Western Graduate&PostdoctoralStudies

Western University
Scholarship@Western

Electronic Thesis and Dissertation Repository

February 2013

A Support System for Graphics for Visually Impaired People

Hao Xu
The University of Western Ontario

Supervisor
Dr. Helmut Jürgensen
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

© Hao Xu 2013

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Graphics and Human Computer Interfaces Commons

Recommended Citation
Xu, Hao, "A Support System for Graphics for Visually Impaired People" (2013). *Electronic Thesis and Dissertation Repository*. 1124.
https://ir.lib.uwo.ca/etd/1124

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca.

A Support System for Graphics for Visually Impaired People

by

Hao Xu

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Hao Xu 2013

THE UNIVERSITY OF WESTERN ONTARIO
School of Graduate and Postdoctoral Studies

CERTIFICATE OF EXAMINATION

Supervisor

Helmut Jürgensen

Examiners

Mike Bauer

Sylvia Osborn

Stuart Rankin

The thesis by

Hao Xu

entitled:

A Support System for Graphics for Visually Impaired People

is accepted in partial fulfillment of the
requirements for the degree of
Master of Science

Date

Chair of the Thesis Examination Board

Abstract

As the Internet plays an important role in today's society, graphics is widely used to present, convey and communicate information in many different areas. Complex information is often easier to understand and analyze by graphics. Even though graphics plays an important role, accessibility support is very limited for web graphics. Web graphics accessibility is not only for people with disabilities, but also for people who want to get and use information in ways different from the ones originally intended.

One of the problems regarding graphics for blind people is that we have few data on how a blind person draws or how he/she receives graphical information. Based on Katz's pupils' research, one concludes that blind people can draw in outline and that they have a good sense of three-dimensional shape and space.

In this thesis, I propose and develop a system, which can serve as a tool to be used by researchers investigating these and related issues. Our support system is built to collect the drawings from visually impaired people by finger movement on Braille devices or touch devices, such as tablets. When the drawing data is collected, the system will automatically generate the graphical XML data, which are easily accessed by applications and web services. The graphical XML data are stored locally or remotely. Compared to other support systems, our system is the first automatic system to provide web services to collect and access such data. The system also has the capability to integrate into cloud computing so that people can use the system anywhere to collect and access the data.

Keywords

Accessibility, Web Graphics, Semantics, XML

Acknowledgments

I would like to gratefully acknowledge the enthusiastic supervision of Dr. Helmut Jürgensen during this work. His continued guidance, perceptive comments, and invaluable inspiration made my experience as a graduate student extremely fruitful. He was always there to offer encouragement and it is difficult to overstate my gratitude to him.

I would like to thank Dr. Mike Bauer, Dr. Sylvia Osborn and Dr. Stuart Rankin for reading this thesis and for their comments and suggestions on how to improve this work.

I wish to thank Dr. Stephen Watt, Dr Steven Beauchemin for the courses they offered in the past.

Last but not least, I want to express my gratitude to my wife Jie Chen, for teaching me how to appreciate and enjoy life, and my parents, Liucun Xu and Jiying Qu. To them I dedicate this thesis.

Table of Contents

List of Tables

List of Figures

List of Appendices

Chapter 1

1 Introduction

As the Internet plays an important role in today's society, graphics is widely used to present, convey and communicate information in many different areas [1]. Complex information is often easier to understand and analyze by graphics. Even though graphics plays an important role, accessibility support is very limited for web graphics [2]. Web graphics accessibility is not only for people with disabilities, but also for people who want to get and use information in ways different from the ones originally intended.

The web accessibility rules for graphics state the following: the information presented in images must be accessible to all users, including users with non-visual devices [2]. These rules are often violated. A simple compliance test, WAVE [3], is available on the internet. However, compliance does not mean that the graphics is truly accessible as essential information may be missing in an alternative representation.

There are two major categories of web graphics today, raster-based and vector graphics formats. Raster-based graphics is widely used in photographic images, but has many limitations. Modern raster graphics standards include tiff, jpeg, bmp etc. The problems with raster graphics include the following: it does not support searching the graphics for contents, changing graphics, nor interacting with the graphics at download time. Vector graphics has many advantages compared to raster-based graphics, such as scaling without losing quality and smaller data size. Scalable Vector Graphics is more a ‘‘final form’’ presentation, which involves some drawbacks in the direct creation, modification and access to complex, highly structured, diagrams [4]. Some drawing tools embed additional information in an SVG document, so that higher-level structure (e.g., connectivity information) can be recovered by the drawing tool, when the document is subsequently modified [4]. Both types of graphics formats have the following problem in common: they capture images at a low level of abstraction.

Many researchers introduced alternative approaches to make graphics accessible to people with disabilities by representing graphics through an auditory interface, tactile

drawings [5], text description, etc [4]. Generally speaking, there are two major categories for these approaches. One is to use alternative presentations. Another one is to overcome the SVG (Scalable Vector Graphics) limitations by adding additional information. In the paper of "GraSSML: Smart Schematic Diagrams, XML and Accessibility" [6], these two categories are termed "bottom-up approach" and "top-down approach".

The "Bottom-up" approach starts from the graphical representation, uses or extracts information from it, and generates alternative presentations. This procedure seems to address the graphics accessibility problem. However, there are some limitations, such as the lack of information about overall features.

Instead of the "bottom-up approach", some researchers now tend to a "top-down" approach. This approach starts with the meaning of the diagram instead of its representation. The diagram will no longer be defined by its graphical representation. Instead, an abstract entity including structure and semantics is used. This approach is exemplified by GraSSML. GraSSML defines a high-level description language. It collects the structure and semantics of a diagram at the creation stage and generates different presentations for every module. The paper [7] on GraSSML inspired me to create the graphics support system for visually impaired people.

One of the problems regarding graphics for blind people is that we have few data on how a blind person draws or how he/she receives graphical information. Based on Katz's pupils' research [8], one concludes that blind people can draw in outline and that they have a good sense of three-dimensional shape and space. In this thesis, I propose and develop a system which can serve as a tool to be used by researchers investigating these and related issues. Our support system is built on collecting the drawings from visually impaired people by finger movement on Braille devices or touch devices, such as tablets. When the drawing data is collected, the system will automatically generate the graphical XML data, which are easily accessed by applications and web services. The graphical XML data are stored locally or remotely. Compared to other support systems, our system is the first automatic system to provide web services to collect and access such data. The

system also has the capability to integrate into cloud computing so that people can use the system anywhere to collect and access the data.

This paper begins with an introduction of web graphics accessibility for visually impaired people. The background and related work are discussed in Chapter 2. Following the discussion of the support system overview in Chapter 3, the paper covers the support system hardware and software in Chapter 4 and 5. Chapter 6 examines the support system and relevant results. The final section contains conclusions and thoughts on future work.

Chapter 2

2 Background and Related Work

In the past, a collection of tools has been offered to access information. PC (personal computer), mobile devices and tablets have changed the way people communicate. Ideally, one hopes that these new technologies would benefit everyone equally. Even though hope has been placed in the early development stage, there are still a lot of accessibility and usability issues for visually impaired people. Researchers introduced a series of presentation alternatives to resolve the accessibility issues for visually impaired people, such as audio presentation, tactile presentation, text presentation, mathematics presentation and graphics presentation. This paper focuses on the discussion of the graphics presentation.

Pictures are the main media in graphics presentation. Graphics can be distinguished according to two categories based on their presentation formats [9], real-world phenomena representation, and abstract representation. The real-world phenomena representation refers to pictures, and abstract representation refers to diagrams.

Under the category of pictures, there are photographs, navigational maps and structure diagrams such as architectural plans and medical sketches [10]. Diagrams are a larger category, as humans tend to use many kinds of diagrams to organize data and to ease interpretation tasks [10]. Sub-categories of diagrams include statistical charts including bar charts, histograms, pie charts and graph diagrams [10]. In addition to type of graphics, time also plays an important role for the presentation of graphics. Static graphics will not change once they are complete. Dynamic graphics change from time to time. Considering for example a middle-sized city map, it will change in different stages of city infrastructure development. Once a new subdivision construction plan has been approved, the map is unlikely to change. Later, the subdivision needs to expand; the map will be examined and changed accordingly. This type of punctuated change occurs in many types of graphics, including architectures for buildings and software, city maps (as new roads are built) and circuit diagrams [10]. This implies that there is an iterative life

cycle for graphics where a graphic has its requirements specified, it is authored and after an arbitrary amount of time, the requirements are changed and the graphic is updated [10].

The above discussion provides the interpretation of graphics. However, it does not offer any solutions for how audio/tactile diagrams should be presented to visually impaired people, or how to make a "meaningful" tactile graphic. As a result, there were a large number of projects to approach graphics presentation through audio, tactile/haptic or multimodal presentation [10].

2.1 Audio Presentation of Graphics

Brown et al. [11] identified several design principles for non-visual diagram access, including overview, search, recognition and representational constraints. Overview provides general navigation information [10]. A facility to search explicit information is essential for providing diagram details [10]. Recognition provides access to implicit features [10]. Challis et al. [12] show in their work on music presentation that simple visual to tactile translations do not always result in a workable document. Graphic user evaluation is able to define representational constraints. The guidelines provide a good start to design tactile graphics. The "Technical Drawings Understanding for the Blind" (TeDUB) [13] project is significant to establish the guidelines. The researchers approached experts and users regarding what type of information should be conveyed through particular types of diagrams, such as those of software architecture [10]. This type of user engagement produced a deeper understanding of the intentions of authors and readers in order to produce sensible interactions that would not only allow the user to perceive aspects of the diagram, but also aid in the navigation [10].

