Bluetooth Host-side Protocol Stack Development using Formal Design Techniques

Anne Margrette Q. Caccam, Myra Colina B. Dideles,
Bienvenido H. Galang, Jr. and Ian C.Wong
Advanced Science and Technology Institute
UP Technopark, Diliman, Quezon City
Email: bluetooth@asti.dost.gov.ph

***Abstract*—Bluetooth is a wireless technology standard for the interconnection of electronic devices in the personal space. It started out as a cable replacement technology and became a wireless personal networking (WPAN) solution. This paper describes the development of a Bluetooth Host-side Protocol Stack using Object-oriented and Formal Design Techniques such as SDL, ASN.1 and MSC. This marks the initial effort of the Advanced Science and Technology Institute[1] in developing a complete System-on-a-Chip (SOC) solution within five years time.**

***Index Terms*—Bluetooth, WPAN, Communications Protocol Software, SDL**

I. Introduction

Bluetooth is a wireless personal area networking (WPAN) standard intended to interconnect diverse electronic devices in short range[2] using a ubiquitous wireless link at the 2.4 GHz ISM Band. Bluetooth uses a frequency hopping spread spectrum radio signal to communicate with devices within a piconet[3].

Bluetooth is intended to be a single-chip, low-power, wireless communication module that will fit inside mobile phones, laptops, personal digital assistants, digital cameras and other mobile and stationary devices. It allows users to easily interconnect these devices and communicate seamlessly and effortlessly with each other. The simplicity of the wireless solution and its technical soundness has made Bluetooth the fastest growing technology since the Internet or the cellular phone. Various market projections have actually placed Bluetooth as one of the more profitable technologies in the near future.

A complete Bluetooth solution is realized using hardware and software. Bluetooth hardware consists mainly of the radio transceiver and the baseband link controller. Bluetooth software on the other hand, consists of the application software, protocol stack, and firmware. The firmware usually comprises of a portion of the Link Controller, Link Manager, and Host Controller Interface.

The Bluetooth Protocol stack determines how a device can connect to another device via Bluetooth technology. There are three approaches in the implementation of the Bluetooth Protocol Stack as shown in Fig. 1 [1]. The first approach is the standard two-processor architecture, which uses a host processor and a target processor. The two processors communicate through the Host Control Interface (HCI). This approach is used for PC applications. The second approach is the embedded two-processor architecture, which also uses two processors but most of the layers are embedded to the target processor. This is used for devices with limited resources such as mobile phones. The last approach is the wholly embedded single processor architecture, which is typical for a single chip solution.

The standard two-processor architecture was used in the development of the Bluetooth Host-side Protocol Stack. A host processor residing in a PC communicates with a target processor through the USB. The host side layers ported to the PC were modeled using SDL.

Specification and Description Language (SDL), was used to develop the communication protocol software. Using SDL to develop protocol software allows the developer to formally describe the functional aspects of the system. This in turn allows detection and remediation of functional errors early in the development cycle [2,3]. The SDL model is then used to generate C code in preparation for targeting to the specific operating system.

II. Bluetooth Protocol Architecture

The Bluetooth Protocol Architecture is shown in Fig. 2 [4]. The complete stack comprises of both Bluetooth-specific protocols like LMP and L2CAP and adopted protocols such as Object Exchange Protocol (OBEX) and WAP. In designing the protocols and the whole protocol stack, the main principle has been to maximize the re-use of existing protocols for different purposes at the higher layers, instead of re-inventing the wheel. The protocol re-use also helps to adapt existing applications to work with the Bluetooth technology and to ensure the smooth operation and interoperability of these applications. Thus, many applications already developed by vendors can take immediate advantage of hardware and

[1]DOST-ASTI is the Research and Development Institute of the Philippine government mandated to pursue R&D in the advanced fields of Microelectronics and Information and Communication Technologies.

[2]10-100 meters depending on the Class of the device, typically 10 meters

[3]A small network of devices connected in an adhoc fashion consisting of a master and at least one slave. The piconet can support up to seven slaves.

software systems, which are compliant to the Bluetooth Specification.

The Specification is also open, which makes it possible for vendors to freely implement their own or commonly used application protocols on top of the Bluetooth-specific protocols. Thus, the open Specification permits the development of a large number of new applications that take full advantage of the capabilities of the Bluetooth technology [5].

