Evaluating Performance of a Bluetooth-Based Classroom Tool

Reggie Davidrajuh
University of Stavanger, Department of Electrical Engineering & Computer Science
PO Box 8002, 4036 Stavanger, Norway
Email: reggie.davidrajuh@uis.no

Biographical notes: Dr Reggie Davidrajuh received a Masters degree in Control Systems Engineering and a PhD in Industrial Engineering from Norwegian University of Science and Technology (NTNU) in 1994 and 2000, respectively. He is currently an Associate Professor of Computer Science in the Department of Electrical and Computer Engineering at the University of Stavanger, Norway. His current research interests include e-commerce, agile virtual enterprises, discrete event systems and modeling of distributed information systems.

ABSTRACT

This paper focuses on the performance evaluation of a Bluetooth wireless technology based classroom tool. The classroom tool is to help lecturers automate their assignment tests. First, a testing prototype of the tool is presented; the testing prototype uses one master device and two slave devices on Pico-net. As it will be expensive to build prototype for a large class, simulation study is done to evaluate performance of the tool for large classes. The simulation study reveals that the proposed tool performs well also for large class size, if multiple sub-master devices are utilized in Scatter-net environment.

Key words: Mobile Technology in Education and Training, Bluetooth, mathematical modeling and simulation, Usability Issues

1. Introduction

This paper presents design aspects of developing a wireless classroom tool. The tool is to assist student evaluation processes. This paper is based on ongoing research at the University of Stavanger, Norway, to develop tools and devices that will help gauge students' participation during lectures.

There are many reasons for using tools in education; there are also many technology options available for realizing these tools. Thus, a systematic approach is needed towards developing a classroom tool, from needs analysis to the final design. This paper presents a systematic approach that consists of the following stages:

- Problem definition or scenario: Identification of the education processes that should be supported by tools
- Search for enabling technology: Choosing the appropriate technology; the technology options are basically mobile or stationary.

- Development process of the tool: E.g. programming languages, development platform, etc.
- Usability analysis (performance analysis): Evaluating how the developed tool can influence the quality of education processes; e.g. with the help of mathematical modeling and simulation, testing a prototype, etc.

The scope of the paper: The scope of this paper is limited to technological aspects only. This is a theoretical paper exploring the potentials of using newer technologies in classrooms. The prototype made for this paper consists of one master device and two slave devices on Piconet, as shown in figure-1. There is no attempt made yet to make a testing prototype of a large class consisting of many students (demonstrate proof-of-concept), thus this paper does not present pilot results or results of comparative analysis.

The structure of this paper: The next section (section-2) presents the scenario. Section-3 presents an overview of the Bluetooth technology as a technology option for developing wireless classroom tools. Section-4 presents available technologies for developing wireless applications. Section-5 analyses usability of the proposed design using a Petri net model of the tool.

Insert Figure-1 here:

Figure-1: The classroom scenario involving lecturer and students

----

2. The Scenario

The scenario describes the need for utilizing a wireless tool that complements traditional classroom evaluation. In this scenario, students use their mobile phones to submit answers for assignment tests during a lecture. The lecturer's computer automatically collects the answers (messages) sent by the students, see figure-1. Thus the proposed system is paperless, faster, and effortless. By automating the assignment tests, the faculty is also relieved from the tedious task of correcting and registering the assignment tests. The system is also cheaper because all the students' need is their (Bluetooth-enabled) mobile phones.

Insert Figure-2 here:

Figure-2: Initiating Assignment test: Students send their ID first

----

During a lecture, the lecturer asks multiple-choice questions (MCQ) that can be answered easily and quickly with a compact mobile device like mobile phone or PDA. The students first enter their student ID number (figure-2) and then they send their answers (set of lines consisting one or more of the possible answers 'A'-'E') for the questions (figure-3).

3. Bluetooth Wireless Technology

There are many wireless technologies that enable developing wireless information systems, like Wireless Personal Area Networking (WPAN) with Bluetooth Wireless technology, Wireless Local Area Networking (WLAN) with IEEE 802.11, Cellular radio network, etc. (Halsall, 2005). Out of these wireless technologies, Bluetooth wireless technology is the best technology option for developing the classroom tool described in the scenario above, for the following two reasons:

- Location: Bluetooth provides a cheap and effortless ad-hoc wireless network (called ‘Piconet’) involving the lecturer’s computer as the master and the students’ mobile phones as slaves, and
- Cost: Devices and components for creating an ad-hoc Bluetooth-enabled wireless network such as Piconet are also cheap, as the devices are inexpensive. A Bluetooth-enabled mobile phone costs as less as US$ 80.

A very short introduction to Bluetooth wireless technology is given in this section. There are numerous references available on Bluetooth technology, for example Bala Kumar (2004) and Halsall (2005).

Insert Figure-3 here:
Figure-3: Sending the answers to the Lecture’s computer

3.1 Basics: the 4-layer model

Table-1 presents the 4-layer model for communication in Bluetooth-enabled wireless information systems.

Layer-1: Bluetooth Radio Technology

Layer-1 is the host-to-host communication layer that represents Bluetooth radio technology and the hardware devices and components based on this technology. Bluetooth supports two types of channels: voice channel and data channel. Both data channel (packet switched asynchronous connection-less link - ACL) and voice channel (circuit switched synchronous connection-oriented link - SCO) communications uses a total bandwidth of 1 Mbps. Bluetooth also supports a combined data-voice SCO packet. Bluetooth communication occurs in the unlicensed Industrial, Scientific, and Medical (ISM) band at 2.4GHz.

Layer-2: Bluetooth Protocol Stack

To perform some basic functions, Bluetooth provides a number of protocols at two levels: lower-level protocols and higher-level protocols. Lower-level protocols such as host controller interface (HCI) and packet segmentation and reassembly (L2CAP) deals with fundamental communication linkage establishment and data streaming; higher-level protocols such as Service Discovery Protocol (SDP) deals with service utilization and data packing.