2.2 Tactile Presentation of Graphics

Most of tactile graphics research projects try to duplicate the visual scene as closely as possible. For the tactile rendering of photographs, the most notable example of a system is the work by Way [14] on the TACTICS project [12]. This system uses image processing techniques to discover boundary areas and height differences in order to

generate a static tactile picture explored by a blind person [14]. Tiger embossers (Braille Printer) produced and sold by ViewPlus Technologies are able to produce superior high resolution graphics. The Tiger Braille dot has preferred shape, customizable height and enhanced durability features [15]. It is smoother to the touch, making frequent and extended Braille reading more enjoyable while maintaining Tiger's detail in producing tactile graphics. Tiger offers three different Braille dot heights allowing users to customize their documents to better suit individual needs. It employs a pyramid-like shape making it more durable than conventional Braille while continuing to maintain perfect Braille spacing [15]. Like the TACTICS project [14], the Tiger embossers did not use visually impaired people's input data to represent the graphical information. The graphical information is extracted from existing graphics for sighted people.

2.3 Multimodal Presentation of Graphics

A common trend in tactile graphics in recent years is to combine audio output with tactile pictures to assist information navigation and comprehension. This process takes a pre-defined static printout placed on a sensitive touch pad which transmits finger positions to a computer to play associated audio information. These systems have the advantages to convey speech and non-speech information to the reader with tactile navigation. The examples of this approach can be found as follows: the Talking Tactile Tablet by Touchgraphics Incorporated and the IVEO Tablet by Viewplus Technologies [14, 16, 17]. One of the first such systems was proposed in detail in [18]. There are several problems related to this type of technology. First, it cannot support dynamic graphics. Everything in the system is predefined, which means that it needs a lot of effort to generate a single multimodal tactile graphic. Second, the detection of the finger positions is difficult on static printout, which can result in inaccurate audio reporting, or no audio information at all [10].

In the above discussion, we have presented some type of audio and tactile graphics through several media. Most of the approaches were "bottom-up" starting from the graphical representation, using or extracting information, and generating alternative presentations. They seem to address the graphics accessibility problem, but they have

some limitations due to the lack of explicit information. Also, they generate the graphics presentation through manual preparation so that human intervention may be needed. A newly developed automatic finger tracking system makes it possible to reconstruct blind people's tactile reading in real time and to automatically analyze finger movements during Braille text reading and tactile picture recognition [19]. The system is using high resolution cameras and it presents results indicating how Braille readers can increase awareness of their own reading styles. This approach opens up the opportunity for future Braille education to become more evidence-based and, at the same time, creates a new research field: contrastive studies of language in its auditory, visual and tactile manifestations [19]. However, this tracking system will generate a large amount of data with one simple finger movement. The recorded data cannot be easily accessed by other applications.

The recent research project, "GraSSML: accessible smart schematic diagrams for all" [7] provides a new approach called Graphical Structure Semantic Markup Languages (GraSSML) to capture the structure and semantics through high-level diagram description languages and to transform information into various types of multimedia models, such as voice, text, and sound to make the graphics smart. Therefore, the diagram structure and semantics will be generated at the creation stage. In this way, graphics will be widely accessible to sighted or visually impaired people and to people with other special needs.

Chapter 3

3 Support System Overview

In discussions with research groups in Germany and UK, it was found that there is no such a system to record data on how a blind person draws or how he/she receives graphical information. We develop a system which can serve as a tool to be used by researchers investigating these and related issues. Our support system include two .NET [20] applications. A client (front end) application is used to collect and generate graphical XML data based on people's drawings. The server (back end) application stores the data in an SQL [21] database using the XML data format. The server application provides the web service. The client and server applications communicate through a SOAP interface seamlessly.

A planar Braille display is used to capture the graphics information by tracing the finger movements of a visually impaired person. Other devices could be used as well like the Talking Tactile Tablet [22], a digitizing tablet or a tablet PC. Finger movement coordinates are converted into various XML formats, such as INKML [23] or SVG [4]. A client application renders the finger movement on the screen simultaneously. When the movement is complete, the XML file is generated, sent to the web server and stored in a database.

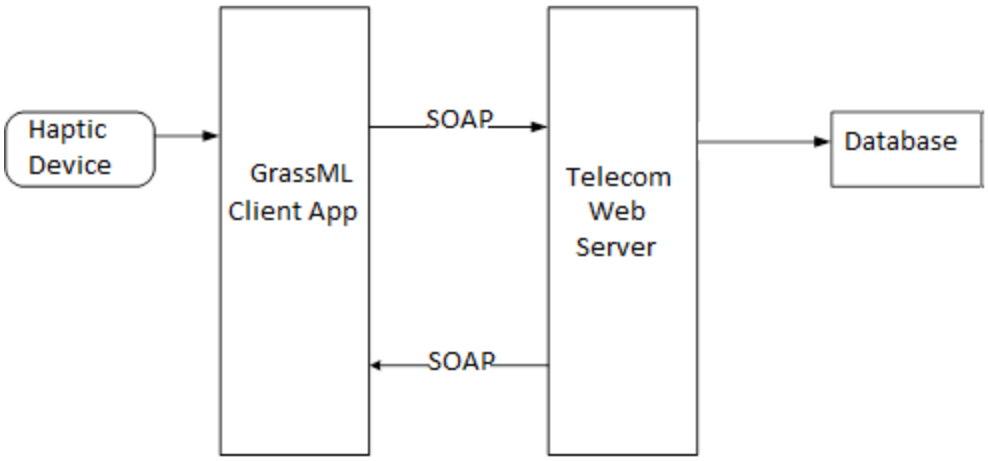

Figure 1: Support System Architecture

Chapter 4

4 Support System Hardware

The support system is implemented to use the tactile display device DMD 120060. Other tactile devices could also be used. Some examples are described below.

4.1 Tactile Display DMD 120060

The tactile display DMD 120060 [24] is a graphics and text display for blind computer users. It consists of the following items:

- The display unit case with the display surface and the driver and decoding electronics
- The processor unit case containing the power supply, most of the microprocessor control hardware and various switches and connectors
- Two operational finger position sensors

Figure 2: Braille Display DMD 120060

4.1.1 The Display Unit

The display unit is contained in a steel case. The connection to the processor unit is established via two cables, a 50-wire flat cable and a separate round cable supplying the power. The actual display area on its top is a rectangle of about 370 × 186 mm. It uses the

single-Braille-point B60001 as its basic component. It consists of a matrix of Braille points, arranged in 60 rows and 120 columns, about 3.08 mm apart horizontally and vertically.

4.1.2 The Processor Unit

The processor is contained in a separate case, together with the power supply for the whole tactile display and the various connectors and switches. It uses a Motorola 6809 as the CPU. The CPU has about 16K of RAM and 32K of EPROM. Several serial communication ports are available for communication with the host computer. Communication to the display is in binary via several PIA ports.

4.1.3 The Sensors

If the finger reading of the display also moves a sensor, then the position of the finger on the display can be determined. This can be used for a systematic study of movements during reading as well as for the interaction with the computer. The principle is as follows: The coils of the pins receive a very short impulse one after the other; the coil in the sensor detects the resulting magnetic field and returns an interrupt to a counter. As the sensor coil covers more than one point, every “scan” will yield more than one point for the sensor position. Several strategies for sampling these data are available:

- Output of every single point;
- Output of every second point (by rows and columns);
- Output of the centre of gravity of the observed points.

We can switch the different modes to verify the precision and error of the finger position.

4.1.4 The Data Transmission Interface

The processor unit has two serial ports with 9600 baud. One transmits the sensor position data. The other sends the data to execute the commands. The flow control uses RTS and CTS. The data format is one start bit, eight data bits with bit eight not used, two stop bits, and no parity. The data flow from the processor unit to the computer can be interrupted

by a hardware handshake. If this happens for an extended period, position data in the output buffer of the processor unit may be overwritten by new position data.

At the beginning of a scanning cycle, the micro-processor sends STX CR. About 3 milliseconds after this, the first scan impulse affects the sensor coils. If there are no position data, again STX CR is sent again after the scan period is complete. The repeat time is less than 4 sec if all pins are sampled.

The coil position data contains the row number and one or more column numbers with sensor identification. Entries in a line are separated by commas, and CR indicates the end of a line. A typical data sequence would look as follows:

STX CR
STX CR
STX CR
$r_{1,1}r_{1,2},s_{1,1}c_{1,1,1}c_{1,1,2}c_{1,1,3},s_{1,2}c_{1,2,1}c_{1,2,2}c_{1,2,3},s_{1,3}c_{1,3,1}c_{1,3,2}c_{1,3,3}$CR
$r_{2,1}r_{2,2},s_{2,1}c_{2,1,1}c_{2,1,2}c_{2,1,3}$CR
STX CR

All data are in ASCII. Together $r_{i,1}r_{i,2}$ denote the *i*th row co-ordinate. An entry specifies the sensor number $s_{i,j}$ 1 or 2 and column co-ordinate $c_{i,j,1}c_{i,j,2}c_{i,j,3}$ recognized by that sensor. We can use Com Port monitor software to explore the communication data shown in figure 3.

Sensor Data Samples:

26,1039
27,1039
26,1039
27,1039
24,1042
19,1050
15,1059
14,1075
15,1090
17,1105
21,1116
22,1117
32,1115

For example, 26,1039 means that sensor 1 has been found in row 26 and column 39.

