. The interaction is two-away, from the user to the system (e.g. to display a PowerPoint^{TM}presentation on a public display) and from the system to the user (e.g. for Bluetooth marketing purposes). The BlueMall system [3] is an example of a Bluetooth-based advertisement system for marketing purposes: users receive marketing information as they enter a specific place.

Other works use the RFCOMM protocol for application specific communications. LectComm, an open-source software, is an application for smart classrooms. Students, equipped with LectComm client applications send their answers to questions and quizzes to the LectComm server application, running on lecturer's computers. Nintendo's Wimote communication is also based on this type of connections, allowing applications to recognize user's gestures [4] as a way of interaction.

3 Discussion

We propose to build the Bluetooth Hotspot as a middleware component for the development of Bluetooth interaction-based applications. This component should provide the most common Bluetooth services and serve the representative scenarios surveyed in the last section. We now briefly discuss the key design issues of such a component.

The first design issue concerns the integration models with applications which presents two main requirements: (1) how to configure and request the Hotspot services and (2) how to exchange information (Bluetooth sights and files) with applications.

The second key design issue concerns the global state management of the system. This issue is raised because (1) Hotspots are shared between different applications, and thus configuration conflicts may appear, and (2) some scenarios require multiple

Hotspots to cover all the user space in the context of the same application, which may raise global state inconsistency problems.

Finally, considering the existence of additionally Bluetooth services and future application-dependent requirements (for example, Wiimote gesture recognition), the Hotspot architecture should be modular, providing the easy integration of third-party extensions.

4 System Architecture

Our global architecture comprises a central server capable of configuring and managing a set of Hotspots components and the Hotspot itself. The central server is the main point for integrating applications with the Hotspot and for global state managing. The main function of this server is to receive interaction requirements from applications and provide to the Hotspot a set of rules, which model the Hotspot behaviour. This rules describe parameters for scanning intervals, OBEX delivery and OBEX receiver instructions and instructions concerning data exchanging with applications. The Hotspot architecture is depicted in figure 1. It comprises five essential components:

1) the **Bluetooth Modules**, responsible for managing basic Bluetooth functions like device discovering, scanning, and receiving and sending files using OBEX. These modules are initiated by the **Scheduler** module. 2) the **Internal Managers** are responsible for managing the Bluetooth radio interfaces (dongle manager) and the interactions between the Bluetooth component and the applications (rules manager and resources manager). The **rules manager** is responsible for receiving applications configuration rules (possibly from a configuration web site). The **resources manager** is responsible for the communication between the Bluetooth component and the applications. 3) the **Events Channel** implements a decoupled communication model between system components. 4) the **Scheduler** interprets the rules and triggers the correct action for each new event that enters the events channel. 5) a set of third-party **Extensions**, installed by the Hotspot administrator, allowing to extend the system with specific functionality.

5 Conclusions and Future Work

We have presented a short survey on representative Bluetooth-interaction based applications, which are based on the most common Bluetooth services: device and service discovery, Bluetooth OBEX and RFCOMM connection. We support on this study the specification, design and implementation of a Bluetooth Hotspot software component and the corresponding integration models with applications and present our work in progress.

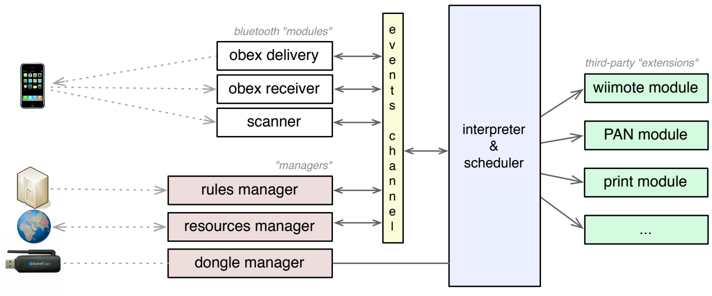

Fig. 1. Proposed component architecture for the Hotspot software.

We have developed a few Hotspot prototypes, which meet partially our architecture. They implement some basic modules such as the scanning, OBEX receiver and OBEX delivery modules. Those Hotspot prototypes are being currently integrated, deployed and shared in the context of different systems.

We are now finishing the implementation of the remaining architecture modules and proceeding to the evaluation of the different deployed systems. We expect to learn from those deployments the necessary lessons in order to validate our main key design issues of global state management, application integration and system modularity. Finally, we expect to evaluate the Hotspot software component in the context of the broader research area of system support for ubiquitous computing.

Given the open nature of the Hotspot component and the natural evolution of Bluetooth interactions, we aim to initiate a process for building the software as an open-source software. We expect to encourage the development of new third-party extensions as well as promote the utilization of Bluetooth as an interaction device in smart spaces.

References

1. Arianna Bassoli Martin Wisniowski Stephan Baumann, Björn Jung. Bluetuna: let your neighbour know what music you like. *ACM*, 2007.
2. Stan Young. Bluetooth traffic monitoring technology. *Center for Advanced Transportation Technology, University of Maryland*, September 2008.
3. José-María Sánchez, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. Bluemall: A bluetooth-based advertisement system for commercial areas. *ACM*, 2008.
4. Zachary Fitz-Walter, Samuel Jones, and Dian Tjondronegoro. Detecting gesture force peaks for intuitive interaction. *ACM, 5th Australasian Conference on Interactive Entertainment*, 2008.