The 12th Winona Computer Science Undergraduate Research Symposium

April 25, 2012

11:00am to 1:00pm

Watkins 108

Winona State University
Winona, MN

Sponsored by the Department of Computer Science
at Winona State University

WINONA
STATE UNIVERSITY
Computer Science Department
http://cs.winona.edu

Table of Contents

GPS Automated Tractors

Ben Mahoney
Department of Computer Science, Winona State University
175 W Mark Street
Winona, MN 55987
1-507-457-5000
BMahoney08@winona.edu

ABSTRACT

There is not much known about the technology of GPS automated tractors. They can be very beneficial to a farmer in many ways. The goal of this paper is to survey the studies and experiments that others have performed on GPS automated tractors. From this, we can determine how beneficial they are to a farmer and in what ways and which technology proves to be the best GPS system. The field of GPS automation has improved greatly over the past couple of decades. The latest technology of kinematic mechanics (RTK-GPS) has made GPS automation very precise when compared to previously used GPS systems. Overall, it has been found that GPS automated tractors have made field work much less labor intensive.

Keywords

Automation; Real Time Kinematics; GPS; Differential GPS; Deviations

1. INTRODUCTION

Agriculture is typically a topic people do not relate to computer science. There have been many advancements made with the aid of computer technology in the agriculture field. One of the areas that have been affected is tractors and the equipment involved in field work. Traditionally, farmers would use hand eye coordination while doing field work with a tractor. As the global population grows and the agricultural field has become more competitive, people resort to higher technologies to improve production and monetary gains. The idea of automated tractors began in 1979 by having a tractor follow a buried wire [11]. Since then, Global Positioning Systems have given this idea of automated tractors a lot more success.

Global Positioning Systems, or GPSs, are commonly used in many different fields of technology. Nowadays, most people are in possession of a GPS. Many cell phones now contain a GPS and can be tracked. Many drivers on the highway often rely on one in their car to travel to the correct destination. The use of GPS has expanded to many different areas. The use of GPSs in agriculture has been expanding the past few decades with the improvement and enhancement of the technology.

Proceedings of the 12th Winona Computer Science Undergraduate Research Seminar, April 25, 2012, Winona, MN, US.

With the addition of GPSs in agriculture, it has helped to make crop farming easier by reducing the amount of labor. Farmers can also minimize the amount of chemicals and materials needed for crop production while maximizing their profits. Currently the technology most used auto aligns the tractor to run in the correct direction. Farmers must still assist tractors in making turns at the end of each pass in the field. It has started to become popular among farmers that run a large amount of acreage that involve row crops [3]. Farmers are allowed to work more hours being that GPS automation makes field work less labor intensive. On top of that, less skilled drivers are needed to complete the job.

The goal of this survey is to determine that GPS automation is truly beneficial. There have been many studies conducted on the algorithms and techniques used in GPS automation. Being that we do not have enough resources available to us, we surveyed the testing done in other studies to help determine the benefits and preciseness of GPS automated tractors. There has been a range of methods and technology used in the testing done in these studies including the new Real Time Kinematics systems (RTK) and the older Differential Global Positioning Systems (DGPS) [3].

Global positioning systems use satellites to determine the location of where they are at. The accuracy of this is increased by the number of satellites used as well as how fast the algorithms run to check for position.

Before we look at some of the experiments surveyed, we will first define some key variables in these experiments. Typically, these experiments used RTK (Real Time Kinematic) GPS, DGPS (Differential Global Positioning System), or a combination of two (RTK-DGPS) [1]. CP-GPS (Carrier Phase Global Positioning System) is also seen in these experiments. CP-GPS was the early stages of what we now know as RTK-GPS [1]. For a DGPS to work, it needs a minimum of three satellites and a ground station to connect to. RTK systems require a minimum of five satellites for initial connection and a ground station [1]. Ground stations are required for both of these systems to have a stable ground location. DGPS can use a correction service provider or radio beacon broadcasts as a ground station. However, RTK-GPS requires that you own your own ground station that is within ten kilometers of where you are working [1]. RTK provides a higher accuracy in three dimensions, whereas DGPS is less accurate and is only two dimensional [1]. Many of the experiments performed measure their results in terms of deviation. Deviation is the measured difference from where the vehicle traveled from where it was expected to travel.

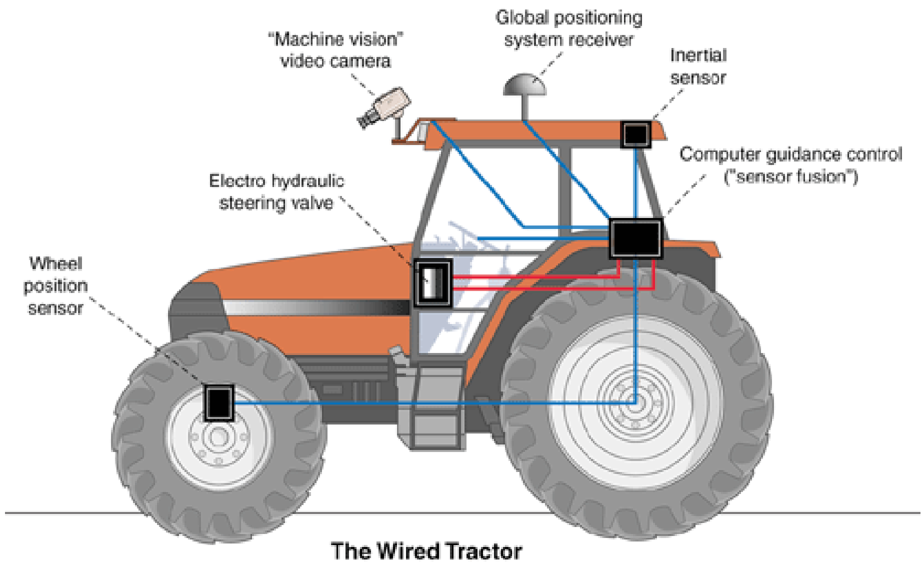


Figure 1. Equipment involved in GPS Tractor Automation [8]