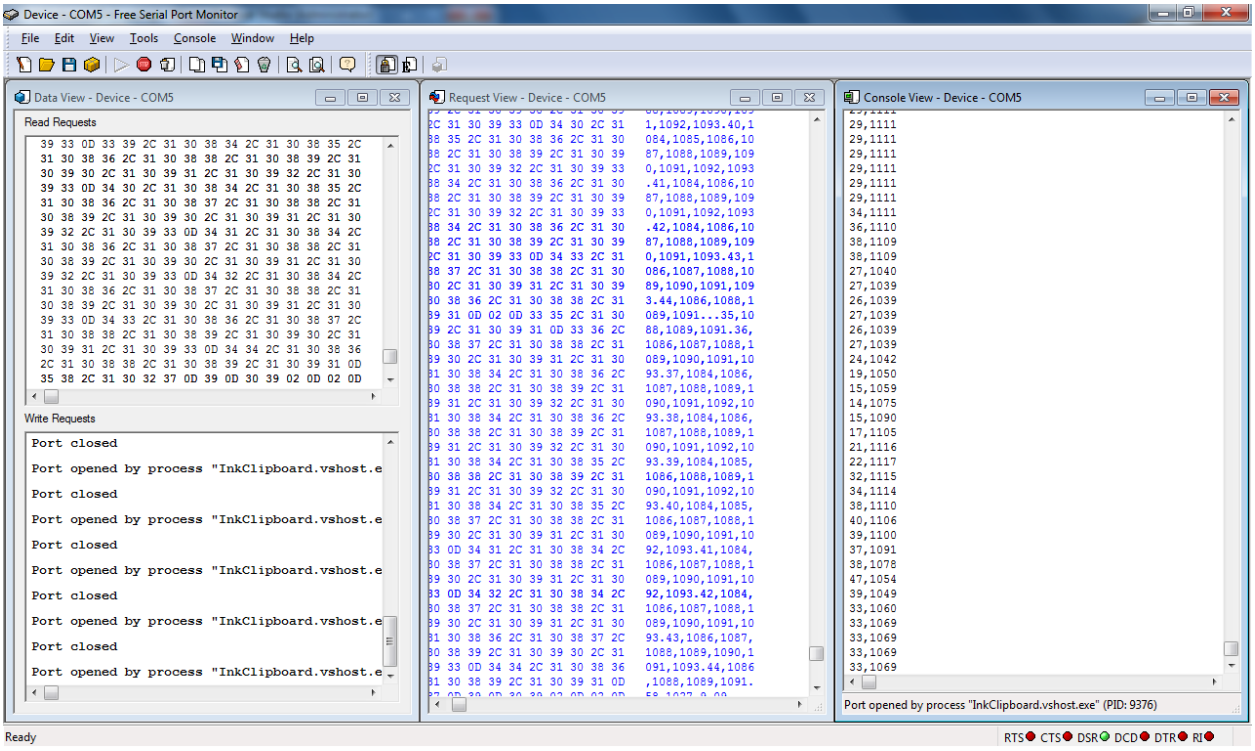

Figure 3: COM Port Data Monitor

4.2 The New Devices

4.2.1 The Novel BrailleDis 9000 Pin-Matrix Device

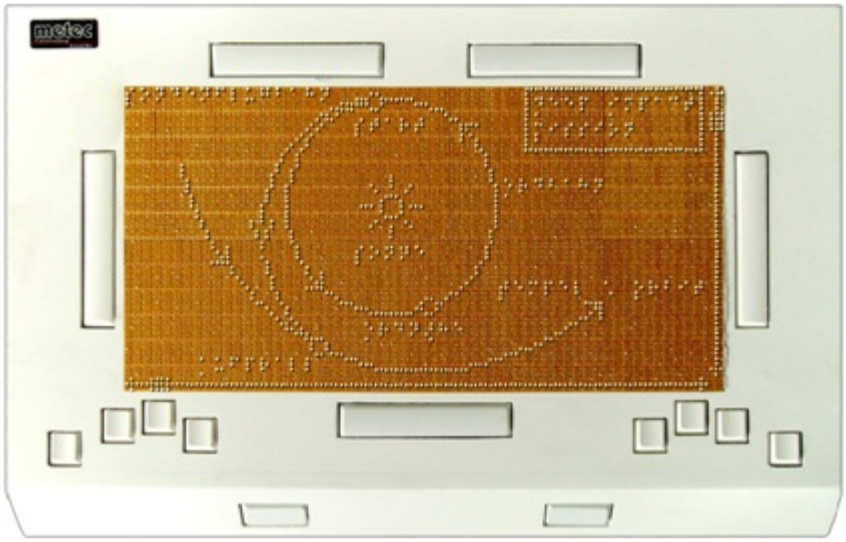

Figure 4: Novel BrailleDis 9000 Pin-Matrix Device

The new BrailleDis 9000 pinmatrix device consists of 7200 pins provided by 720 10 pin vertical Braille modules [25]. It is connectable via the standard USB 2.0 interface, but an external power supply is necessary [25].

4.2.2 The HyperBraille Device

Figure 5: New HyperBraille Device

Funded by the German Federal Ministry of Economics and Technology (Bundesministerium für Wirtschaft und Technologie, BMWi), HyperBraille is a project engaged in developing a touch-sensitive tablet display for blind and partially sighted users [26].

This pin-matrix display drastically increases the amount of information perceivable to blind computer users through both hands, and enables them to experience spatial structures and graphic symbols as additional information [26]. Ideally, objects such as text blocks, tables, menus and other elements of the Windows user interface can be fully mapped to the pin-matrix display [26]. Moreover, geometric drawings, floor plans, diagrams and other information can be made accessible to visually impaired students in teaching [26]. Developing the software required to control the tablet display is a major part of the HyperBraille project, which focuses mainly on all standard Office and Internet applications [26].

4.3 Talking Tactile Tablet

Figure 6: Talking Tactile Tablet Display

The Talking Tactile Tablet, nicknamed “TTT” or "T3", is an inexpensive, rugged and simple new computer peripheral device designed for use as a “viewer” for audio/tactile materials [27]. It consists of a computer touch screen connected to a PC via a USB cable. A hinged, weighted frame holds one of a large collection of tactile graphic sheets motionless in place against a touch-sensitive surface [27]. When a user presses points on the tactile sheet that is mounted onto the TTT, his or her finger pressure is transmitted through the flexible PVC material, and the location of the point pressed is sent to the computer [27]. By comparing the position of each point against a database of predefined hotspots, the computer is able to provide identifying audio feedback to the user, as confirmation and elaboration of the information supplied through touching [27].

4.4 Other Haptic Input Devices

There are other haptic input devices on the market, such as graphics tablets, touch screens and tablet computers. Compared with tactile devices, they are less useful for visually impaired people because there is no haptic feedback from these devices.

4.4.1 Graphics Tablet

A graphics tablet [28] is a computer input device to collect hand-drawn images and graphics from users. These tablets may also be used to capture data or handwritten signatures, or to trace an image from a piece of paper.

Figure 7: Graphics Tablet Display

4.4.2 Touch Screen

A touch screen [29] is an electronic visual display that can detect the presence and location of a touch within the display area. Touch screens can also be operated by a stylus. They are common in devices, such as all-in-one computers, tablet computers, smartphones, and game consoles.

4.4.3 Tablet Computer

A tablet computer is a general-purpose computer contained in a single panel, which uses a touch screen [30] as input device. Modern tablets are operated by fingers, but earlier tablets needed a stylus. In 2000, Microsoft introduced its first Windows tablet. After years of limited success, Apple's iPad created a tablet revolution in 2010. Android tablets and the BlackBerry PlayBook also joined the tablet market.

Chapter 5

5 Support System Software

Haptic sematic graphics means that graphics can be presented in a meaningful way by touch. Our support system is used to collect graphical information entered by visually impaired people. One can use different tactile devices to record the data. In the future, when one uses these recorded data to train different groups of users, they will be able to access the graphics by haptic feedback or other multimodal representation.

The test person draws graphics on the haptic device through finger movement. The GraSSML Client generates semantic graphics. Sematic information, such as the shape or size of the drawing, will be generated by either user input through speech recognition [31] or some evaluation tools, such as shape recognition [32]. Evaluation tools can be used from the client side or the server side. Graphical XML data with special sematic tags will be generated, transferred, and stored in a remote database based on web services. Today most applications of "graphics for blind people" do not attempt to collect data at all. They are stand-alone applications, which mean that they will only be able to collect and process data locally. We built the GraSSML Client to collect data both locally and remotely by configuration settings. This will enable researchers to collect data world-wide on diverse devices. This issue is potentially important for the community of visually impaired individuals as there are usually too few test persons available at any given location. The GraSSML Client collects raw data of the finger movement on the haptic device. The raw data are transformed into different XML data files and rendered in the client GUI. When the finger movement is complete, the XML data are transferred to a telecom server.

The primary objectives for the client include the provision of:

- a device driver to collect the finger movement data,
- the data conversion from raw data to Graphical XML data,
- web service operations and
- integrity and security guarantees for all communication

Client application technical requirements:

- Use of Microsoft .NET 3.5 or greater,
- Support of both windows x86 and 64-bit platforms,
- Remain database agnostic: minimal support of SQL Server and Oracle DBMS,
- Should not require Internet Information Server (IIS),
- Development in C# and Visual Studio 2005 or greater.