Layer-3: Bluetooth Profiles

A Bluetooth profile is a specification that defines the minimum requirements that the Bluetooth-enabled device must support in a specific usage scenario or user model (e.g. PDA, mobile phone, game device, etc.). These requirements define the end-user services and the features and procedures that the device must support to enable interoperability with peer devices. There are two types of profiles: *conforming* and *interoperability*. Conforming profiles like General Access Profile (GAP) and Serial Port Profile (SPP), define the core requirements for Bluetooth-enabled devices. Interoperability profiles are based on the conforming profiles; Interoperability profiles, like LAN Access Profile (LAP) and File Access Profile (FAP), define the minimum requirements for Bluetooth-enabled devices to support specific applications.

Insert Table-1 here:

Table-1: The basics of Bluetooth wireless technology

Layer-4: Bluetooth-enabled Application Types

There are two types of Bluetooth-enabled applications: 1) based on the role of the application, and 2) based on the topology of connection of the application with the other applications; these two types are not mutually exclusive.

Insert Figure-4 here:

Figure-4: State diagram for server-side and client-side activities

----

The role of the application often assumes either a server or a client role, depending on the initiator of the communication. The roles may vary dynamically; for example, in peer-to-peer environment an application may swap its role from server to client, thus classified as a peer (both client and a server). A server first advertises a service, and then it waits for any clients. Whenever a client contacts the server, the server creates a separate client handler to service the client. When the time comes to cease the service, the server stops advertising and removes the service from its database; see figure-4. A client first searches the nearby devices for services that are of interest. When a service is found, the client establishes a connection with the server and then consumes the service (figure-4). A peer in a peer-to-peer network is both a client seeking for some services, and a server offering some services. Hence, both the server-side functions and the client-side functions will be available on a peer.

Based on the topology of connection between the Bluetooth-enabled applications, applications can be one-to-one (or point-to-point) or one-to-many (multipoint). The popular configuration of Bluetooth-enabled multipoint communication is the *Piconet*, also known as ‘wireless personal area network (WPAN): In a piconet, one device can act as the master and there can be up to seven active slave devices. In addition, a piconet can support up to 255 non-active slaves that are waiting to be connected (also called parked slaves). The main functional differentiation between a master and a slave is that a master can interact with any of its slave directly, whereas no two slaves can interact directly, the interaction has to go through the connecting master. All slave devices must be located within a radius of maximum 100 m (typically 10m) of the master device. Thus, a piconet covers a small area such as a classroom, an office, or a conference room. This means, piconet can be used to set up a spontaneous ad-hoc conferencing involving multiple Bluetooth-enabled devices like laptop computers, mobile phones, PDAs, Pocket PCs, smart phones, Tablet PCs, etc.

4. Technologies for Developing Bluetooth Wireless Tool

The previous section shows that it is easy to comprehend the basics of Bluetooth wireless technology. For developing wireless information systems, a developer need not worry much about technical details as the Application Programming Interface (API) normally hides these details. This section is about how easy it is to program a Bluetooth-enabled wireless information system.

There are generally two different platforms available for developing Bluetooth-enabled wireless applications:

- The Microsoft .NET platform, and
- The Java Enterprise Edition (J2EE) platform and its vendor specific implementation.

This paper emphasizes use of J2EE simply because of practical reasons (our university teaches Java language courses only). Whether or not J2EE is better than .NET for developing Bluetooth-enabled applications is not discussed here; it is enough to say that according to our experiences, we see little difference between these two approaches. For further information on mobile application development on .NET platform, see Fox and Box (2004).

4.1 Java API for Bluetooth Wireless Technology

A very short introduction to the Java API is given in this section. For further information on Java Bluetooth API, see Bala Kumar et al (2004), Hopkins and Antony (2003).

Java API for Bluetooth Wireless Technology (JABWT) is the basic API that supports development of Bluetooth-enabled application on the Java platform. JABWT consists of two optional packages (Ortiz, 2005a; Ortiz, 2005b):

- The core Bluetooth APIs (javax.bluetooth), and
- The Object Exchange (OBEX) APIs (javax.obex).

The core Bluetooth APIs expose the Bluetooth protocol stack to developers, especially, the Service Discovery Protocol (SDP), the Serial Port Profile RFCOMM for serial emulation, and the Logical Link Control and Adaptation Profile (L2CAP), which provides connection-oriented data services to other protocols such as segmentation and reassembly operation, and protocol multiplexing. OBEX is a high-level API for exchanging objects such as electronic business cards and calendar items transmitted in the vCard and vCalendar formats. On Bluetooth, object exchange occurs over RFCOMM.

Insert Figure-5 here:
Figure-5: The change of state due to some of the important APIs
----

4.2 Implications of using JABWT API

Though there are a lot of classes and interfaces involved in developing a simple Bluetooth-enabled client-server application, a developer need not be concerned with all these details. The main functions that are involved in program are only a small set of functions. Figure-5 shows the state-diagram for the main activities that take place during a simple client-server interaction; figure-5 shows only the client-side activities. The main functions of Java API that utilized in a client application are also shown in the figure. In figure-5, a client searches nearby Bluetooth-enabled devices in a Piconet for some services.

4.3 Developing Bluetooth Applications with Microsoft .NET Technologies

On the Microsoft .NET technology platform, there are two Bluetooth stacks (also called as the Bluetooth driver) are widely in use today:

1. The Widcomm Bluetooth stack, which was the first Bluetooth stack for Windows operating systems. Most Bluetooth adapters use the Widcomm Bluetooth stack driver. Widcomm (Broadcom, 2008) licenses its software to most Bluetooth manufacturers. Consequently, the installation, configuration and use of different Bluetooth devices on Windows have a nearly identical look and feel.
2. The Microsoft Bluetooth stack, which comes with Windows XP Service Pack 2. Microsoft's Bluetooth System Development Kit (SDK) is free, whereas Widcomm Bluetooth Stack and SDK were commercial products and therefore were quite expensive.