The technology has developed quickly over the years and has become much more advanced and precise. Originally, GPS tracking in tractors was very imprecise which is why it wasn't used more. One of the biggest additions that made it more precise was kinematic mechanics. With this, the preciseness of tracking has been made to within a few centimeters, given optimal conditions [3]. It also can take in some of the affect of wheel slip, and roll that can occur while traversing a field. Wheel slip deals with the loss of traction in a field, which causes the wheels to spin. Roll of a tractor refers to the effect of driving on a hillside, causing the tractor to be unlevel. This paper shows how GPS automation is done and why it is very beneficial. It has been found that GPS automation in tractors has been highly beneficial. It has made its greatest impact in row cropping being it has been made very accurate when driving the rows. Overall, it greatly reduces the stress and intensity of labor found in driving tractors for long hours.

2. HYPOTHESIS

The technology of GPS automation in tractors has allowed farmers to become more productive by cutting costs and labor.

3. METHODS

This project was done by use of the survey methodology. Given that we did not have the resources needed to conduct the needed experiments to exhibit the use of GPS automation, we studied the work of others that has been done on this topic. There have been several studies and experiments done on this issue, or related to it.

First, we looked at the technologies needed in GPS automation, whether they are in current use or past use. We considered what can affect the precision of GPS automation, being that precision is often the measureable of these experiments. It is critical that we establish the developments that have been made in the field and what has been improved. These improvements of GPS technologies are proven by the experiments surveyed. We summarized these experiments and compared them to each other. Through this, differences can be seen between the different types of systems. This should go to show which GPS automation systems are the best at the present moment.

Then, we concluded with an analysis of these experiments as well as explaining the barriers that still exist in GPS automation. Due to these barriers, there hasn't been a large use of automation systems yet.

4. RESULTS

Figure 1 illustrates some of the equipment involved. The GPS receiver feeds coordinates to the computer guidance control [8]. Based off of these coordinates, the computer will correct the tractor to whichever coordinates it needs to be. The electro hydraulic steering valve turns the steering wheel based on what it is told from the computer [8]. The wheel position sensor detects the wheel angle, thus letting the computer know how much it needs to turn the wheel. The inertia sensor will feed the computer information such as roll and yaw, which can affect the tractors true position being that the GPS receiver is on the roof as opposed to the ground [8].

Table 1 summarizes the data collected in the experiments surveyed. Although they all may not look at the same type of data, they give good data on the GPS system that was used. The experiments that excelled the most were the ones that used the newer RTK-GPS or RTK-DGPS. This was as expected being that

Table1. Experiments Surveyed

Institute	Sensor/GPS System Used	Machine/Implement Used	Surface Type	Test Results	Reference
Standford University	CP-DGPS	1984 Yamaha Fleetmaster (golf cart)	smooth, flat	Data collected from 10 trials of following a straight path, standard deviation of 5 cm	O'Connor [7]
University of Georgia	RTK-DGPS	Tractor	pavement, flat	Lateral deviations of the rear axle were from 2.09 cm to 2.35 cm depending on speed	Gan-Mor [4]
University of Georgia	RTK-DGPS	Tractor	rough terrain, gravel	Lateral deviations of the rear axle were from 3.64 cm to 4.05 cm depending on speed	Gan-Mor [4]
University of Valladolid	RTK-GPS	John Deere Model 6400	grass, flat	Circular arcs and straight lines followed, experimental route followed simulated route	Gomez-Gil [5]
University of Illinois	Kinematic Differential GPS	Case-IH Model 7220	flat surface, soil	Deviation of steering angle frequency range was 0.04 Hz to 0.5 Hz	Stombaugh [9]
Technische Universitaet Muenchen, Germany	DGPS, BUS-System	Tractor Fendt Favorit 714 Vario	soil	Actual tillage done, cultivated area was 113% of field size	Demmel [2]
Lasmea and Cemagref, France	CP-DGPS	Tractor	Soil, flat, rough	Difficulty following curved paths, sometimes with a derivation up to 50 cm.	Thuilot [10]

the Real Time Kinematic system is much more precise and newer. The experiments done by Gan-Mor showed that the deviation of the RTK-DGPS is fewer than 4.05 cm, even on rougher surfaces [4]. Although the experiments done by Gomez-Gil did not measure deviation, they provided good data on the consistency of a tractor following an arc, circle and cornering [5]. The experiment done by Stombaugh made their measurements based on the amount that the steering had to be corrected instead of measuring the tractors path. It also showed a good consistency when following a straight line being the tractor had to make steering corrections down to 0.04 Hertz [9].

The experiment that had the least success with deviation was done by O'Connor and company. They were using the older CP-DGPS system. The experiment showed a deviation up to 5 cm [7]. This may not seem like much, but given the fact that all of the experimenting was done on a golf cart, it is inadequate being these systems are used on tractors. Another CP-DGPS experiment done by Thuilot also showed little success being that they had difficulty getting a tractor to consistently follow arcs and curves.