5.1 GrassML Client Application

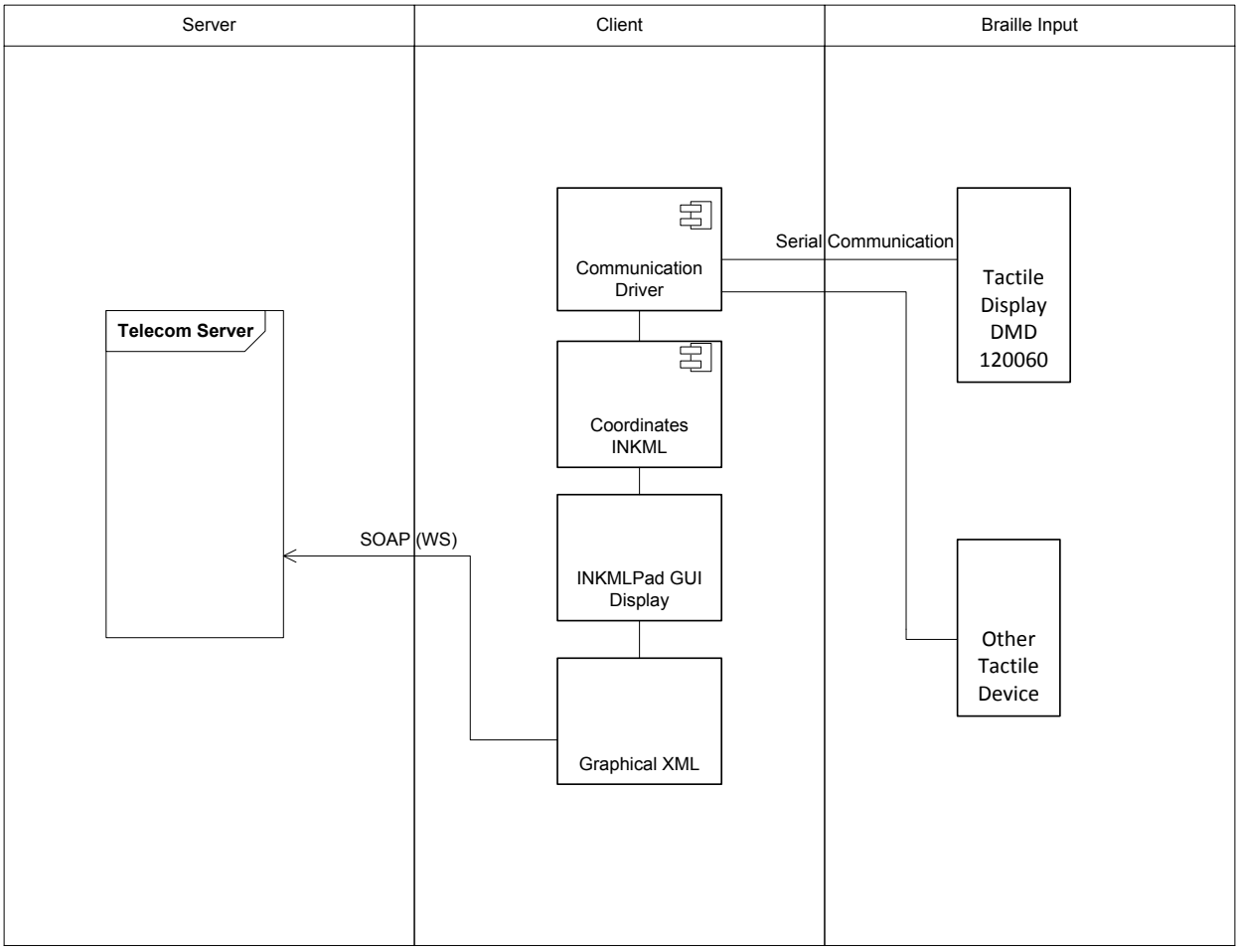

Figure 8: GrassML Client Application Architecture

The above diagram depicts the major components involved in the GrassML client architecture. The GrassML Client application uses the Windows .NET framework, in

order to integrate the Tactile Display interface, render finger movement into the InkPad application and transmit graphical XML data into the telecom server. This allows the GrassML Client architecture to collect and store data remotely.

5.1.1 InkMLPad Application

InkMLPad [33] is an ink editor application to create, edit and view digital ink files on a Windows platform. The features of InkMLPad are:

► It uses the Microsoft Tablet PC SDK for ink capture, rendering and edit operations.
► It uses the ISFInkMLConverter library and saves files in the InkML or the ISF formats.
► It uses the following tools: pen, eraser and lasso select (move, resize ink strokes)

Figure 9 demonstrates the drawing of finger movement.

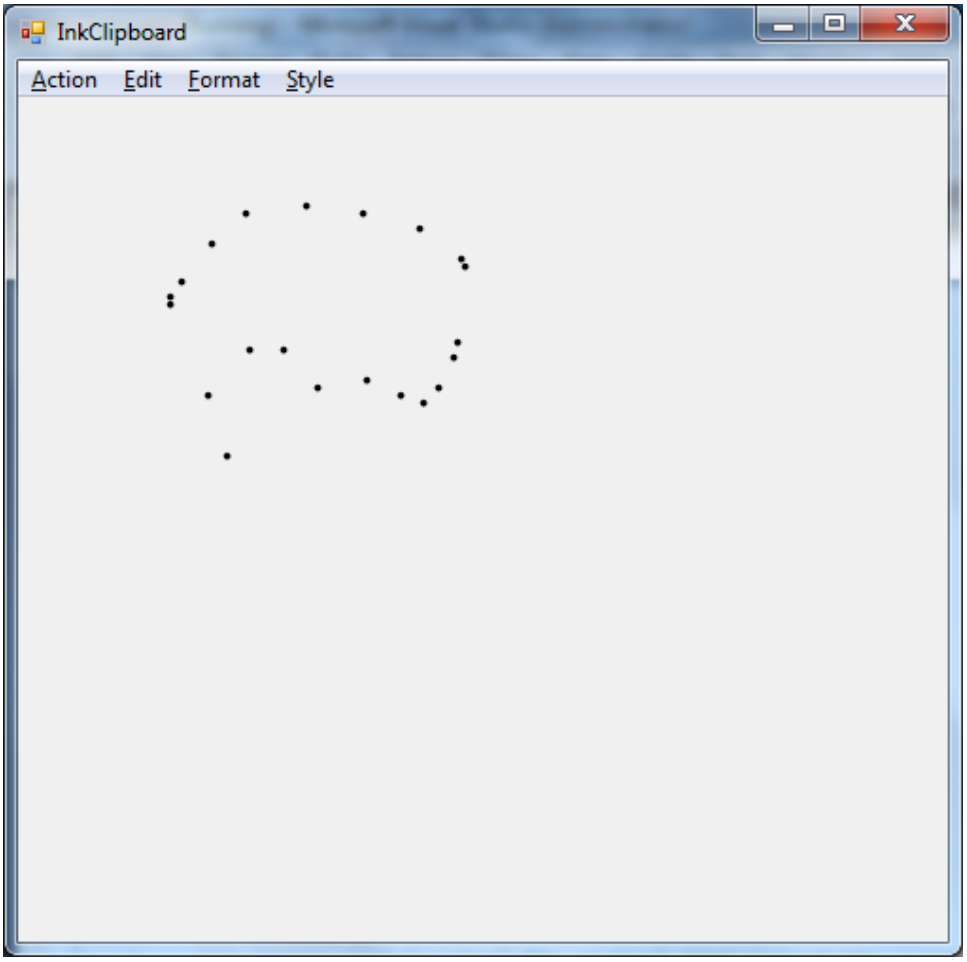

Figure 9: InkMLPAD Application with Sensor Moves

5.1.2 Braille Communication Driver

Based on the data format of the DMD 120060 [24] transmission interface standard, the communication driver module creates a thread to collect all sensor positions through a USB to serial converter. Once a new sensor position is captured, the data are transferred and parsed into coordinates as InkML data and rendered into the InkMLPad application.

The InkMLPAD application is a real-time rendering graphical user interface application to collect all sensor coordinates. We use the sensor coordinates including column and row data as base points. The Braille DMD 120060 [24] has only 60 rows and 120 columns. It is hard to analyse the data visually. We scale the row number by 2 and the column number by 4. To render the finger movement in a window form, we simulate mouse up

and down events with corresponding cursor positions. When the test person stops the finger movement for 5 minutes, the InkMLPad application will generate a graphical XML data file to represent the drawing and send the XML data to the telecom server through the SOAP interface. The XML data will be formatted based on InkML [23].

The DMD 120060 Communication Driver is able to collect two finger positions simultaneously. Based on the sensor identified, two finger movement traces will be recorded.

5.1.3 Client SOAP Interface

Once the InkMLPAD application finishes collecting the input data, it will generate an InkML, store it locally and then send it to the telecom server through the SOAP [34] client interface. SOAP interface provides reliable network communication services.

SOAP[34], originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on the Extensible Markup Language (XML) for its message format, and it usually relies on other Application Layer protocols, most notably the Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission. SOAP is versatile enough to allow for the use of different transport protocols. The standard stacks use HTTP as a transport protocol. Since the SOAP model tunnels define in the HTTP get/response model, it can tunnel easily through existing firewalls and proxies without modification to the SOAP protocol, and it can use the existing infrastructure.

SOAP version 1.2 (SOAP) is a lightweight protocol intended for exchanging structured information in a decentralized distributed environment. It uses XML to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics [35].

The technological foundation that makes up web services includes SOAP, the Web Service Description Language (WSDL), Universal Description, Discovery, and Integration (UDDI), and XML. Specifically, SOAP provides a heterogeneous mechanism to allow the invocation and communication between web services [35]. Microsoft .NET 2.0 supports SOAP 1.2.

The System.Web.Services.Protocols.SoapHttpClientProtocol class that is used as the base class of the generated client proxies has a new property SoapVersion. This is set to SoapProtocolVersion.Soap12 to indicate that SOAP 1.2 should be used [36].

5.1.4 Other Tactile Devices Interface

Other tactile devices can be integrated into the support system very easily through different input interfaces as well. The InkMLPAD application has the ability to parse and map different formats of the coordinates to generate graphical XML data. However, multiple finger movements on the tactile will be much more complicated than DMD 120060.

5.2 Telecom Server

The telecom server handles web messages and stores the data in a backend database. Today, most “graphics for the blind” projects are stand-alone applications, which mean that they will only be able to collect data locally. It is hard to generate profiles for different blind users without data information. We built the telecom server to achieve this goal. The Telecom Server will connect through web services so that it is able to collect information locally and remotely. In this way, we can generate different user group profiles and use the data to train visually impaired people or to support scientific evaluations of the respective behavior. The most likely client candidates include the GrassML client product suite and the Braille device driver integration. The Telecom server is an application server designed to provide web services through an extensible service framework. The primary objectives for this framework include provision for:

- web service operations,
- facilities for rapid service development,

- centralized persistence mechanisms for data objects,
- relational schema and data mappings for all data objects, and
- integrity and security guarantees for all communication

Telecom Server Technical Requirements

- Microsoft .NET 3.5 or greater,
- Support both Microsoft Windows x86 and 64-bit platforms,
- Remain database agnostic: minimal support of the SQL Server and Oracle DBMS,
- Should not require IIS,
- Development in C# and Visual Studio 2005 or greater.