Bluetooth stacks can be roughly divided into two purposes:

1. General-purpose implementations that are written usually for desktop computers.
2. Embedded system implementations intended for use in devices where resources are limited and demands are lower, such as Bluetooth peripheral devices (printers, scanners etc).

Developing mobile systems with Microsoft and Widcomm Bluetooth Stacks require different approaches; for instance, while Microsoft follows its regular C style, Widcomm SDK is implemented as a set of C++ classes. Besides, the latter SDK does not support sockets, but does expose virtual COM ports. As a result, one might need to maintain separate wrapper classes for each Bluetooth stack, meaning an application developed using the Widcomm SDK will not work on devices that has the Microsoft Bluetooth stack. This is the major difficulty in developing Bluetooth devices on the Microsoft platform.

4.4 Building a prototype: comparing the technologies

As noted previously, different developers using different type of Bluetooth stack (Microsoft, Widcomm, other any other propriety software) creates compatibility issue in the Microsoft .NET platform. It could be a problem as a developer has to take care of the different types of stacks. Other than the Bluetooth stack, a developer normally use that following software for developing a Bluetooth device on the Microsoft .NET platform: Bluetooth SDK, Microsoft .NET framework, Visual Studio.Net for programming, and ActiveSync; ActiveSync is a synchronization program developed by Microsoft, that allows a mobile device to be synchronized with either a desktop PC, or a server running Microsoft Exchange Server (Forsberg and Sjostrom, 2002). During development, some other APIs may be also needed, like Direct3D (Wigley, 2007) for performance improvement.

Visual Studio.Net comes with a couple of built-in emulators to *test* mobile applications. These emulators, rather surprisingly, did not allow any sort of remote connectivity between devices, perhaps due to security reasons; this means, we do need physical mobile devices for testing.

Java is known for platform-independency whereas Microsoft .Net technologies are for Microsoft platforms. Most of the mobile phones and PDAs in the mobile market today do have the support in running java applications. Java environment provides a lot of free and very powerful APIs, including Java API for Bluetooth Wireless Technology, to help developers develop Bluetooth applications for embedded devices such as mobile phones and PDAs. A Java application that runs on embedded devices is called as a Java Midlet (Riggs et al, 2003).

It is important to investigate how two different platforms (Microsoft .Net and Java) will be able to exchange data via a common set of business procedure (interoperability), and how the data exchanged can be used correctly. To be more precise, how the server application implemented in C#, will be able to handle data sent by a java Midlet through Bluetooth. Sending data as a raw stream of bytes over serial ports might work rather than other methods such as the use of the Object Push Profile (OPP) in the Bluetooth specifications. OPP can be used to push file(s) to a Bluetooth device capable of the Object Push Profile. The possibility of reading an object in a C# implemented mobile Bluetooth device is minimal is the object is created by a Java based Bluetooth device.

5. Modeling and Simulation of the Proposed System

It will be difficult for us to test the proposed system in a real classroom setting as it would demand too many mobile devices. Hence, modeling and simulation of the proposed system is done in-order to determine the usability of the system.

5.1 Usability Analysis

A crucial element in the information system success is usability, the ease or difficulty users experience with online systems and processes. For mobile applications, usability is associated with five basic design elements such as (Salmre, 2005):

- Startup time: Quick startup time is an important characteristic of mobile applications,

- Response time: Providing the response (needed information) quickly-preferably within a few seconds,
- Focused purpose: The application must have a clearly defined set of things it does very well; it must do them with a minimum number of clicks, taps, or other user gestures,
- Interactivity: Giving users the opportunity to customize their interactions with the system, with the help of icons such as shopping carts, other interactive features provides faster access, browsing, searching, and checkout procedures, and
- Consistency of experience: Because mobile devices are compact and self-contained, users naturally view the whole device as a single unified experience.

Due to brevity, this paper focuses only on response time. In addition, response time is more critical for the scenario, than the other design elements for usability.

5.2 A Simple Model of the Classroom Tool

In this subsection, a simple model is given for evaluating performance of the proposed classroom tool. The model is mainly for calculating the response time:

As mentioned earlier, the basic unit of networking in Bluetooth is a Piconet, consisting of a master and up to seven active slave devices. The lecturer's PC is designated as the master, and the first seven student mobile phones waiting to send their answers are designated as the active slaves. The other students are parked slaves, waiting for their turn to send their answers.

The lecturer's PC (the master) makes the determination of the channel (frequency-hopping sequence) and phase (timing offset-that is, when to transmit) that will be used by all student mobile phones (slave devices) on this Piconet. The radio designated as master makes this determination using its own device address as a parameter, while the slave devices must tune to the same channel and phase. A slave may only communicate with the master and may only communicate when granted permission by the master. Figure-6 shows the Piconet connecting the lecturer's computer and the student's mobile phones.

Insert Figure-6 here:

Figure-6: The Piconet connecting the lecturer's computer and the students' mobile phones.

For simulations, a mathematical model of the Piconet is needed; a mathematical model of the Piconet is shown in figure-7. Figure-7 is a Petri net model. For more details about Petri net, interested readers are referred to the standard textbooks on Petri nets, e.g. Cassandras and LaFortune (1999).