These characteristics of the Bluetooth protocol stack has caused a lot of software vendors to develop their own protocol stacks where application developers can license and easily embed Bluetooth functionality into their existing products and services. Most protocol stack vendors have their stacks available in C /C++ or Java code, targeted particularly to a number of operating systems. The advantage of developing in these languages is their portability to various platforms and operating systems. However, improvements and modifications to the Bluetooth specification happen very swiftly in industry standards. This puts a lot of pressure on software vendors to market their software, which is complies with the newest available standard. In this regard, the use of a higher-level language is very useful because modifications could be easily migrated to the entire protocol easily. SDL is such a language that actually specifies and describes the entire protocol stack hiding the implementation-specific details such that top-level modifications could easily be incorporated into the entire stack.

The host side of the protocol stack communicates with the target processor through the HCI as shown in Fig. 1. It converts the L2CAP packets into transport packets depending on the medium to be used, in this case USB. It also converts the USB packets back to L2CAP packets for the host processor. The Logical Link and Adaptation Protocol (L2CAP) is the layer above HCI. It is responsible for segmentation and reassembly. It provides channels for the upper layers such as RFCOMM and SDP. The Service Discovery Protocol (SDP) is a database of available services of the device. RF Communications (RFCOMM) emulates the RS-232 port for serial communications. It forms a virtual serial port with the port entity. The Stack Manager oversees smooth operation between layers. It also sets the security level of the device.

III. Stack Development using SDL

Formal Description Techniques (FDT) is very useful in specifying complex communication protocols. FDTs guarantee syntactically and semantically unambiguous formal descriptions of communication protocols. In addition, they also guarantee interoperable and compatible implementations of these protocols independent of their implementers. This removes a lot of the anxiety of current software vendors to make their stack interoperable with other stacks developed by other vendors. Furthermore, the conformance of these protocols to a given standard can be checked with the help of predefined tests [3].

SDL [6] is the most widely used FDT in the area of telecommunications. SDL supports object-oriented software design by dedicated elements of the language in contrast to other FDTs and its quick to learn graphical notation is self documenting, and thus, easily maintainable [7]. These are the reasons why SDL was chosen to specify the Bluetooth Protocol Stack.

In addition to the use of SDL, complementing languages like Message Sequence Charts (MSCs) [8] and Abstract Syntax Notation 1 (ASN.1) [9] were used. MSCs are widespread means for the description, and particularly graphical visualization, of selected traces within telecommunication systems. MSCs show sequences of messages interchanged between entities such as SDL services, processes and blocks and their environment. Since each sequentialization of a MSC describes a system trace, an MSC can be derived from an existing SDL system specification, and then serves as a statement of requirements for SDL specifications and a basis for selection and specification of test cases [6].

ASN.1, on the other hand, when used in combination with SDL, permits a coherent way of specifying the structure and behavior of communication systems, together with data, messages, and encoding of messages that these systems use. The structure and behavior are defined using SDL concepts, and the data and parameters of signals are defined using ASN.1 [10].

IV. Bluetooth Stack Specification using SDL

The specification of the Bluetooth Software stack using SDL follows the guidelines set forth in the SDL, MSC, and ASN.1 standards. The specific techniques for developing Open Systems Interconnect (OSI) protocols are defined clearly [11]. Each layer has a specific function comparable to the OSI reference model as shown in Fig. 3. The layers in the OSI model do not automatically map to each layer in the Bluetooth Protocol Stack.

The Bluetooth Host-side Protocol Stack consists of the core layers namely the HCI[4], L2CAP, SDP, and RFCOMM. The Stack Manager was included to handle the general stack management procedures and to ensure smooth operation of the protocol stack. Each layer was modeled as a block in SDL as shown in Fig. 4. Each block or layer communicates using signals passed through channels between blocks. The signals actually represent the service primitives each layer provides to the layer above it. The signals that extend to the environment are treated as Application Programming Interfaces (APIs), which the application developer can access to develop proprietary applications. These blocks are further refined to sub-blocks, processes, and procedures. ASN.1 was used to model the Protocol Data Units (PDUs) and MSCs were used to simulate the communication between the layers.

Each block consists of a client side and a server side. The client side of the block is responsible for initiating

[4] The HCI is currently being modeled in the second phase.

communication between blocks and the server side is responsible in giving the proper response. As shown in Fig. 4, two channels are provided for each block, one for signals initiated by the client side and another for the signals given by the server side. The signals received by each block are processed accordingly as stated in the Bluetooth specifications.

V. RESULTS

The Bluetooth Host-side Protocol Stack was modeled in SDL as a block type. Two instances of this type were put in a system to simulate two devices communicating with each other. The first instance represents the initiator and the second instance represents the responder. The initiator was able to create a connection with the responder. All the primitives needed to create a connection from L2CAP up to the upper layers (RFCOMM and SDP) were simulated. The environment simulated signals coming from the software application. It also simulated signals from HCI. A simple data transfer from an emulated serial port from the initiator to the emulated serial port of the responder was successful. The disconnection scenario including the release of the L2CAP channel was also successfully simulated.