5.2.1 Telecom Server Architecture

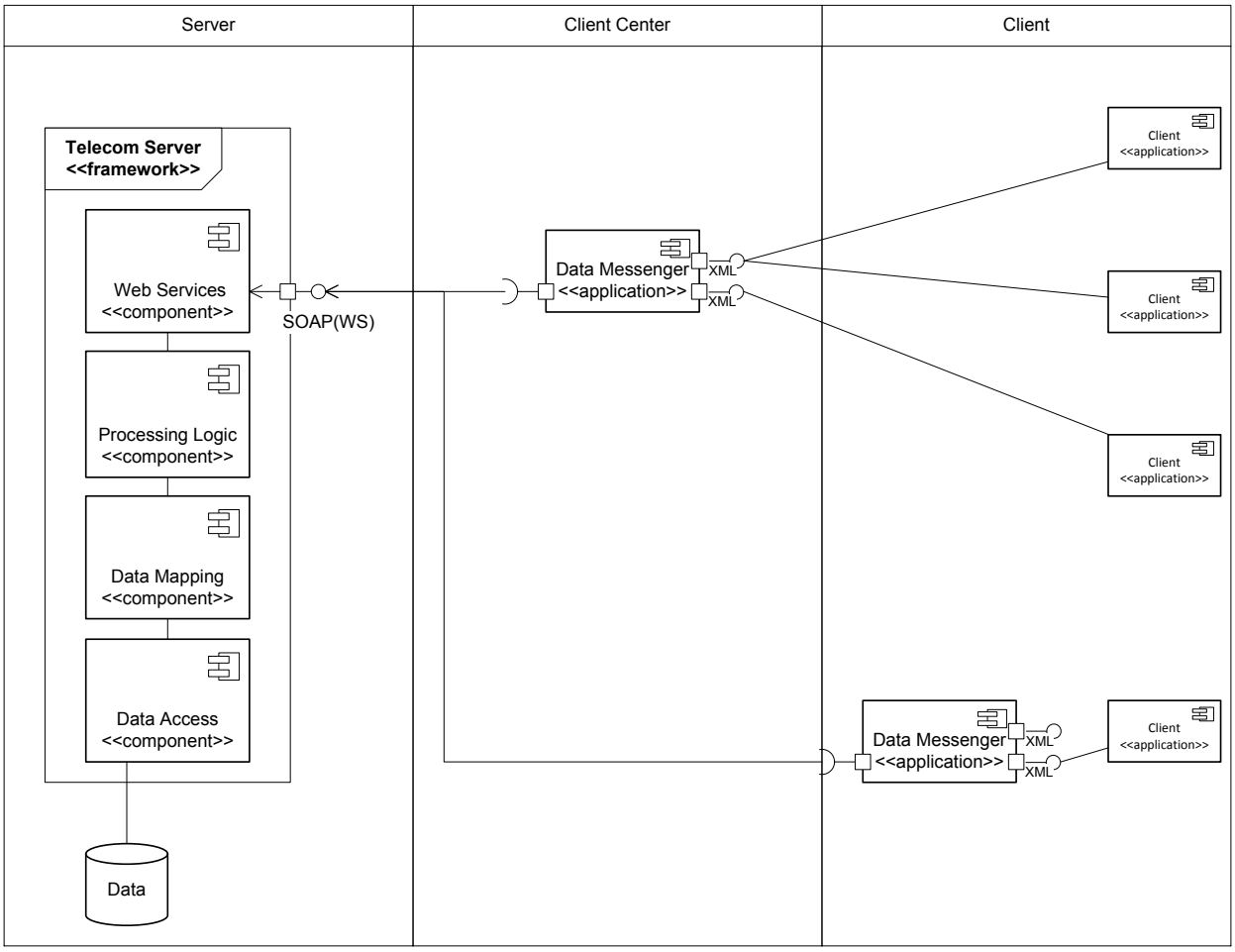

Figure 10: Component Diagram for Telecom Server

Figure 10 depicts the major components of the Telecom server architecture. The Data Messenger application is a separate, lightweight service that can be run from client devices/platforms. This particular client-server design is intended to encourage performance, reliability, security and separation of use.

The Telecom server uses the Windows Communication Foundation (WCF) framework [37], in order to leverage a host of performance, security, scalability and communication benefits. This allows the Telecom server architecture to focus on specific requirements: WCF decouples data contracts, transport and behavior so that the service design does not have to account for message transport or protocol.

5.2.2 Telecom Server Framework

The Telecom Server Framework (TSF) is comprised of a Windows service responsible for instantiating global resource controllers (database, file or network controller services), configuring the environment (establishing logs, loading configuration data, etc) and launching configured services. The TSF service launches a controller thread that is responsible for starting, stopping and monitoring services. Launch conditions defined in the configuration file control the start-up sequence and ensure that dependencies are secured before starting services. The Telecom server is designed to be launched either as a Windows service or from the command-line and depends on a primary configuration file whose default name is "TelcomServer.xml". The configuration for the Telecom server specifies services to launch, service configurations, controller configurations, and data mapping configurations, logging behavior and database connections.

5.2.3 Server Web Services

All web services are defined by a data contract based on an XML schema. The schema defines the composite data types and is processed by Microsoft compilation tools to generate C# class objects for service definitions: this prevents having to duplicate the data contract unnecessarily. Services are defined by an interface, and class implementation defines how the service behaves. The service logic is primarily controlled by the implementing service class.

Services can implement any of the WCF bindings defined in Table 1 (taken from the MSDN WCF reference) [38]. Custom bindings can also be used and the TSF design strategy is to permit such customization for future development.

Binding	Description
BasicHttpBinding	A binding that is suitable for communicating with WS-Basic Profile conformant Web services, for example, ASP.NET Web services (ASMX)-based services. This binding uses HTTP as the transport and text/XML as the default message encoding.
WSHttpBinding	A secure and interoperable binding that is suitable for non-duplex

	service contracts.
WS2007HttpBinding	A secure and interoperable binding that provides support for the correct versions of the Security, ReliableSession, and TransactionFlow binding elements.
WSDualHttpBinding	A secure and interoperable binding that is suitable for duplex service contracts or communication through SOAP intermediaries.
WSFederationHttpBinding	A secure and interoperable binding that supports the WS-Federation protocol, enabling organizations that are in a federation to efficiently authenticate and authorize users.
WS2007FederationHttpBinding	A secure and interoperable binding that derives from WS2007HttpBinding and supports federated security.
NetTcpBinding	A secure and optimized binding suitable for cross-machine communication between WCF applications.
NetNamedPipeBinding	A secure, reliable, optimized binding that is suitable for on-machine communication between WCF applications.
NetMsmqBinding	A queued binding that is suitable for cross-machine communication between WCF applications.
NetPeerTcpBinding	A binding that enables secure, multi-machine communication.
WebHttpBinding	A binding used to configure endpoints for WCF Web services that are exposed through HTTP requests instead of SOAP messages.
MsmqIntegrationBinding	A binding that is suitable for cross-machine communication between a WCF application and existing Message Queuing (also known as MSMQ) applications.

Table 1: WCF Binding

Services are not designed to inter-operate. The intent of a service is to define the behavior for a specific domain model and as such is not typically interactive with other services.

5.2.4 Data Mapping

The WCF framework maps XML data to the appropriate object types and invokes the corresponding service class methods. Objects can then be written to or read from the database by mapping object attributes to corresponding database objects. Consider the following example:

- An XML representation of a sample record of GrassML:

```
<UserId=”1234”>
        <GroupId>0001</GroupId >
        <RecordId>0001</RecordId>
        <RecordDate>31-Jul-12</RecordDate>
        <GrassML>…….</GrassML>
</UserId>
```

The record of GrassML includes the test person user id, group id, record id, record date, and GrassML object.

- A sample database table representation of GrassML:

UserId
GroupId
RecordId
RecordDate
GrassML

The database table includes the above data field to hold the graphical data.

In simple cases, associations between database structures and the data contract can be specified with value pairs such as (UserId, GroupId), etc. The Telecom server architecture provides configuration mechanisms for specifying such mappings and facilities within its framework for assigning any domain model to a *mappable* object quickly. The advantage of this representation is that developers can map objects without being concerned with SQL or relational data semantics, providing a database-agnostic

mechanism for storing and reading database objects. Conceptually, the data mapping idiom could be extended to other devices, including file-based formats on disk.

Many other data mapping technologies exist that perform similar functions. The main advantages to developing an in-house data mapping architecture are the elimination of the dependency on third-party technologies and the freedom to customize the mapping implementation to suit the needs of the Telecom server product. Other implementations include LINQ [39], the Entity Framework and NHibernate [40].

When we transfer XML elements from the client to the server, we use CDATA to avoid the XML parser. Characters like "<" and "&" are illegal in XML elements. In an XML document or external parsed entity, a CDATA section is a section of element content that is marked for the parser to interpret as only character data, not markup. A CDATA section is merely an alternative syntax for expressing character data; there is no semantic difference between character data that manifests in a CDATA section and character data that manifests as in the usual syntax where "<" and "&" would be represented by "<" and "&", respectively [41].

5.2.5 Data Messenger

The data messenger is a standalone service tightly coupled to the server. The messenger facilitates the transmission of data to the server by taking full advantage of the WCF framework and insulates the client center domain model from data representations used by the server. The messenger takes advantage of the WCF framework for session management, data integrity, security and performance and leverages the same ORM implementation as the server, except messenger mappings occur between C# objects rather than a C# object and a database object.