5.3 The Simulation

The model shown in figure-7 can be simulated by using any Petri net simulation tools like GPenSIM (2006). The following assumptions were made for simulations (a time slot (TS) in Bluetooth technology is 0.625 seconds):

- The students take about 12-15 TS (uniform distribution) to key their answers,
- The lecturer's computer (Piconet Master) takes 2 TS processing the incoming messages.
- The channel (radio transmission) takes 2 TS and 1 TS, for transmitting messages (slave to master) and acknowledgements (master to slave) respectively.
- Link selection (removing an active slave and selection and connection of a parked slave) takes one TS.

Insert Figure-7 here:
Figure-7: Petri net model of the Piconet-based Assignment Test
----

5.4 Interpreting the Simulation Results

Figure-8 shows the simulation results for class size of 5, 10, 15, 20, and 25 students. The total time taken for all students to submit their answers increases *linearly* with the size of the class (number of students). The average waiting time for a student to submit his answer after keying his answer, is about 4 seconds.

However, figure-8 also shows a worsening maximum waiting time; the last student connected to the Piconet may have waited for longer period of time as the size of the class increases. For example, when the class size was 5 students, the maximum waiting time was about 18 seconds. When the size of the class was 10 students, the maximum waiting time was found to be 33 seconds.

According to Palmer (2002), the likelihood that a user will not wait for interaction from an information system increases significantly when the delay extends beyond 10 seconds. Thus, for using the tool, the optimal size of the class is about 5 students. However, if a larger class is to be supported, then *Scatternet* is the solution. Multiple masters are utilized in a Scatternet, as shown in the extended model in the following section.

Insert Figure-8 here:
Figure-8: Running simulations on the Petri net model.
----

6. Improving the Performance

In order to reduce waiting times, a number of design improvements are considered:

1. *Scatternet with many sub-masters*: In addition to the master, a number of sub-masters will be employed. The sub-masters, will collect messages from a number of clients (answers from students), pack them and then forward it to the master as a single message.
2. *Acknowledgements to clients*: Clients will get acknowledgements from both the master and from the sub-master that received the message from the client. Thus, the students can leave the classroom as soon as they have their acknowledgements.
3. *XML file format for further processing*: All the data transmission between clients and sub-masters, and between sub-masters and the masters will be in XML format. This is to help further processing of the data (that is, correcting the answers and filing and record keeping).

In the extended model described in the next subsection, only the design issue dealing with Scatternet is considered; just to make the model simpler, the mechanism of acknowledging clients is not incorporated into the model. Data exchange in XML format (or in any other format) will not have any influence in the model described below.

6.1 The Extended Model

A Petri net model of the Scatternet is shown in figure-9. In this system, in addition to the master, a number of sub-masters are employed. This means, in addition to his computer, the lecturer will also carry a couple of mobile devices as sub-masters to the classroom. The Petri net model shown in figure-9 is simplified in the sense that the Internet and Bluetooth communication details are abstracted away in the figure (whereas the Petri net model shown inn figure-7 reveals these details).

Insert Figure-9 here:

Figure-9: Petri net model of the Scatternet-based Assignment Test

----

6.2 Calculating Waiting Time with Analytical Method

Waiting time is defined as the time difference between completion of processing and the start of processing when all the student answers are ready for submitting.

Let,

$n, n_{SUBM}, n_{PSIZE}, n_{PACKETS}, n_{BLOCKS}$ be number of slaves (student devices), number of sub-masters, number of messages per packet (packet size), and number of blocks (explained later), respectively.

Let also,

t_{PMSM}, t_{PMM} be the average processing time of a message at a sub-master, and the average processing time of a packet of messages at the master, respectively.

6.2.1 Basic Piconet

In a basic Piconet (with no sub-masters), total processing time is equal to:

$$T = n \times t_{PMM} \qquad (1)$$

6.2.2 Scatternet

In a Scatternet with sub-masters,

Number of packets used:

$$n_{PACKETS} = \lceil n \; div \; n_{PSIZE} \rceil$$

The fix operator $\lceil \; \rceil$ rounds the number upwardly towards nearest integer

Number of blocks needed:

$$n_{BLOCKS} = \left\lceil \frac{n_{PACKETS}}{n_{SUBM}} \right\rceil$$

Time needed for creating a packet of message at a sub-master:

$$t_{PACKETING} = n_{PSIZE} \times t_{PMSM}$$

Hence, If:

$$t_{PACKETING} > (n_{SUBM} \times t_{PMM})$$

Then,

Total processing time is equal to:

$$T = (n_{BLOCKS} \times t_{PACKETING}) + (n_{SUBM} \times t_{PMM}) \qquad (2)$$

Otherwise,

$$T = t_{PACKETING} + (n_{PACKETS} \times t_{PMM}) \qquad (3)$$

6.2.3 Application Example

For a classroom of 50 students, assuming $t_{PMSM} = t_{PMM} = 2T_S$:

If sub-masters are not used (Piconet), the waiting time for the assignment test:

$T = n \times t_{PMM} = 50 \times 2T_s = 62.5$ seconds

If five sub-masters are used then the waiting time for the assignment test:

$T = (n_{BLOCKS} \times t_{PACKETING}) + (n_{SUBM} \times t_{PMM}) = 18.5$ seconds

7. Conclusion

Wireless technologies are becoming extremely popular around the world. Consumers seem to appreciate more and more of these wireless devices. Both our lifestyle and the global economy are much dependent on the flow of information through wireless mediums such as telecommunication etc. Mobile phones have become highly available during the last decade or so. Most of us own a mobile phone nowadays, making people available almost from anywhere.

New wireless technologies are introduced at an increasing rate, and they have spread rapidly during the last few years. Wireless networks are no longer provided by big companies and corporations alone. Consumers can set up their own wireless networks these days.