At times, there was a deviation up to 50 cm [10]. These experiments were performed in the earlier stages of CP-GPS, as opposed to the experiments done later in development by Stombaugh and Gan-Mor, which displayed better results.

The experiment done by Demmel had complete different methods and measureables. Instead of having it just follow a straight line, they had a tractor till a complete field [2]. The tractor was set up using a DGPS system. They made many comparisons, including how much area the tractor tilled compared to the area of the field. The experiment showed that the tractor tilled 113 percent of the total acreage [2]. To show comparison, human drivers have an average overlap of 10 percent [6]. This indicated that overlap wasn't completely eliminated in the experiment, but was impressive being that is was all done on complete automation.

Many of these experiments exhibited the same barriers. The deviation of the tractor path was increased with increased speeds as well as rougher surfaces. None of the experiments were performed on very hilly surfaces. Being the GPS system is place at the top of the tractor, the system needs to take into accordance the roll and wheel slip of the tractor. This part hasn't been completely mastered but has been improved. Many of these experiments did not include the tractor pulling implement. This would have increased the deviations because of affect that implement can have with steering and a tractors ability to follow a path.

Figure 2 displays the graphics of the maximum deviation found in the GPS systems gathered from the previous studies. It is apparent that the RTK-DGPS is much more accurate than the older CP-DGPS.

Figure 3 shows the cost of these two types of GPS systems. Though the RTK-DGPS is much more accurate, it is much more expensive. Costs of these systems can be found to be around 20,000 dollars while DGPS can cost around a fourth as much at 5,000 dollars.

5. CONCLUSION

From the information that has been gathered from the experiments, it is proven that the RTK-GPS and RTK-DGPS are the most accurate systems. They provided the highest accuracy as well as the ability to follow several different paths. The downside to them is that they require more equipment and technology, therefore they cost more. If these systems can be adapted to better take in the affect of wheel slip and vehicle roll, the use of them can increase.

There are still several other barriers that exist. Further advancements have been made to make a tractor completely automatic. John Deere has made several steps in providing a tractor that can change gears, turn, and raising and lowering various implements [6]. In an interview with Daniel Lubahn who is a John Deere mechanic at a local dealership in Wanamingo, Minnesota, it becomes apparent why they automation is not commonly used more. The main issue against automation surrounds several unique use cases. For example, an automatic tractor cannot detect a mechanic failure or if a dog runs in front of the vehicle. Once these types of scenarios can be overcome, then automated tractors may become more popular. Google is currently working with autopilot cars being able to detect these types of scenarios. Perhaps the technology advancements made by Google in automation can help improve the automation of tractors [12].

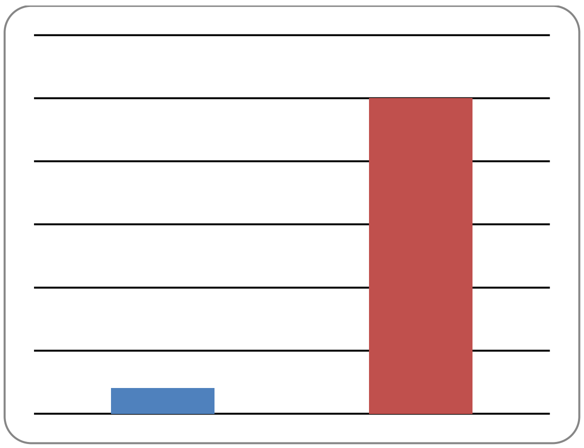


Figure 2. Maximum deviation (cm) found in GPS systems.

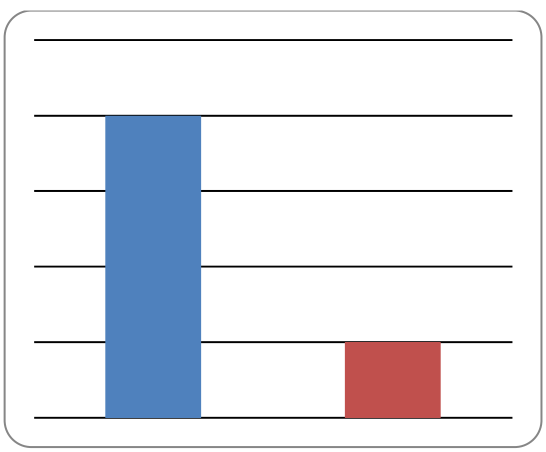


Figure 3. Approximate cost of GPS systems.

6. REFERENCES

[1] Buick, Roz. "RTK base station networks driving adoption of GPS +/- 1 inch automated steering among crop growers." Trimble Agriculture Division. Westminster, Co. August 2006.

[2] Demmel, M., Rothmund, M., Spangler, A., Auernhammer, H. "Algorithms for Data Analysis and First Results of Automatic Data Acquisition with GPS and LBS on Tractor-Implement Combinations." Technische Universitaet Muenchen, Germany. 2001.

[3] Forrest Laws Farm Press Editorial Staff. "New AutoFarm System Putting Tractor Guidance on the Curve." Delta Farm Press. 18, Feb. 2005.

[4] Gan-Mor, Samuel, Clark, Rex L., Upchurch, Bruce L. "Implement lateral position accuracy under RTK-GPS tractor guidance." Computers and Electronics in Agriculture. Volume 59, 31-38. April 2007.

[5] Gomez-Gil, J., Ryu, J.C., Alonso-Garcia, S., Agrawal, S.K. "Development and Validation of Globally Asymptotically Stable Control Laws for Automatic Tractor Guidance." Applied Engineering in Agriculture. Volume 27, 1099-1108. 2011.