Chapter 6

6 Demonstrations of the System

The following figures were captured by the GrassML client application in different scenarios. Figure 11 and 12 use the original coordinates captured from the tactile display DMD 120060. The GrassML client serial port driver will capture the finger movement data from Braille DMD 120060. Finger position data will be rendered as points in the application window simultaneously. The default screen width is 500, and the height is 472. The dots on the Braille are 120 by 60. The finger traces in Figure 11 and 12 are relatively small in the window. The shape of the drawing is hard to recognize by sight. Figure 11 and 12 also move in two different speeds. The speed of the finger movement could be computed from DMD 120060 scanning rate or a timestamp tag in the graphical xml data. Due to the low scanning speed, the drawing in Figure 11 is faster than Figure 12. If we use a high scanning speed Braille device, the drawings in Figures 11 and 12 are likely to have more similarity using the same finger movements. Currently, timing data are not available.

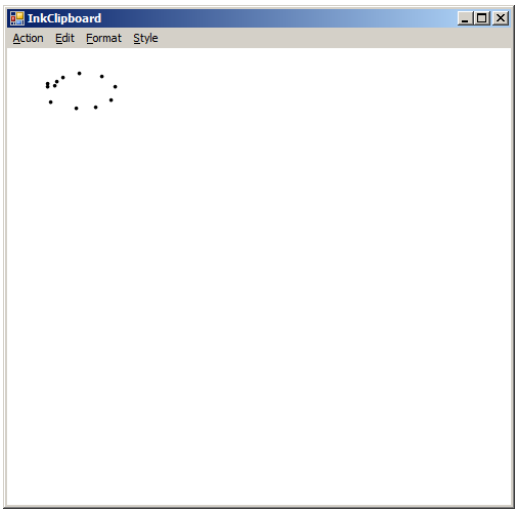

Figure 11: Sensor Data Input With Fast Finger Movement

InkML Example for Fast Finger Movement:

```
<ink>
 <definitions>
  <traceFormat id="tf0">
   <channel name="X" type="DECIMAL" max="1920" min="0" orientation="+ve" units="inch" />
   <channel name="Y" type="DECIMAL" max="1080" min="0" orientation="+ve" units="inch" />
  </traceFormat>
  <brush id="br1">
   <annotationXML>
    <color>Black</color>
    <width>53</width>
    <transparency>0</transparency>
    <antialiased>True</antialiased>
    <fittocurve>False</fittocurve>
    <height>1</height>
    <ignorePressure>False</ignorePressure>
    <pentip>Ball</pentip>
   </annotationXML>
  </brush>
  <brush id="br2">
   <annotationXML>
    <color>Black</color>
    <width>100</width>
    <transparency>0</transparency>
    <antialiased>True</antialiased>
    <fittocurve>False</fittocurve>
    <height>1</height>
    <ignorePressure>False</ignorePressure>
    <pentip>Ball</pentip>
   </annotationXML>
  </brush>
  <context id="ctx3" brushRef="#br2" traceFormatRef="#tf0" />
 </definitions>
 <trace id="T1" brushRef="#br2" contextRef="#ctx3">0.479133858267717 0.375196850393701 </trace>
 <trace id="T2" brushRef="#br2" contextRef="#ctx3">0.5 0.333464566929134 </trace>
 <trace id="T3" brushRef="#br2" contextRef="#ctx3">0.56259842519685 0.291732283464567 </trace>
 <trace id="T4" brushRef="#br2" contextRef="#ctx3">0.729133858267717 0.25 </trace>
 <trace id="T5" brushRef="#br2" contextRef="#ctx3">0.958267716535433 0.281102362204724 </trace>
 <trace id="T6" brushRef="#br2" contextRef="#ctx3">1.09370078740157 0.385433070866142 </trace>
 <trace id="T7" brushRef="#br2" contextRef="#ctx3">1.05196850393701 0.520866141732284 </trace>
 <trace id="T8" brushRef="#br2" contextRef="#ctx3">0.895669291338583 0.593700787401575 </trace>
 <trace id="T9" brushRef="#br2" contextRef="#ctx3">0.698031496062992 0.604330708661417 </trace>
 <trace id="T10" brushRef="#br2" contextRef="#ctx3">0.43740157480315 0.541732283464567 </trace>
 <trace id="T11" brushRef="#br2" contextRef="#ctx3">0.406299212598425 0.385433070866142
</trace>
 <trace id="T12" brushRef="#br2" contextRef="#ctx3">0.406299212598425 0.354330708661417
</trace>
</ink>
```

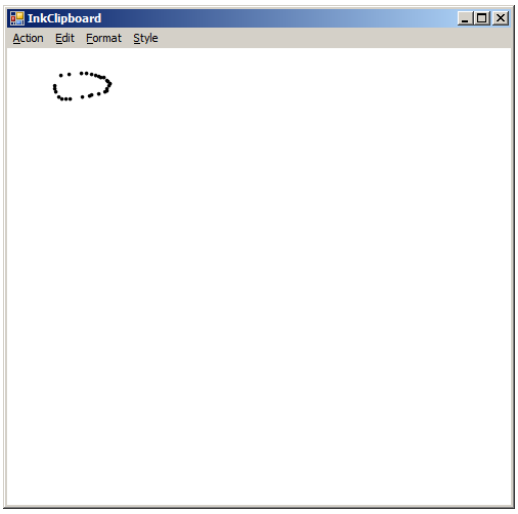

Figure 12: Sensor Data Input With Slow Finger Movement

InkML Example for Slow Finger Movement:

```
<ink>
 <definitions>
  ……….
 </definitions>
 <trace id="T9" brushRef="#br2" contextRef="#ctx3">0.541732283464567 0.270866141732283 </trace>
 <trace id="T10" brushRef="#br2" contextRef="#ctx3">0.625196850393701 0.260236220472441
</trace>
 <trace id="T11" brushRef="#br2" contextRef="#ctx3">0.75 0.25 </trace>
 <trace id="T12" brushRef="#br2" contextRef="#ctx3">0.801968503937008 0.25 </trace>
 <trace id="T13" brushRef="#br2" contextRef="#ctx3">0.854330708661417 0.260236220472441
</trace>
 <trace id="T14" brushRef="#br2" contextRef="#ctx3">0.895669291338583 0.270866141732283
</trace>
 <trace id="T15" brushRef="#br2" contextRef="#ctx3">0.927165354330709 0.281102362204724
</trace>
 <trace id="T16" brushRef="#br2" contextRef="#ctx3">0.948031496062992 0.291732283464567
</trace>
 <trace id="T17" brushRef="#br2" contextRef="#ctx3">0.979133858267717 0.291732283464567
</trace>
 <trace id="T18" brushRef="#br2" contextRef="#ctx3">1.01023622047244 0.322834645669291 </trace>
 <trace id="T19" brushRef="#br2" contextRef="#ctx3">1.02086614173228 0.333464566929134 </trace>
 <trace id="T20" brushRef="#br2" contextRef="#ctx3">1.04173228346457 0.354330708661417 </trace>
 <trace id="T21" brushRef="#br2" contextRef="#ctx3">1.03110236220472 0.375196850393701 </trace>
 <trace id="T22" brushRef="#br2" contextRef="#ctx3">1.01023622047244 0.406299212598425 </trace>
 <trace id="T23" brushRef="#br2" contextRef="#ctx3">1.01023622047244 0.427165354330709 </trace>
 <trace id="T24" brushRef="#br2" contextRef="#ctx3">0.989763779527559 0.43740157480315 </trace>
 <trace id="T25" brushRef="#br2" contextRef="#ctx3">0.927165354330709 0.458267716535433
</trace>
 <trace id="T26" brushRef="#br2" contextRef="#ctx3">0.854330708661417 0.468897637795276
</trace>
 <trace id="T27" brushRef="#br2" contextRef="#ctx3">0.833464566929134 0.479133858267717
</trace>
 <trace id="T28" brushRef="#br2" contextRef="#ctx3">0.760236220472441 0.489763779527559
</trace>
 <trace id="T29" brushRef="#br2" contextRef="#ctx3">0.635433070866142 0.510236220472441
</trace>
```

<trace id="T30" brushRef="#br2" contextRef="#ctx3">0.593700787401575 0.510236220472441
</trace>
<trace id="T31" brushRef="#br2" contextRef="#ctx3">0.551968503937008 0.510236220472441
</trace>
<trace id="T32" brushRef="#br2" contextRef="#ctx3">0.520866141732284 0.489763779527559
</trace>
<trace id="T33" brushRef="#br2" contextRef="#ctx3">0.489763779527559 0.43740157480315 </trace>
<trace id="T34" brushRef="#br2" contextRef="#ctx3">0.479133858267717 0.406299212598425
</trace>
<trace id="T35" brushRef="#br2" contextRef="#ctx3">0.479133858267717 0.375196850393701
</trace>
</ink>

To get better visual quality, figures 13 and 14 use the coordinates multiplied by 2. The figures are larger than the original ones. However, due to the slow scanning speed, the drawing has fewer points than fast scanning rate devices.