Among wireless technologies, Bluetooth technology has made considerable amount of expansion and development lately. The Bluetooth market was expected to see a 71 percent increase in Bluetooth radio shipments to 500 million in 2006, according to market research firm ABI Research (EE Times, 2006). The Bluetooth market has however targeted mobile phones and PDAs (also called as smart devices) than any other devices. The majority of the new mobile phones have included the Bluetooth technology. Many useful Bluetooth applications have also been developed throughout, enabling consumers to easily connect to other Bluetooth devices as well (Ranaweera, 2007).

This paper focuses on the performance evaluation of a Bluetooth wireless technology based classroom tool. This paper is basically a technology paper; usability issues like context-intensity, Human-Computer-Interaction, user-friendliness, exam frauds like copying, noise pollution due to mobile device use, etc. are not discussed in this paper.

The classroom tool proposed in this paper is to help lecturers automate their assignment tests. Though non-wireless technologies are available to realize the proposed tool, Bluetooth wireless technology is considered because non-wireless technology based devices are fixed (can not be moved from classroom to classroom) and clumsy (they take desk space). In this paper, first, a testing prototype of the tool is presented; the prototype consists of one master device and two slave devices on Pico-net. As it will be expensive to build prototype for a large class, simulation study is done to evaluate performance of the tool for large classes. The simulation study reveals that the proposed tool performs well also for large class size, if multiple sub-master devices are utilized in Scatter-net environment.

- Developing the proposed tool is easy because this basically involves programming, demanding knowledge of Application Programming Interfaces (APIs) like JABWT.
- Developing such a tool is also cost effective because the components and devices involved (such as Bluetooth-enabled mobile phones) are already in use and are cheap.
- Finally, the proposed tool is efficient: with the help of this tool, it is not only possible to automatically collect and correct the assignment tests, it is also possible to inform the results when the final question was answered.

Similar applications: Although the proposed tool is associated with a classroom environment, similar types of tools using Bluetooth technology can be developed to support other environments. For example, a voting application in which a group of people were asked to vote using their mobile devices; if the group is gathered at one place at the same time, then the proposed system can be used without major changes. Another example is during television shows where the audience was asked to take part in answering questions presented by a presenter.

Limitations: Using a mobile phone in a classroom can be considered as an annoyance or disruption of the lectures, considering noises arising from keying the answers. Another issue is copying or dishonesty in completing the assignment tests.

ACKNOWLEDGEMENTS: In his Master thesis, Kushil Ranaweera (Ranaweera, 2007) made a prototype of the tool described in this paper. The practical experience described in Ranaweera (2007) greatly helped redesigning the tool.

REFERENCES

1. Bala Kumar, Kline, C. Thompson, T. (2004) *Bluetooth Application Programming with the Java APIs*. Morgan Kaufmann Publishers
2. BroadCom (2008) Accessed on May 15, 2008: http://www.broadcom.com
3. Cassandras, G. and LaFortune, S. (1999) *Introduction to Discrete Event Systems*. Kluwer Academic
4. EE Times (2006) *Bluetooth shipments could hit 500 million in '06*. http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=183700144 (Accessed May 15, 2008)
5. Forsberg, C. and Sjostrom, A. (2002) *Pocket PC Development in the Enterprise*. Addison-Wesley
6. Fox, D. and Box, J. (2004) *Building Solutions with the Microsoft .NET Compact Framework: Architecture and Best Practices for Mobile Development*. Pearson Education
7. GPenSIM (2008) *General Purpose Petri Net Simulator*. http://www.davidrajuh.net/gpensim (Accessed May 15, 2008)
8. Halsall, F. (2005) *Computer Networking and the Internet*. 5th Edition, Addison-Wesley
9. Hopkins, B. and Antony, R. (2003) *Bluetooth for Java*. Apress
10. Ortiz, E. (2008) *Using the Java APIs for Bluetooth Wireless Technology*, Sun Developer Network Site: http://developers.sun.com (Accessed May 15, 2008)
11. Palmer, J. (2002) Designing for Web Site Usability, *Computer*, pp. 102-103, July 2002
12. Ranaweera, K. (2007) *Building an interactive Bluetooth-based wireless classroom tool with Microsoft .Net Technologies*. Master Thesis, Department of Electrical Engineering and Computer Science, University of Stavanger, Norway, Spring 2007
13. Riggs, R. Taivalsaari, A., Peursem, J., Huopaniemi, J., Patel, M. and Uotila, A. (2003) Programming Wireless Devices with the Java(TM)2 Platform, Micro Edition. 2nd Edition, Addison-Wesley
14. Salmre, I. (2005) *Writing Mobile Code*. Addison-Wesley
15. Wigley, A., Moth, D., and Foot, P. (2007) Microsoft® Mobile Development Handbook. Microsoft Press

Layer	**Issue**	
4. Application	(Two ways of classifying the applications)	
	Role of the application (Client, server, peer)	Topology of connection (point-to-point, multipoint)
3. Profile	(Two types of profiles)	
	Conforming (e.g. GAP, SPP)	Interoperability (e.g. LAP, FAP)
2. Protocols	(Two levels of protocols)	
	Low-level protocols (e.g. HCI, L2CAP)	Higher-level protocols (e.g. SDP)
1. Radio technology	(Two types of channels)	
	Data channel (Asynchronous connection-less)	Voice channel (Synchronous connection-oriented)