[6] Murray, Charles J. "Deere Takes Next Step Toward Driverless Tractor." Design News. Feb. 2008. Referenced February 2012. www.designnews.com/document.asp?doc_id=219094.

[7] O'Connor, Michael, Elkaim, Gabriel, Parkinson, Bradford. "Kinematic GPS for Closed-Loop Control of Farm and Construction Vehicles." ION GPS-95. Palm Springs, CA. Sept. 1995.

[8] Reid, John, Zhang, Qin. "Automated Guidance of Row-Crop Cultivation." Department of Crop Sciences. University of Illinois. 2000.

[9] Stombaugh, T.S., Benson, E.R., Hummel, J.W. "Automatic Guidance of Agricultural Vehicles at High Field Speeds." ASAE Papers. July 1998.

[10] Thuilot, B., Cariou, C., Martinet, P., Berducat, M. "Automatic Guidance of a Farm Tractor Relying on a Single CP-DGPS." Autonomous Robots. Volume 13, 53-71. 2002.

[11] Toller, E. W., R. L. Schafer, and T. K. Amrita. "Sensors and controllers for primary drivers and soil engaging implements." In Advances in Soil Dynamics. Volume 2, 179-224. St. Joseph, Mich.: ASAE. 2002.

[12] Vandermey, Anne. "Wheels of Tomorrow." Fortune. Volume 165, Number 5, 25-32. April 2012.

Usability of Mobile Education Games by Screen Size

Jesse Buss
Department of Computer Science, Winona State University
175 West Mark Street
Winona, MN 55987
507-457-5000
jbuss08@winona.edu

ABSTRACT

We created an Android application to help students learn the basics of math in a fun way. Instead of simply seeing a problem and writing the answer, the user will be making decisions in real time about equations, numbers and math problems. Numbers will be displayed inside of balloons that are floating toward the top of the screen on an Android device. The user will decide if the balloon fits in the category specified on the left side of the screen, right side of the screen, both, or neither and drag the balloon to its corresponding position. We compared the usability of this application on an Android phone with a 4.3” screen and an Android tablet with a 7” screen to show which supported a more intuitive user interface and a better overall user experience.

Keywords

Android, mobile, tablet, phone, education, game, usability, software development.

1. INTRODUCTION

Mobile technology has been rapidly growing the past few years. In December of 2011, Google reported that over 700,000 Android devices were being activated every day [3]. Figure 1 shows steady growth since the inception of the Android operating system, and shows no signs of slowing down. With so many people having access to tablets and smart phones, developers have a good reason to release applications and games for these devices. These tablets and phones provide another medium for entertainment as well as education and learning. The application we wrote is a game that is easy to use and learn.

The target users of our application are elementary students, however it will be challenging enough for high school students as well. We assume that many of these elementary students don’t actually have Android devices of their own and will be using their parents’. Thus, it is important to also market this game to parents, as they may test it out before they allow their children to use it.

Proceedings of the 12th Winona Computer Science Undergraduate Research Seminar, April 25, 2012, Winona, MN, US.

Using the Android operating system for education, as well as entertainment, is an important issue that can improve the math skills of many students across the world. When students are able to sit down in their free time and learn math, English, history and other various subjects, it allows them to excel in school and paves the way for them to be successful adults. We compared the usability of this educational game based on screen size to determine which device created a better user interface and overall experience. Those devices that provide better user experiences should be used in education, as they will likely result in more fun, more efficient learning.

The Android tablet we will be using is the Amazon Kindle Fire, which has a 7” display. We will compare this tablet with the HTC ThunderBolt, which has a 4.3” display. Both devices have touch screens and full-color displays.

We hypothesize that more users will report a better experience playing the game on the Kindle Fire than on the ThunderBolt. We think that the Kindle Fire ultimately creates a better environment due to its large screen size, which will allow the user to see the balloons more clearly and play the game more easily.

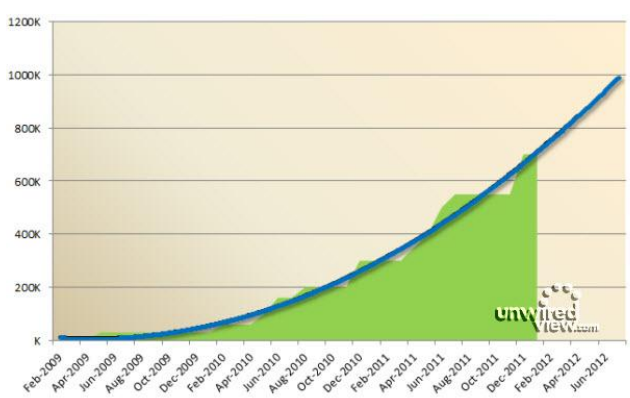


Figure 2. Number of Android Device Activated Daily [2]

2. METHOD

There are different ways to design multi-platform or multi-device applications. One way is to create two separate applications for the different platforms or devices. This solution isn’t popular because it is difficult to maintain and extend. Instead of having one application to update, you now have two, and they may be in different languages. Our application is written for different Android devices, so we do not have to worry about developing the game in multiple languages. We chose not to write two separate applications, one for the Amazon Kindle Fire, and one for the HTC ThunderBolt, because it may cause inconsistencies within the user experience. Instead, we wrote one application that can be run on both the Android tablet and the Android phone [4].