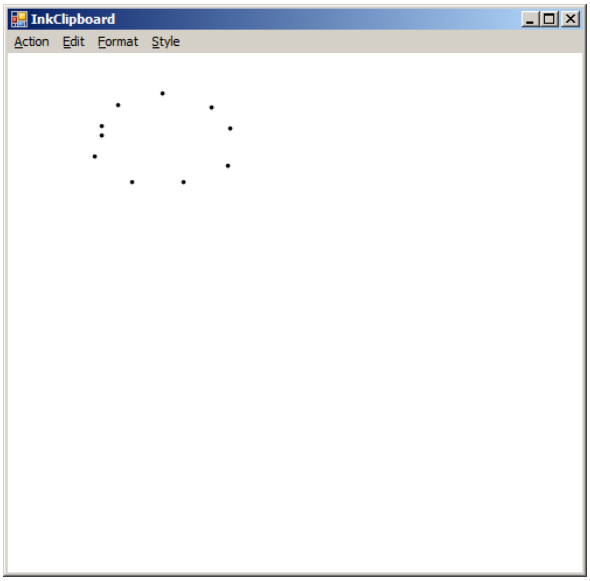

Figure 13: Sensor Data Input Scaled By 2 with Fast Finger Movement

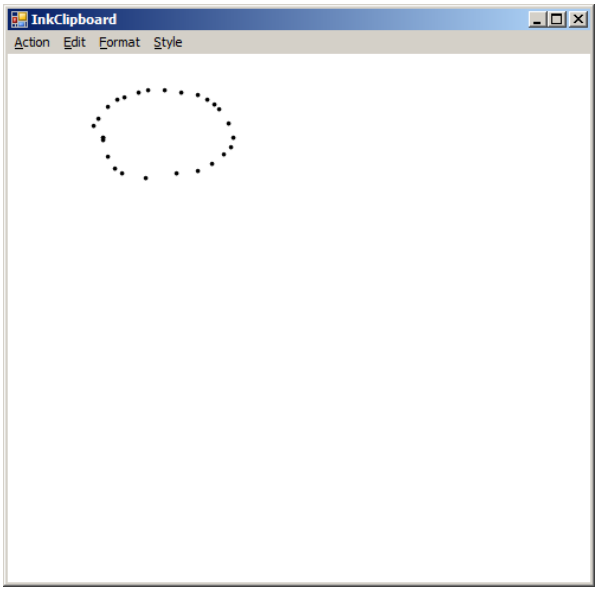

Figure 14: Sensor Data Input Scaled By 2 with Slow Finger Movement

Figures 15 and 16 use the coordinates multiplied by 4. The figures are much larger than the previous ones. Assuming the scan rate of the tactile display DMD 120060 is consistent, Figures 11, 13 and 15 have the same figure movement speed. Figure 15 will give us the characteristics of the finger movement visually in details. Due to the tactile display hardware limitation, it is preferable to use InkML to represent the graphics.

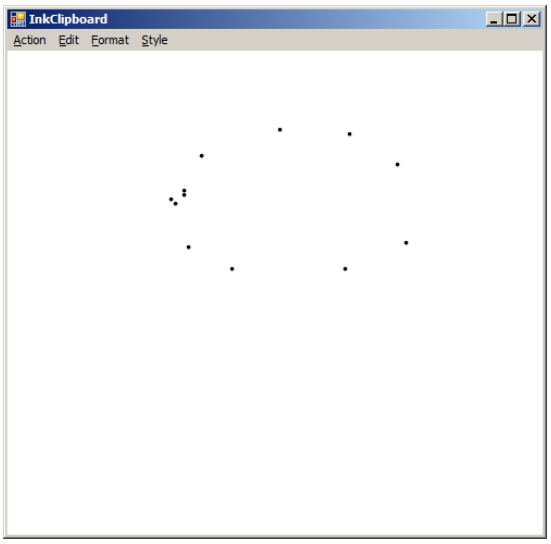

Figure 15: Sensor Data Input Scaled By 4 with Fast Finger Movement

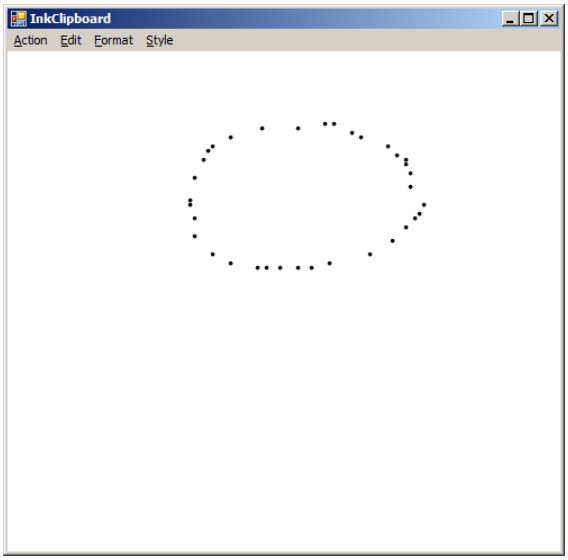

Figure 16: Sensor Data Input Scaled By 4 with Slow Finger Movement

In the future, if we have fast input tactile display, the above graphics will be represented with more points. We can use SVG [4] or anther graphical XML to represent the graphics and to make the graphics more accessible to different systems.

The Telecom server console will display the server status, database operation and exception handling. Figure 17 demonstrates the telecom server status. Due to the large

amount of XML graphical data, the XML data can be encoded. When the data are retrieved from the database, they will be decoded.

Figure 17 Telecom Server Console

Chapter 7

7 Summary

The support system mainly collects raw data from the visually impaired people and generates semantic graphics data. That is rarely done by research groups. Most research projects use the graphics for sighted people, retrieve the information and present it to the visually impaired people. Our support system has the ability to collect the graphics in the creation stage. Semantic graphics will use XML data to represent the graphics, which makes graphics more accessible to people with different needs. Visually impaired people will draw graphics on the haptic device through finger movement. Semantic graphics XML data will be transferred and stored in a remote database through web services. The system will facilitate remote data collection at many different locations.

This support system can be used for many different purposes. We can use the support system to generate semantic graphics to make the graphics accessible to different visually impaired people. The semantics graphics data can train visually impaired people to draw the graphics by using the recorded graphics as a model based on user profiles.

In a future project, the graphical XML data can be used to find common gesture behaviors and scenarios from different groups of users for training and educational purposes. Voice recognition and other assistive technologies can be integrated into the system to make the data more meaningful. The data collection can be processed to generate different user group profiles. People will be able to learn the graphics from similar user profiles.

References

1. Tufte, E.: Visual Explanations: Images and Quantities, Evidence and Narrative. Graphics Press, Cheshire (1997) ISBN 0961392126
2. Wendy Chisholm, Gregg Vanderheiden, Ian Jacobs (editors): Web Content Accessibility Guidelines 1.0, W3C, 1999. http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/ (2012). Accessed Nov 2012
3. WAVE: Web accessibility evaluation tool. http://wave.webaim.org/ (2012). Accessed Nov 2012
4. Erik Dahlström, Patrick Dengler, Anthony Grasso, Chris Lilley, Cameron McCormack, Doug Schepers, Jonathan Watt: Scalable Vector Graphics (SVG) 1.1 Specification. http://www.w3.org/TR/2011/REC-SVG11-20110816/ Publisher: W3C (2011), Retrieved Jul, 2012
5. Rotard, M., Ertl, T.: Tactile access to scalable vector graphics for people with visual impairment. In: SVG Open Conference, Tokyo, Japan, September 7-10, 2004, 3rd Annual Conference on Scalable Vector Graphics, Proceedings. http://www.svgopen.org/2004/papers/TactileAccessToSVG/ (2004). Accessed Nov 2012
6. Z. Ben Fredj, D. A. Duce: GraSSML: Smart Schematic Diagrams, XML and Accessibility. In: M. McDerby, L. Lever (Editors), EG UK Theory and Practice of Computer Graphics, 2006, pp. 47-54. Published in: Proceeding, W4A '06 Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility? Pages 57 – 60, ACM, 2008.

7. Z. Ben Fredj, D. A. Duce: GraSSML: accessible smart schematic diagrams for all. Universal Access in the Information Society Journal (2007) 6:233–247
8. Brown, A., Stevens, R., Pettifer, S.: Issues in the nonvisual presentation of graph based diagrams. In: Eighth International Conference on Information Visualisation, IV 2004, 14-16 July 2004, London, UK, Proceedings, pp. 671–676, IEEE Computer Society, 2004
9. Jürgensen, H.: Tactile computer graphics. Manuscript, 48 pp. (1996)
10. Power, C., Jurgensen, H.: Accessible presentation of information for people with visual disabilities. In: Universal Access in the Information Society Journal 9, 97–119 (2010)
11. Brown, A., Stevens, R., Pettifer, S.: Issues in the nonvisual presentation of graph based diagrams. In: Eighth International Conference on Information Visualisation, IV 2004, 14-16 July 2004, London, UK, Proceedings, pp. 671–676, IEEE Computer Society, 2004
12. Way, T.P., Barner, K.E.: Automatic visual to tactile translation—part II: human factors, access methods, and image manipulation. IEEE Transactions on Rehabilitation Engineering. 5(1), 81–94 (1997)
13. Horstmann, M., Hagen, C., King, A., Dijkstra, S., Crombie, D., Evans, G., Ioannidis, G. T., Blenkhorn, P., Herzog, O., and Schlieder, C.: TeDUB: Automatic interpretation and presentation of technical diagrams for blind people. In: Hersh, M. (Editor): *Conference and Workshop on Assistive Technologies for Vision and Hearing Impairment (CVHI,EURO-ASSIST-VHI-3),29th June – 2nd July, 2004*, Granada, Spain, Proceedings. Published on CD.