VI. FUTURE EFFORTS

ASTI is now at the second phase in developing the Bluetooth Protocol Stack. The lower layers that will be embedded in a target processor are being modeled in SDL. The core layers are being refined and optimized. Modeling of the Generic Access Profile, Serial Port Profile, and Service Discovery Application Profile are also being initiated.

VII. CONCLUSION

Bluetooth is a fast emerging technology and has a very high market projection. Formal Design Techniques ease the development and maintenance of the Bluetooth Protocol Stack. Changes in the specification can easily be modified in the SDL model and verified through MSCs.

REFERENCES

[1] Mezoe. *Bluestack User's Manual*

[2] Andreas Mitschele-Thiel; *Performance Evaluation of SDL Systems*, Friedrich-Alexander-Universitat Erlangen-Nurnberg, Erlangen, Germany.

[3] Martin Steppler, *Performance Analysis of Communication Systems Formally Specified in SDL*, Aachen University of Technology, Aachen, Germany

[4] Jennifer Bray, Charles Sturman, *Connect Without Cables*, Prentice-Hall, NJ 1999

[5] Riku Mettala, *Bluetooth Protocol Architecture White Paper*, 1999 26th September 1997.

[6] Telecommunication Standardization Sector of ITU. *Specification and Description Language (SDL)*. ITU–T Recommendation Z.100, International Telecommunication Union, March 1993.

[7] Dieter Hogrefe, *Estelle, LOTOS and SDL—Standard-Spezifikationssprachen fur verteilte Systeme*. Springer Verlag, 1989.

[8] Telecommunication Standardization Sector of ITU. *Message Sequence Charts*. ITU–T Recommendation Z.120, International Telecommunication Union, 1993.

[9] Telecommunication Standardization Sector of ITU. *Abstract Syntax Notation.1*. ITU–T Recommendation X.680, International Telecommunication Union, 1993.

[10] Telecommunication Standardization Sector of ITU. *SDL Combined with ASN.1*. ITU–T Recommendation Z.105, International Telecommunication Union, 1993.

[11] Jan Ellsberger, Dieter Hogrefe, Amardeo Sarma. *SDL: Formal Object-Oriented Language for Communicating Systems.* Pearson Education Limited, Edinburgh Gate, Harlow