2.1 Game Design and Development

The software-development model that we chose to utilize is the exploratory programming development model (see Figure 2). This model is closely related to the prototyping model and iterative development model of software development. The major difference between prototyping and exploratory programming is that you need to know a complete specification of the project before you begin developing with prototyping. Since we were developing new software that we are unfamiliar with, we wanted to allow ourselves the ability to change specifications during development in order to make our game run more efficiently [3].

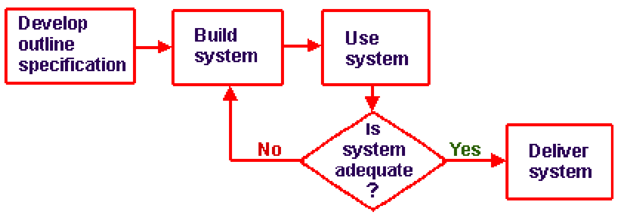


Figure 3. Exploratory Programming Software Lifecycle [1]

As you can see in Figure 2, the first step in this paradigm is to develop an outline specification. After the specification has been defined, you begin the development loop. This loop is an iterative process, in which deliverable software is produced at the end of each loop, or at the "Is the system adequate?" decision point [1].

The application we developed is called "Math is Fun." We created one application that is playable on different types of Android devices regardless of screen size. We used Eclipse as our development environment and Eclipse plug-ins for Android that allow you to emulate a physical android device in software. We developed our application using the Java programming language, as this is the language that Android applications run in.

Throughout the game, balloons will float from the bottom to the top of the screen. Each balloon will have a number or equation on it. To play the game, the user will reposition the balloon to its correct location. For the experiment, we developed the game so the left side of the screen will have a label, "4" which signifies "Multiples of 4." The right side of the screen will be labeled "3" which signifies "Multiples of 3." These labels will be explained to the user before they begin the game. The user will have to determine based on the value of each balloon whether it belongs on the left side (in which the user will drag it to the left) on the right side (in which the user will drag it to the right), neither (in which the user will drag the balloon to the center), or both (in which the user can drag the balloon to the right or left side). See Figures 3 and 4 for a visual representation of this description. As an alternative to dragging balloons, users can simply click the side of the screen they want the balloon to go to. The click must be within the same plane of the y value of the balloon. Figure 3 gives a screenshot of the game in action. The balloons labeled "70" and "58" are in the center since they are not a multiple of 3 or 4. The balloon labeled "66" is in the right column since it is a multiple of 3.

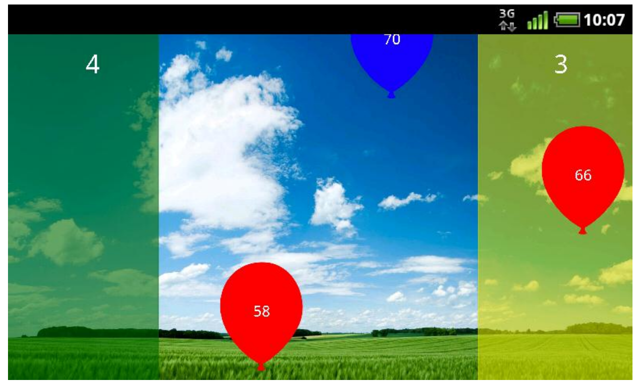


Figure 4. User Interface on ThunderBolt

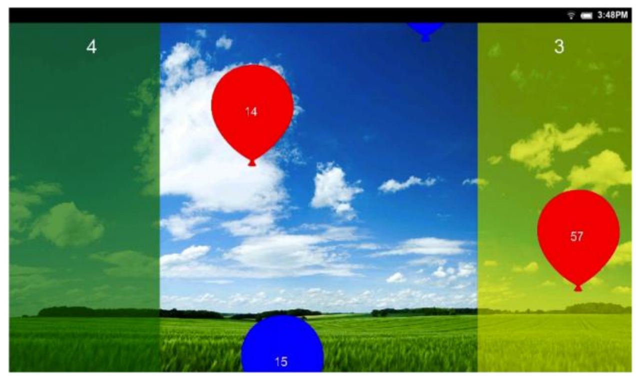


Figure 5. User Interface on Kindle Fire

Figures 3 and 4 show the interfaces of the application on the two test devices. The figures have been adjusted, are not to scale and do not represent the actual size of the devices. The user interfaces on each device are built using the same code. The code does not have two different methods to build the user interface based on screen size, but rather uses the screen size to make calculations. The raw balloon files are scaled down to fit based on the size of the display. The left and right column markers and the background image are also scaled down based on the size of the screen, and the font size is the same on both devices.

2.2 Survey

We had two different Android devices that we used to complete the study; an Amazon Kindle Fire tablet and an HTC ThunderBolt phone. The Amazon Kindle Fire has a 7" full-color multi-touch display, and the HTC ThunderBolt has a 4.3" full-color multi-touch display. In order to remove inconsistencies between the devices, we kept the devices as similar as possible. Neither device had a screen protector, and both devices had the brightness setting set to 100%. The ThunderBolt was on airplane mode (not using any data) and the Kindle Fire was not connected to Wi-Fi in order to minimize interruptions. Some participants of the experiment used the Amazon Kindle Fire first, and some participants used the HTC ThunderBolt first. The device to be used first will be chosen at random. After each participant used each device for five minutes, they filled out a survey that measured which device they preferred and why, and which device they would recommend to elementary students and why.

We tested our hypothesis by doing a usability study of fifteen people of varying of ages. Of the fifteen participants, three were in high school or elementary school, five were in college, one was between 25 and 39 years old, and six were 40+ years old. Seven participants used the Kindle Fire first and eight used the ThunderBolt first. The participants also had a wide range of knowledge of Android devices, some use Android devices every day and others have never used one. We made sure to include a mix of both male and female participants; we tested 9 males and 6 females. We also tested parents of children that may use this game to learn.