14. Landau, S.: Tactile graphics and strategies for non-visual seeing. Thresholds 19, 78-82 (1999)
15. Tiger Braille Technology: View Plus Company Website. http://www.viewplus.com/solutions/Braille-technology/(2012). Accessed on Oct 2012
16. Landau, S., Gourgey, K.: Development of a talking tactile tablet. Information Technology and Disabilities Vol. VII 2001. http://people.rit.edu/easi/itd/itdv07.htm (2001)
17. Viewplus technologies. Online Product Web Site Retrieved from http://www.viewplustech.com/ (2002)
18. Poh, S.-P.: Talking diagrams. Master's thesis, The University of Western Ontario, 1995. Also technical report 459, Department of Computer Science, The University of Western Ontario (1995)
19. B. Breidegard, Y. Eriksson, K. Fellenius, B. Jönsson, K. Holmqvist, S. Strömqvist: Enlightened: The Art of Finger Reading, Studia Linguistica 62 (3) 249 -260, 2008
20. David Chappell, Chappell & Associates: Introducing the .NET Framework 3.5. Introducing_NET_Framework_35_v1.doc. Web. Accessed Jul 2012
21. SQL Tutorial. W3 schools, http://www.w3schools.com/sql/sql_intro.asp. Accessed Aug 2012
22. TTT: Talking Tactile Tablet 2 on Touch Graphics Company Website. http://touchgraphics.com/OnlineStore/index.php/featured-products/talking-tactile-tablet-2-ttt.html (2012). Accessed Oct 2012

23. Stephen M. Watt, Tom Underhill (editors): Ink Markup Language Specification (InkML). World Wide Web Consortium. W3C, 2011. http://www.w3.org/TR/2011/REC-InkML-20110920/. Accessed Aug 2012
24. Tactile Display DMD 120060 user manual (1992), Translated by Helmut Jurgensen
25. Völkel, T., Weber, G., Baumann, U.: Tactile Graphics Revised: The Novel BrailleDis 9000 Pin-Matrix Device with Multitouch Input. In Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, Arthur I. Karshmer, editors, Computers Helping People with Special Needs, 11th International Conference, ICCHP 2008, Linz, Austria, July 9-11, 2008. Proceedings. Volume 5105 of Lecture Notes in Computer Science, pages 835-842, Springer, 2008.
26. Hyperbraille project. Online Project Web Site Retrieved from http://www.hyperbraille.de/ (2012)
27. Touch Graphics Technologies. Online Product Web Site Retrieved from http://www.touchgraphics.com/research/ttt.htm (2012)
28. Gray, E.:, Telautograph, United States Patent 386,815, 31 July, 1888
29. Stumpe, B., Sutton,C.: The first capacitative touch screens at CERN, 31 March 2010 Retrieved from http://cerncourier.com/cws/article/cern/42092 Nov, 2012
30. Editors PC Magazine. "Definition of: tablet computer". PC Magazine. Retrieved from http://www.pcmag.com/encyclopedia_term/0,2542,t=tablet+computer&i=52520,00.asp Nov, 2012.
31. .Net Speech Recognition Technology: Microsoft Company Website. http://msdn.microsoft.com/en-us/library/hh361683(v=office.14).aspx (2012). Accessed Nov 2012

32. Shape Recognition – Myscript technology: VisionObject Company Website. http://www.visionobjects.com/en/myscript/about-myscript/myscript-technology/description/ (2012). Accessed Nov 2012

33. InkMLPad: InkML applications. http://sourceforge.net/apps/trac/inkmltk/wiki/InkMLPad (2012). Accessed Sep 2012

34. SOAP. http://en.wikipedia.org/wiki/SOAP. Accessed Sep 2012

35. SOAP .NET: SOAP 1.2 in .NET Framework 2.0 Introduction http://www.codeproject.com/Articles/11878/SOAP-1-2-in-NET-Framework-2-0 (2012). Accessed Sep 2012

36. SOAP web technology: SoapHttpClientProtocol.BeginInvoke Method Introduction. http://msdn.microsoft.com/en-us/library/system.web.services.protocols.soaphttpclientprotocol.aspx (2012). Accessed Sep 2012

37. Juval Lowy: Programming WCF Services, 2nd Edition. Published in O'Reilly Media 2008

38. WCF Binding: Configuring System-Provided Bindings Introduction. http://msdn.microsoft.com/en-us/library/ms731092.aspx (2012). Accessed Sep 2012

39. Charlie C., Dinesh K.: Essential LINQ, Chapter 1 Introduction, pp. 1-17. Published in Addison-Wesley Professional; 1 edition (March 22, 2009).

40. Dan T.: Introduction to NHibernate Tutorial. http://danny-t.co.uk/tutorials/introduction-to-nhibernate-tutorial/. Web. Accessed Oct 2012

41. CDATA: Character date Introduction. http://en.wikipedia.org/wiki/CDATA(2012). Accessed Oct 2012

Appendices

Appendix A: Sample Source Code for Serial Port Communication

```
void com_DataReceived(object sender,
   SerialDataReceivedEventArgs e)
{
    try
    {
        byte numPortByte = 0;
        while (com.BytesToRead > 0)
        {
            numPortByte = (byte)com.ReadByte();
            if (numPortByte == STX)
                m_blnFirstByte = true;
            else if (numPortByte == CR && m_blnFirstByte)
            {
                m_lstByteBuffer.Clear();
                m_numScanStart = 0;
                m_blnFirstByte = false;
            }
            else if (numPortByte == CR && !m_blnFirstByte)
            {
                m_numScanStart++;
            }
            if (numPortByte != STX && numPortByte != CR)
                m_lstByteBuffer.Add(numPortByte);
            if (m_numScanStart == 2)
            {
                if (numPortByte == CR)
                {
                    if (m_lstByteBuffer.Count != 0)
                        ProcessPositionData(m_lstByteBuffer);
                    m_lstByteBuffer.Clear();
                }
                m_numScanStart = 0;
            }
        }
    }
    catch (Exception ex)
    {
    }
}

private void ProcessPositionData(List<byte> lstByteBuffer)
{
    while (lstByteBuffer.Count > 0)
    {
        string strBuffer = string.Empty;
        if (lstByteBuffer.Count >= 7)
            strBuffer = Encoding.ASCII.GetString(lstByteBuffer.ToArray(), 0, 7);
        if (strBuffer.Length == 7)
        {
            int numRow = int.Parse(strBuffer.Substring(0, 2));
            int numSensor = int.Parse(strBuffer.Substring(3, 1));
            int numCol = int.Parse(strBuffer.Substring(4, 3));
```

```
            if (!m_dictCoordinate.ContainsKey(numSensor))
            {
                List<KeyValuePair<int, int>> lstKeyPair = new
List<KeyValuePair<int, int>>();
                lstKeyPair.Add(new KeyValuePair<int, int>(numRow, numCol));
                m_dictCoordinate.Add(numSensor, lstKeyPair);
                DoMouseClick((uint)numCol, (uint)numRow);
            }
            else
            {
                int numCoordinate = m_dictCoordinate[numSensor].Count;
                if (m_dictCoordinate[numSensor][numCoordinate - 1].Key != numRow
||
                    m_dictCoordinate[numSensor][numCoordinate - 1].Value !=
numCol)
                {
                    m_dictCoordinate[numSensor].Add(new KeyValuePair<int,
int>(numRow, numCol));
                    DoMouseClick((uint)numCol, (uint)numRow);
                }
            }
            lstByteBuffer.RemoveRange(0, 7);
        }
    }
}
```

Appendix B: Sample Source Code for Telecom Server Database Operation

```
public void Add(GraphicsDataType graphics)
{
    try
    {
        log.Debug("Add() called with graphics object.");
        string sqlGraphics = "insert into grassml_records values (" +
graphics.userId + "," + graphics.groupId + "," +
            graphics.recordId + ",to_date('" + graphics.recordDateId +
"','mm/dd/yyyy') " +  "," + " '" + graphics.grassML.Replace("'","''") + "') ";
        int gxmlstringlength = graphics.grassML.Length;

        string compressedString = CompressString(graphics.grassML);
        int compressedStringLength = compressedString.Length;

        string deCompressedString = DecompressString(compressedString);
        int deCompressedStringLength = deCompressedString.Length;

        string sqlGraphicsDec = "insert into grassml_records values (" +
graphics.userId + "," + graphics.groupId + "," +
            graphics.recordId + ",to_date('" + graphics.recordDateId +
"','mm/dd/yyyy') " + "," + " '" + compressedString + "') ";
        using (DbDataReader reader = GetReader(sqlGraphicsDec))
        {
            if (reader.RecordsAffected != 1)
            {
                throw new Exception("The database reported no rows affected during
the last operation.  Check database and server logs for details.");
            }
        }
```

```
    }
    catch (Exception e)
    {
        log.Error("Failed to add the graphics to the database.", e);
        throw;
    }
    log.Debug("Add() succeeded.");
}


public DbDataReader GetReader(string query)
{
    DbConnection connection = ConnectionFactory.GetConnection("GraphicsXml");
    connection.Open();
    log.Debug("Issuing query: " + query);
    DbCommand command = connection.CreateCommand();
    command.CommandText = query;
    command.Connection = connection;
    DbDataReader reader =
command.ExecuteReader(System.Data.CommandBehavior.CloseConnection);
    return reader;
}
```

Appendix C: Sample Source Code for SQL Database Script

CREATE TABLE grassml_records (
user_id NUMBER(4),
group_id NUMBER(4),
record_id NUMBER(4),
record_date DATE,
graphics_data XMLTYPE); [23] }

The SQL script create GrassML data table when graphics_data use an XML data type.

CREATE TABLE grassml_records (
user_id NUMBER(4),
group_id NUMBER(4),
record_id NUMBER(4),
record_date DATE,
graphics_data CLOB); [24] }

The SQL script creates grassml data table when graphics_data use CLOB data type to hold large data.

insert into grassml_records values
(0001,0001,0001,'08/07/2012',XMLType('<inkml><brush>yellow</brush></inkml>')) [23]

The SQL script inserts GrassML data into the table.

Curriculum Vitae

Name:	Hao Xu
Post-secondary Education and Degrees:	Wuhan University of Hydraulic and Electrical Engineering Wuhan, HuBei, China 1996-2000 B.ENG. The University of Western Ontario London, Ontario, Canada 2005-2008 B.Sc.
Honours and Awards:	NSERC Undergraduate Student Research Award (2008)
Job Experience:	Software Developer at Honeywell Canada (2006 – 2007) Software Researcher/Developer at University of Potsdam (2008) Software Developer at RedIron Technology (2008 – 2011) System Engineer at ESAC (2012)