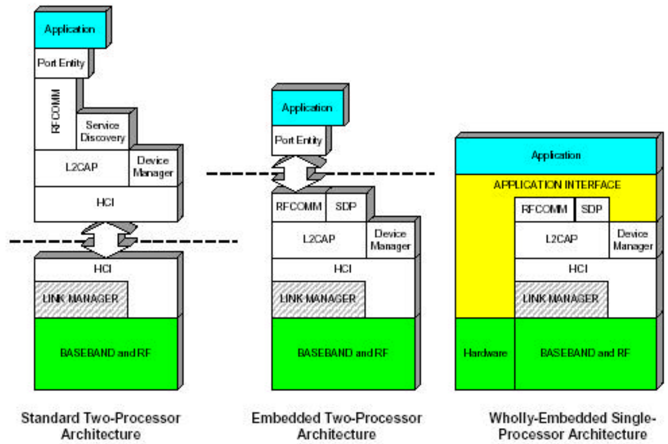

Figure 1. Possible Bluetooth Architectures

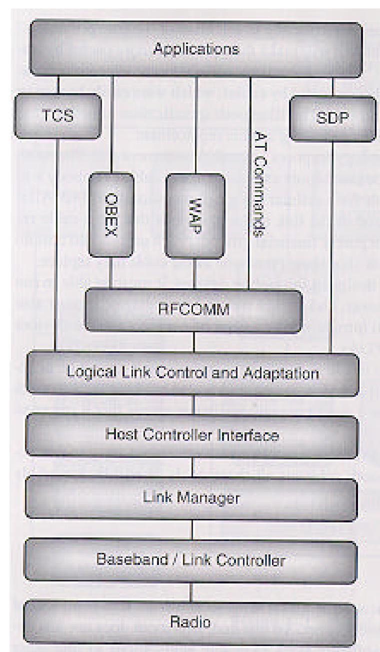

Figure 2. Bluetooth Protocol Stack

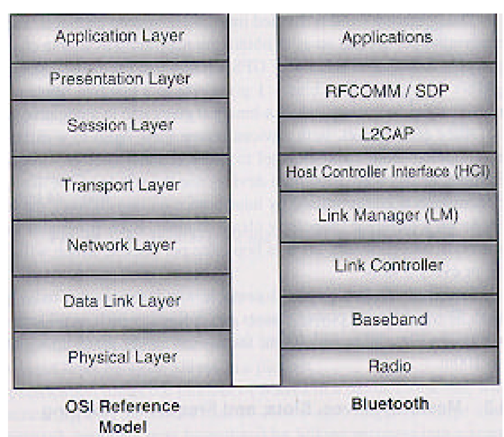

Figure 3. OSI Reference Model compared with Bluetooth Protocol Stack

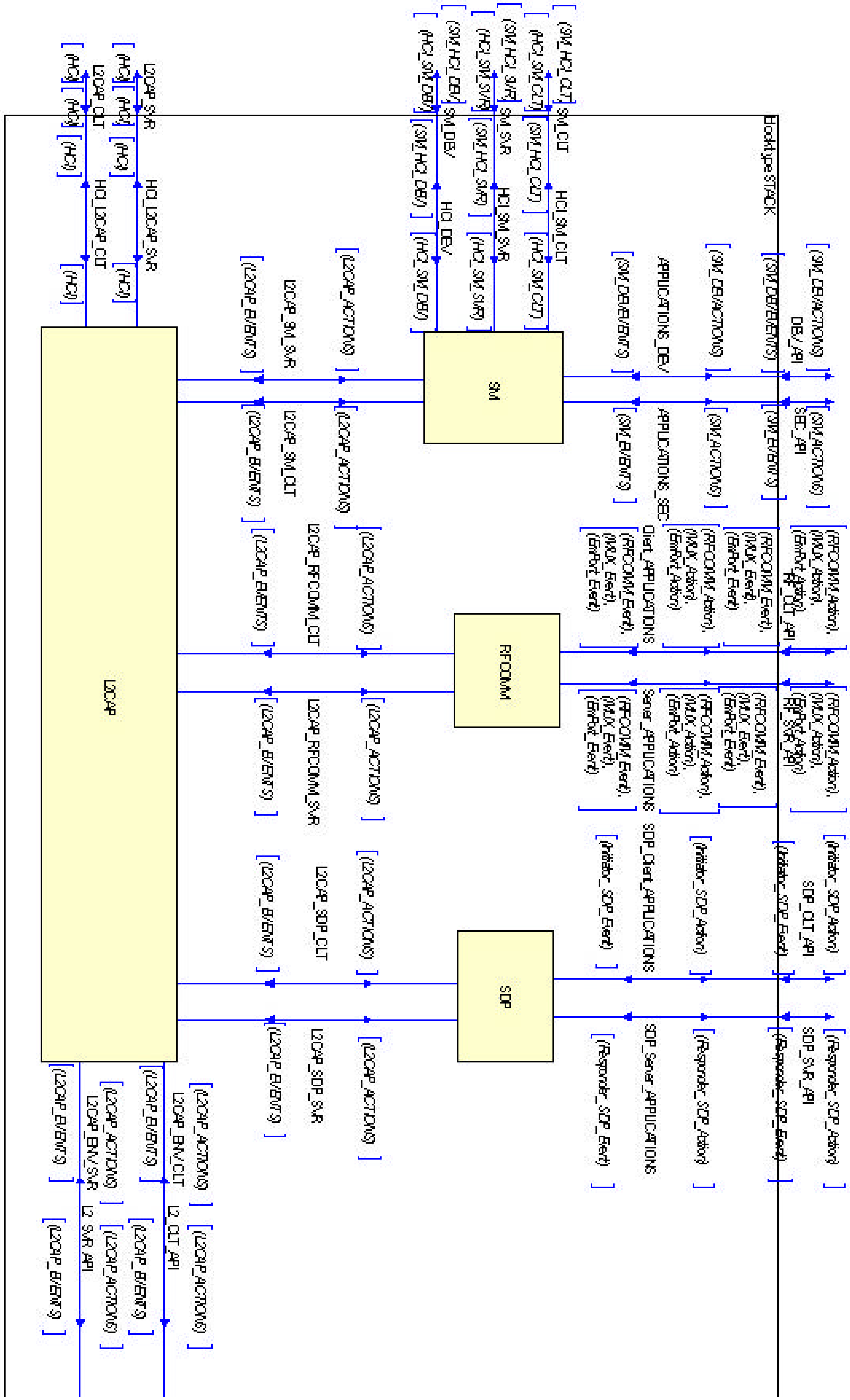

Figure 4. ASTI Bluetooth Protocol Stack using SDL, ASN.1, MSC