Table-1: The basics of Bluetooth wireless technology

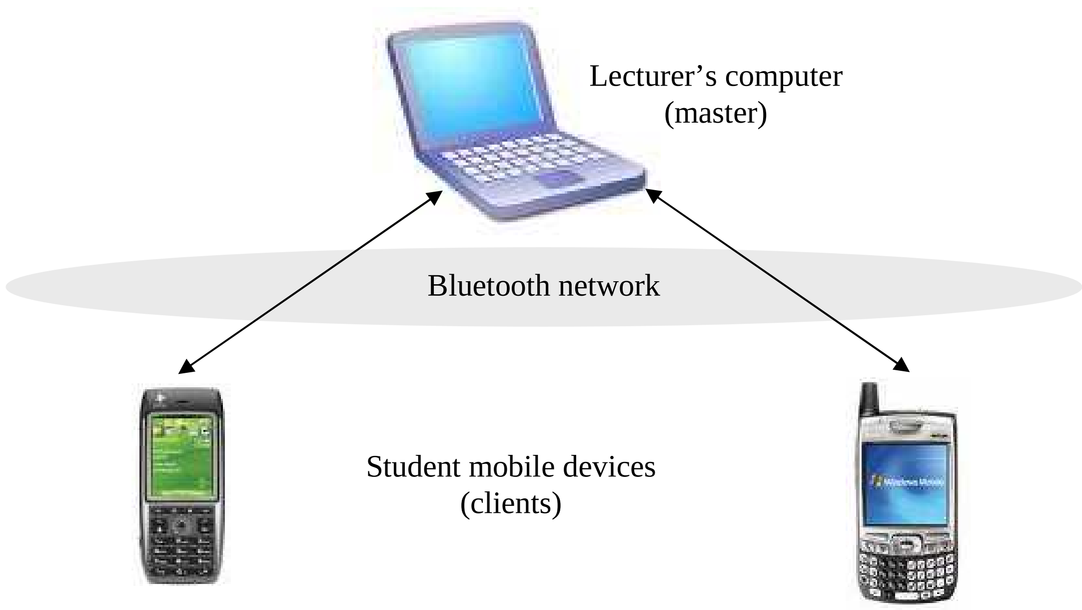

Figure-1: The classroom scenario involving lecturer and students

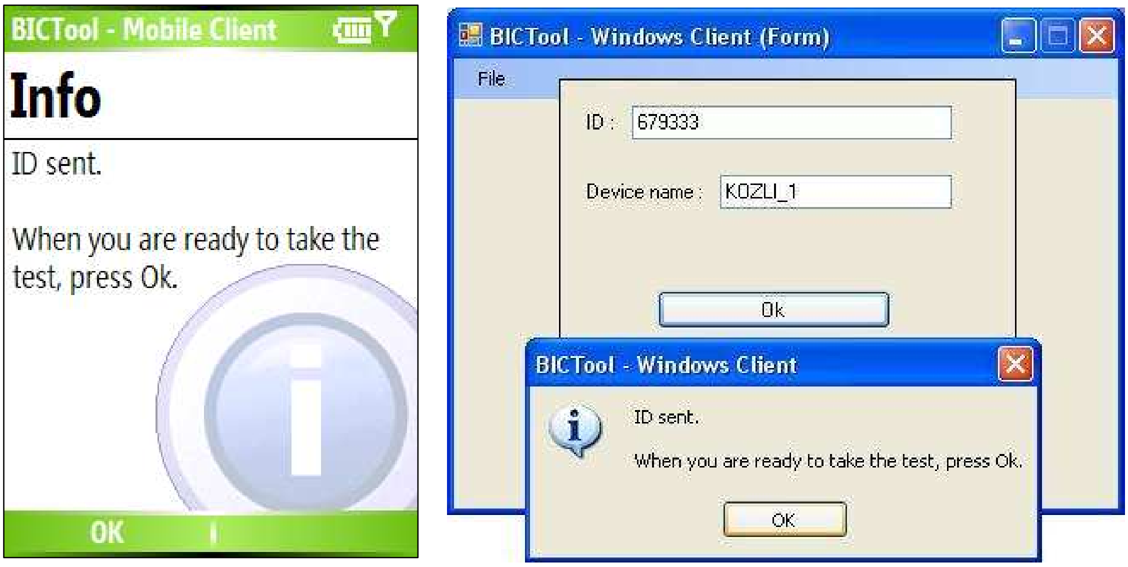

Figure-2: Initiating Assignment test: Students send their ID first

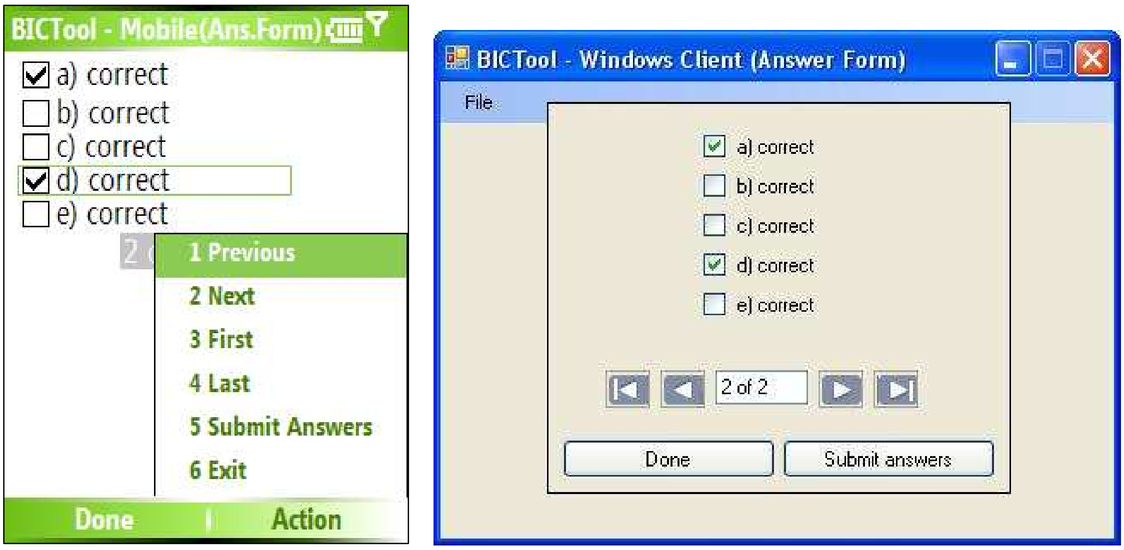

Figure-3: Sending the answers to the Lecture's computer