3. RESULTS

Table 1 shows selected questions and answers from the survey. There were a total of fifteen participants in the study. Three of the users reported that they were in elementary or high school, five reported they were in college, one was between the ages of 25-40, and six were 40 years old or older. Everybody over the age of 25 that participated in the survey was a parent. Of the fifteen people surveyed, seven used the Amazon Kindle Fire first, and eight used the HTC ThunderBolt first. Thirteen of the fifteen stated that the game responded the way they expected on the Kindle Fire. One participant noted that he/she chose “no” because he/she expected the game to be laggy and not run as smoothly. Twelve participants reported that the ThunderBolt responded the way they expected.

The next three questions of the survey -- “Which device had better colors?” “Which device ran the game smoother?” and “Which device did the balloons respond to quicker?” -- were included in the survey in order to determine if the physical device’s hardware (aside from screen size) could have affected the results. We wanted to make sure that users were getting a similar experience between the two devices and eliminate as many variables as possible in order to test screen size most effectively. Eight participants reported that the Kindle Fire had better colors, while four reported that the ThunderBolt had better colors, and three reported a tie. Seven participants said the Kindle Fire ran the game smoother, four said the ThunderBolt ran the game smoother, and four reported a tie. Four participants said that the Kindle Fire responded quicker to balloon touch, seven said the ThunderBolt responded faster, and four said it was a tie.

Nine of the fifteen participants reported that the Kindle Fire provided a better overall experience, while three of the fifteen reported that the ThunderBolt provided a better overall experience, and the remaining three reported a tie. Eleven of the fifteen participants said they would recommend the Kindle Fire to elementary students. Three of the fifteen said they would recommend the ThunderBolt, and one person reported a tie. Nine participants reported that they were most satisfied while playing the game on the Kindle Fire, four reported they were most satisfied when playing on the ThunderBolt, and two reported a tie.

4. CONCLUSION

More survey participants preferred the game on the Kindle Fire. Some reasons that the participants gave were “larger screen size,” “easier to see the balloons,” “easier to drag the balloons,” amongst others. The reasons given were not indicative of the ThunderBolt not providing a far inferior experience. The Kindle Fire did better than the ThunderBolt when asked about the color quality and the smoothness of the game, but the ThunderBolt was the majority’s choice for the responsiveness of the balloons.

Table 1. Usability Survey Results

Did the game respond the way you expected on the Kindle Fire?	
Yes	No
13	2

Did the game respond the way you expected on the ThunderBolt?	
Yes	No
12	3

Which device had better colors?		
Kindle Fire	ThunderBolt	Tie
8	4	3

Which device ran the game smoother?		
Kindle Fire	ThunderBolt	Tie
7	4	4

Which device did the balloons respond to quicker?		
Kindle Fire	ThunderBolt	Tie
4	7	4

Which device provided a better overall experience?		
Kindle Fire	ThunderBolt	Tie
9	3	3

Which device would you recommend to elementary school students?		
Kindle Fire	ThunderBolt	Tie
11	3	1

Which device were you more satisfied with while playing the game?		
Kindle Fire	ThunderBolt	Tie
9	4	2

Because the user experience on each device was comparable, we assume that the main difference between the two devices was screen size. The larger screen of the Kindle Fire allowed the participants to see the numbers in the balloons easier. The Kindle Fire’s screen also caused fewer errors in moving the balloons, since the participants had a larger area to drag the balloons, as reported by one participant. For the “Math Is Fun” application, more users preferred the experience on the Kindle Fire than on the ThunderBolt.

5. REFERENCES

[1] Drewry, Tony. "UWE-CSM - IT System Development Lifecycles." Software and System Development Life-cycles. University of the West England. Web. 03 Feb. 2012. <http://www.cems.uwe.ac.uk/~tdrewry/Lifecycl.htm>.

[2] Duke, Kevin. "Fabulous !! 3.7 Android Devices Activated on Christmas (24/25) December." Vandtab : Cell Phone News. 28 Dec. 2011. Web. 2 Feb. 2011.

[3] Nickinson, Phil. "More than 700,000 Android Devices Activated Daily." Android Central. 20 Dec. 2011. Web. 02 Feb. 2012. <http://www.androidcentral.com/more -700000-android-devices-activated-daily>.

[4] Paternó, Fabio. "Models for Universal Usability." ACM (2003): 9-16. Print.

Band-Saw Blade Estimator

Tyler Kulig
Winona State University Computer Science Department
Fastenal Company Information Technology Department
106 E Sanborn
Winona MN, 55987
(715) 533-3951
Tbkulig07@winona.edu

ABSTRACT

Designing a good user interface requires usability testing on some scale. Fastenal is a company that sells industrial and construction supplies and offers services including inventory management, manufacturing and tool repair. Fastenal evaluates software on three standards: effectiveness, efficiency and scalability. One type of item that is sold is called band saw blades. Fastenal's band saw blade ordering system hasn't been drastically changed in over 15 years. The two methods brought into question are the old Macro Excel style, which is the current format, and a newly created web based style. Fastenal employees acting as test subjects processed orders on both ordering systems (Web and Macro). The newer more intuitive style is compared against the old defined type. The study conducted showed that the new web based style was overall favored more for its ease and its non-confusing page flow. Most users whether they have used the old application before or not rated it higher in almost every category. Although the subjects didn't always grade the Macro style as poor, those same subjects scored the web-based style higher.

