

Patterns in Bluetooth

Aniruddha Gokhale
Bell Laboratories
Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ 07974
agokhale@lucent.com

Abstract

Bluetooth is a de facto standard for low-cost, short range radio links between mobile phones, mobile PCs and other mobile devices.

This paper identifies several patterns used by Bluetooth.

1 Overview of the Bluetooth Standard

The Bluetooth [1] specification is a *de facto* wireless communications standard formed by a consortium of telecommunications and data networking companies for short-range, radio communication between portable devices such as mobile phones and handheld devices. The goal of the standard is to eliminate the need for any cable connectivity and promote full-scale *ad hoc* networking. The key features include robustness, low complexity, low power, and low cost.

Bluetooth operates in the unlicensed ISM band at 2.4 GHz. The protocol uses a combination of circuit switching and packet switching thereby supporting both synchronous voice channels and asynchronous data channels.

The primary components of Bluetooth include a radio unit, a link control unit, and a support unit for link management and host-terminal interface.

2 Patterns in Bluetooth

This section illustrates several patterns implemented by Bluetooth. For each pattern we describe the pattern's intent, how it is used in Bluetooth, and its known uses.

2.1 Broker pattern

Intent: The *Broker* [2] pattern is used to structure distributed systems with decoupled components that interact by remote method invocations.

Bluetooth Usage: Bluetooth defines an interoperability protocol called IrOBEX. The IrOBEX protocol provides a model for representing objects and a session protocol. This protocol structures a dialog between two remote devices that follow a client/server *request-response* paradigm.

Known Uses: The Object Management Group's Common Object Request Broker Architecture (CORBA) [3], Microsoft's Distributed COM (DCOM) [4], and Sun Microsystems's Java Remote Method Invocation (RMI) [5] are the dominant technologies for distributed computing that implement the Broker pattern.

2.2 Layers pattern

Intent: The *Layers* pattern helps to structure applications that can be decomposed into groups of subtasks in which each group of subtasks is at a particular level of abstraction.

Bluetooth Usage: The Bluetooth specification defines a protocol stack. At the bottommost layer is the physical link (Baseband Layer) comprising the radio frequency channel. Above this is the Link Manager Protocol (LMP) layer used for link setup and control. The data link layer above LMP is called the Logical Link Control and Adaptation Protocol (L2CAP) whose objective is to support higher level protocol multiplexing, packet segmentation and reassembly, and relaying quality of service parameters. Above the L2CAP layer are higher layer protocols that get multiplexed into L2CAP. These include the Service Discovery Protocol (SDP) which does service lookup, the RFCOMM protocol that provides a serial port emulation, and the Telephony Control Protocol (TCS) for establishing speech and data calls between Bluetooth devices.

Known Uses: The OSI [6] protocol stack is the most famous layered architecture that uses the Layer pattern.

2.3 Lookup pattern

Intent: The *Lookup* [7] pattern describes how to find and retrieve initial references to distributed objects and services.

Bluetooth Usage: Bluetooth's Service Discovery Protocol (SDP) uses the Lookup pattern for locating services provided by or available through a Bluetooth device. In addition, SDP also allows applications to determine the characteristics of the services.

Known Uses: OMG CORBA's Naming Service and Trading Service, and Sun Microsystem's Jini Lookup service implement the Lookup pattern.

2.4 Bridge

Intent: The *Bridge* [8] pattern decouples an abstraction from its implementation so that the two can vary independently.

Bluetooth Usage: Bluetooth uses PPP as a communications bearer to provide communication between WAP (Wireless Access Protocol) clients and servers. In this way Bluetooth implementers can provide access to value-added services using the WAP environment.

Known Uses: [8] mentions the Bridge pattern being used in ET++, libg++, and NeXT's AppKit.

2.5 Facade

Intent: The *Facade* [8] pattern provides a unified interface to a set of interfaces in a subsystem. Facade defines a higher layer interface to make it easier to use the subsystems.

Bluetooth Usage: Bluetooth's IrOBEX protocol provides a unified interface to the Bluetooth's protocol stack as well as the IrDA (Infrared Data Association) protocol stack. Bluetooth's RFCOMM uses the Facade pattern to provide a unified interface to different types of serial ports. Bluetooth's host controller interface (HCI) provides a uniform mechanism to access Bluetooth's underlying baseband capabilities.

Known Uses: The ADAPTIVE Communications Environment (ACE) [9] uses the Facade pattern to provide a unified interface to several operating systems.

2.6 Service Browser pattern

Intent: The service browser pattern allows browsing all services offered by a lookup service.

Bluetooth Usage: Bluetooth's Service Discovery Protocol (SDP) allows applications to browse all the services offered by a SDP lookup service.

Known Uses: The Jini lookup service offers a mechanism whereby applications can browse all the services registered with it.

2.7 Miscellaneous Patterns used in Bluetooth

Bluetooth's host controller interface implements the *Command* and *Interpreter* patterns by providing a command interface to the underlying baseband and link controller.

The L2CAP protocol layer implements a *Dispatcher* pattern that demultiplexes and dispatches packets to higher layer protocols.

The *Master-Slave* and *On-demand Activation* patterns are used in Bluetooth in the Bluetooth device test mode. In this mode, a test controller acts as the master and the device under test (DUT) acts as a slave. The controller activates the DUT whenever it is to be tested. The Master-Slave pattern is also implemented in Bluetooth's baseband specification when a point to multipoint connection is provided by the Bluetooth system. In such a scenario, the channel is shared by several Bluetooth devices. One of the devices is the master and the others are slaves.

3 Concluding Remarks

Bluetooth is a de facto standard for short range wireless communication between mobile devices. Although Bluetooth differs from traditional *ad hoc* networks, its design is based on well-known patterns. This paper describes patterns used by the Bluetooth *ad hoc* networking technology.

References

- [1] Bluetooth Consortium, "Bluetooth Wireless Technology Specification v1.0 Core," http://www.bluetooth.com/developer/specification/core_10_b.pdf, 1999.
- [2] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal, *Pattern-Oriented Software Architecture - A System of Patterns*, Wiley and Sons, 1996.
- [3] Object Management Group, *The Common Object Request Broker: Architecture and Specification*, 2.3 edition, June 1999.
- [4] Microsoft Corporation, *Distributed Component Object Model Protocol (DCOM)*, 1.0 edition, Jan. 1998.
- [5] Sun Microsystems, Inc., *Java Remote Method Invocation Specification (RMI)*, Oct. 1998.
- [6] U. Black, *OSI: A Model for Computer Communications Standards*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [7] Michael Kircher and Prashant Jain, "Lookup Pattern," in *Submitted to EuroPLoP 2000 conference*, Irsee, Germany, July 2000.
- [8] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA, 1995.
- [9] Douglas C. Schmidt, "The ADAPTIVE Communication Environment: An Object-Oriented Network Programming Toolkit for Developing Communication Software," in *Proceedings of the 12th Annual Sun Users Group Conference*, San Jose, CA, Dec. 1993, SUG, pp. 214-225.