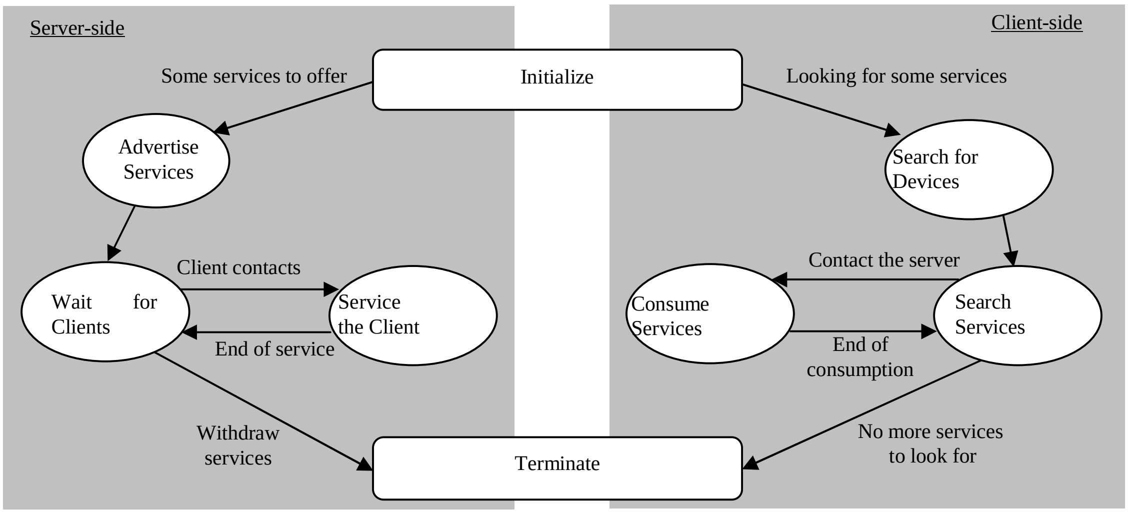

Figure-4: State diagram for server-side and client-side activities

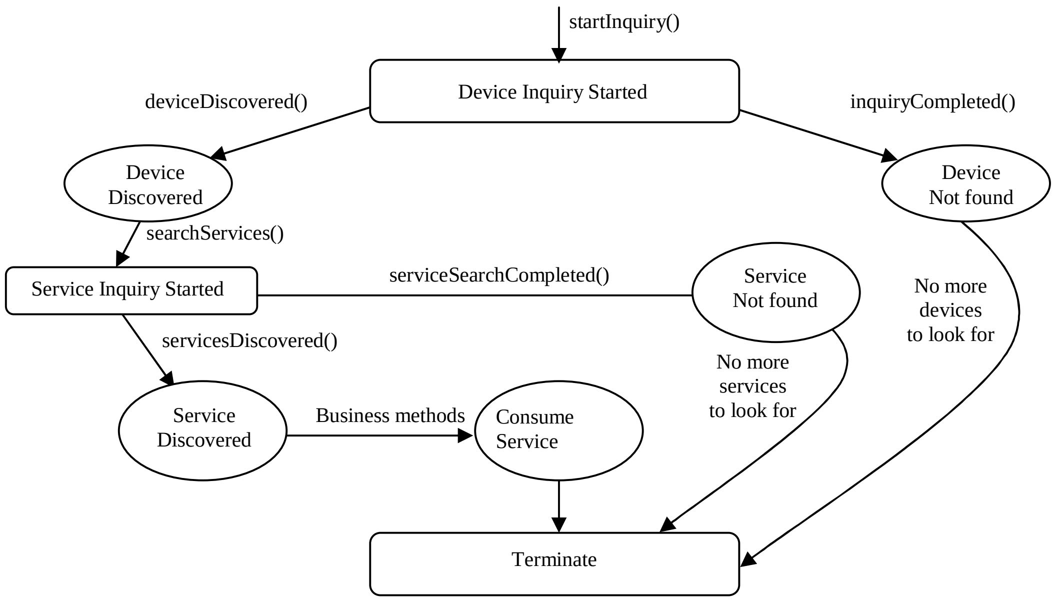

Figure-5: The change of state due to some of the important APIs

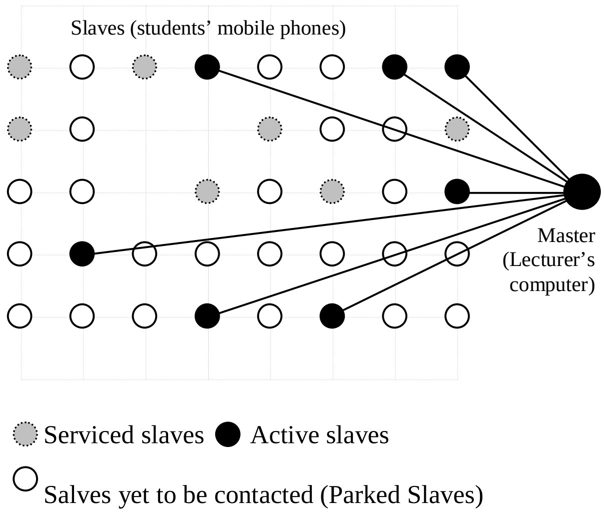

Figure-6: The Piconet connecting the lecturer's computer and the students' mobile phones.