General Terms

Human Factors

Keywords

User interface, survey, heat maps

1. INTRODUCTION

Our study focuses on an area in which deals with a regular human interface. The user interface is called Fastenal's Band-saw Blade Estimator (BSBE). The estimator is used company-wide across Fastenal to create custom orders [1]. Each store around the globe has access to the basic estimator. It's a simple file, run in Excel using Macros, and some Visual Basic [2]. People use many different ways to interface with an actual system to buy, sell and return product. Due to the fact that almost every business is out to be fast, concise, profitable and useful. Without over doing it, it's better to makes things easier than to make them complicated. As a business you would not want to over simplify a user interface. The key fields and entered information should flow without long delays or without confusing dialect. Currently there are a lot of different GUI's (Graphical User Interfaces) available. Some use a lot of text fields and some use a lot of imagery to convey their purpose. The social responses to media and computer-based interfaces have become more of a focal point in HCI studies following previous works and building on those successes [8]. Tsung-Hsiang, and Li have a perfect example of research on this topic; which was the inability or ability to use images to help with searches compared to just text searches [3]. If it's simple enough of a user interface it can prove effective as means of searching for something just like text. Even things on a more simple level such as what type of display is being used can be significant. Not everyone uses the same display type, and a single screen. Some use multiple screens. Working this into the program would allow it to be more approachable for users. The interface design of our BSBE considered these factors and scaled the size of the user program to work effectively on a single display screen giving people with low resolutions a completely functioning GUI. Yanagida, Nonaka, and Kurihara reviewed this type of method, they worked with what they called flexible widget layout (FWL), which could similarly be described as a GUI [4]. This allows for the change and adaptation of user's screens sizes to not be a factor.

We studied and reviewed an improvement of an estimator from excel using Macro's to a new web-based style of program to see if this style can be considered an improvement. The macro uses an outdated but still affective system. Our new program will allow for multiple stores that access it via Fastenal's own inter-network of programs/website to quickly produce order forms and or returns of product than before. It is concerned with a smaller section of orders in a specialized portion called welded band-saw blades. It's a circular blade cut to specifications for each particular order. The old process is an eight-step process that included extended waiting times including additional interactions with the customer between some steps of the ordering process. The new web-style was condensed into 4 steps [2].

2. HYPOTHESIS

A web-based-style interface of the Fastenal's BSBE alone is enough to replace the old Macro BSBE which is currently in production.

3. METHOD

Our study compared the differences between the older well-worked non-sequential GUI and a newer sequential GUI. The study was conducted to demonstrate the differences in usability for each section of the program. As well as cover the time it takes to complete orders. This includes the effort level, which can be

Proceedings of the 12th Winona Computer Science Undergraduate Research Seminar, April 24, 2012, Winona, MN, US.

measured by average time to complete and total amount of needed effort to complete a task or order. Each program has a supporting database of parts with key information tied to them. Figure 1 shows populated list of parts and it's from there is where you start producing the order.

From that point the sections are used to fill out details needed to complete the other steps of the eight-step process. It could be navigated simply by using a combination of mouse and keyboard commands [1]. The page offered no images or instructions for use. The field sections only offered headers to give you a brief layout of what each section was responsible for. All the text was displayed in a legible format. Figure 2 shows the bi-product of completing the form in Figure 1. You would then have (for most cases) a completed order ready to be faxed for confirmation. Figure two has no other purpose for the order than showing a more cleaned up format of that order.

3.1.1 Participants

Fourteen Fastenal employees (8 males, and 6 females) of which some were software developers and the others non-developers where the participants. The age range included users in their 20's, 30's, 40's and 50's with the average falling in the 30's. Some users had some experience with the old BSBE and some had no experience with the BSBE in any form. All participants had knowledge of Excel and all formats of software used. All participants volunteered of their own free will. The participants were briefed on the intent of what was to be done in each test run. Also they were informed of the questionnaire that follows each run of the software. The order of the two programs was chosen randomly.

3.1.2 Procedure for tests

(1) Participants would verbally describe their feelings of the layout of the page. This included describing content and concerns with their usability. This also allowed for users who never used it before to get somewhat comfortable with what they are shown to reduce shock of the unfamiliar. As well they had access to the Table 1 to give them an idea of what each program key points were.

(2) Participants were tasked to follow two premade orders that encompassed finding the part numbers from the list that were not displayed when the page loaded. The instructions described what had to be entered, but not where to locate the fields. This was done to show if there was a simple flow or if things seemed hidden or confusing,

(3) After the two test runs of orders the participants would then fill out a questionnaire based on the framework of the user interface. Questions included things pertaining to aesthetics, identification, content, and usability. The answers to the questions were described in words for example, Easy/Difficult, Agree/Disagree, Likely/Unlikely, and Fast/Slow.

(4) After completion of the first program the process would be repeated for the other program.

(5) Finally comments/concerns were then listened to after completion of both questionnaires.