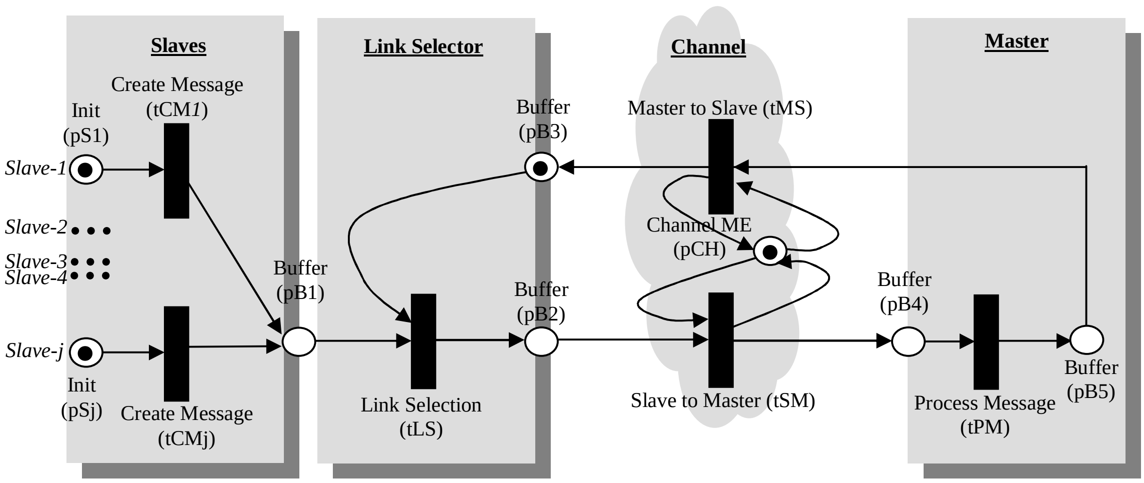

Figure-7: Petri net model of the Piconet-based Assignment Test

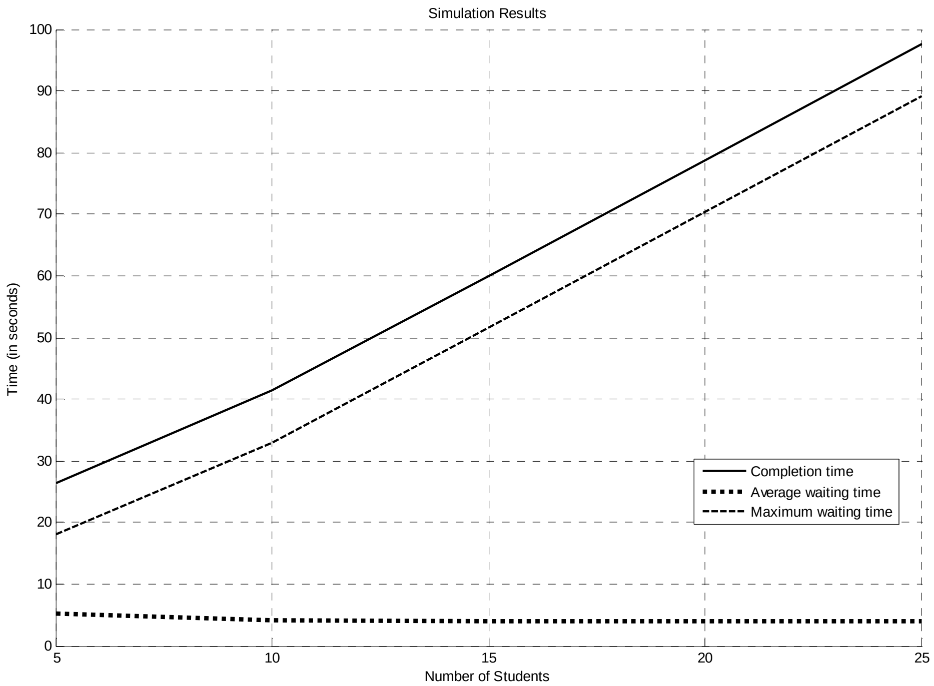

Figure-8: Running simulations on the Petri net model.

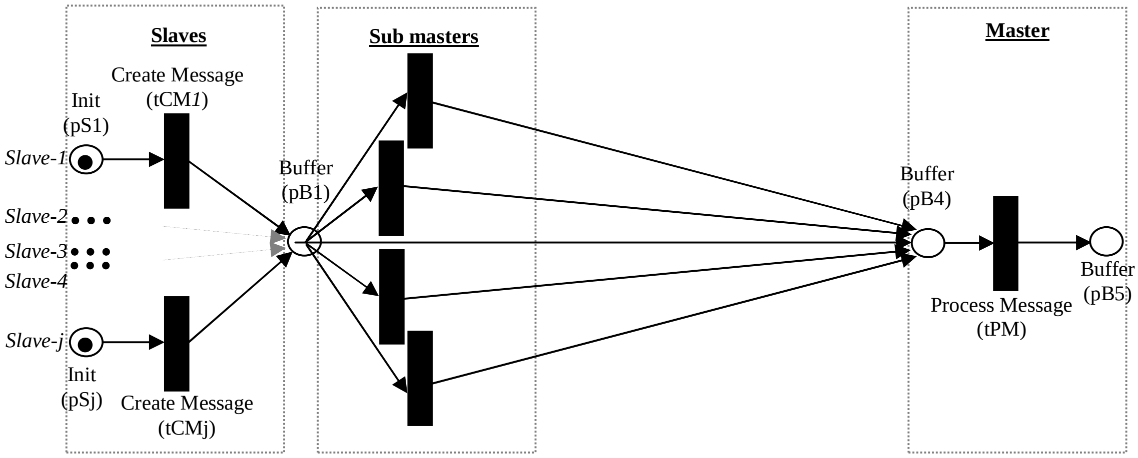

Figure-9: Petri net model of the Scatternet-based Assignment Test