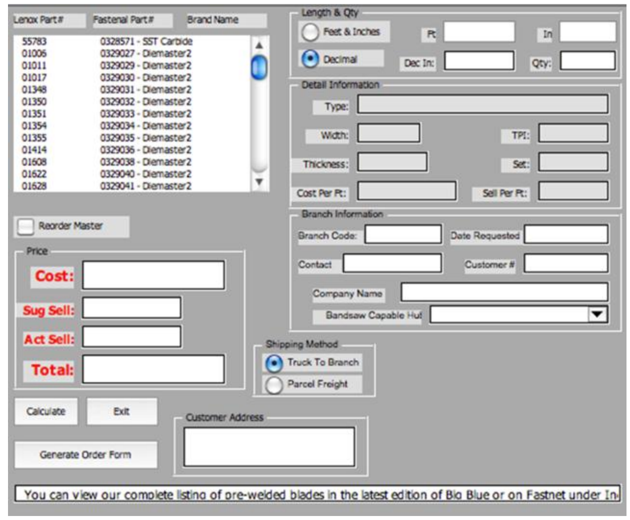


Figure 1. Fastenal's page layout of the original band saw blade estimator

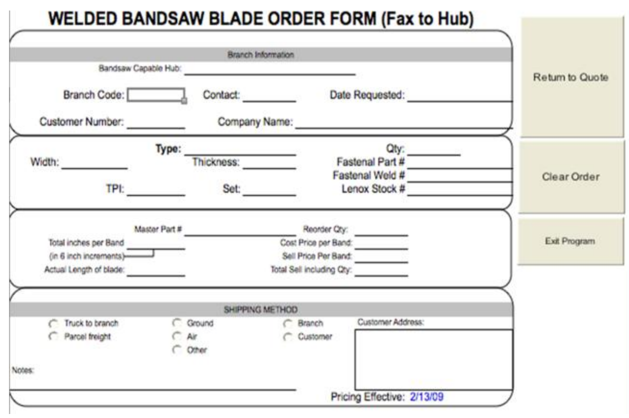


Figure 2. The entered data from Figure 1 fills this page that is required to be faxed out.

Figure 3. Locations of fixation by participants

Table 1. Key points of each BSBE

Key points/features of UI	Macro BSBE	Web BSBE
Content of each GUI	Store information, pricing information, part detail, part quantity, shipping method	Part detail, part quantity
Navigation type	Non-sequential	Sequential
Reputation	Well established	New
Imagery	No	Yes

4. RESULTS

Figures 3 - 5 show images of the program GUI's with a heat map showing where points of interest were located for users on first inspection. A heat map is any data visualization, which uses color to represent data values in a two-dimensional image. The maps were tabulated in an online survey via Qulatrics. This effectively allows for users to take identical questionnaires.

On a scale of 5 being the fastest and 1 being the slowest the average for the Macro-BSBE was at a 2.79, whereas the Web-style was at 4.86. On the same scale 1-5 the purpose of the Web-style was .5 higher than the Macro. Table 2 below shows the average of each sections main goals on a scale of 1-6, six being the highest.

4.1.1 Analysis

Completing a questionnaire requires more than just reading questions. You have to think, process information, and make decisions. According to Sharon Oviatt the field of the research for psychology pertaining to UI's has been focused on quantifying the cognitive load and finding ways to reduce problematic spots. This will in turn minimize load and increase effectiveness and efficiency [7]. The same approach was used as a guideline for this study.

The Web-style interface lines up the order of operations into a noticeable flow that from the gathered data seems to be more efficient and faster. This is a noticeable improvement to generate one order to the next. As described by C. Halverson, "The end game always must be sales dollars, or a large gain in efficiencies…sales should go up because we have made it easier for our customers to buy and our sales personal to sell"[2]. The difficulty of a program can easily turn people away from using it and repeating business with Fastenal or getting any business at all. Thus when the data was compared for difficulty, users rated ease of use of the old one on average at 3.19 out of 6. Inversely, the Web-style received an average of 4.71 signifying that it was in the range "fast".

We also analyzed the first overall position of the users eyes when they saw the program's screen, using heat maps. This data allows you to quantify where people's attention is drawn, which in turn shows where your program should logically start from to avoid confusion. Figure 3 and 4 show the same images as Figures 1 and 2 but these have heat maps showing points of original interest on the page. Figure 3 gives a good indication of where to start, unfortunately the flow from one point to the next causes confusion. Participants rated the Macro-style to be "Probably not obvious" on what to do when you saw the GUI. The heat map indicates that users split into three main spots for where they focused first, a large menu of parts (which is the start), a secondary step deciding on Length and Quantity, and finally a display section that had no real required input.

Figure 3 showed some consistency on what was presented. Participants either first noticed the key buttons, or read the header of the page before anything else. This significantly shows that its main points are focused and enhanced to help reduce cognitive load. The Web-style program showed consistency relating to the normal human behavior upon reading. The flow started from the top down and was left to right. Most participants looked first the main field required to be entered. From the numbers generated by the questionnaires and the heat maps the cost of creating and

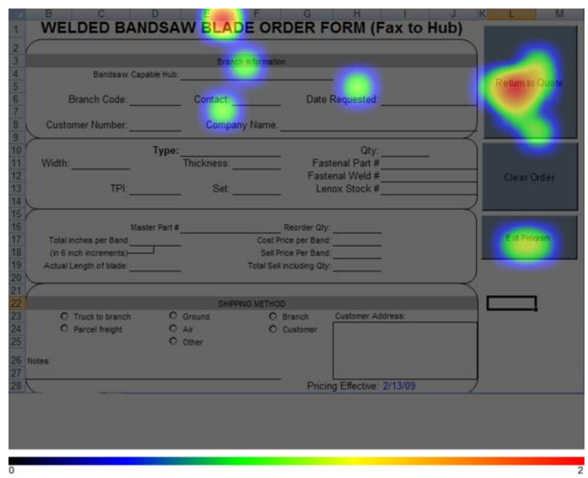


Figure 4. Shows the invoice splash pages key points.

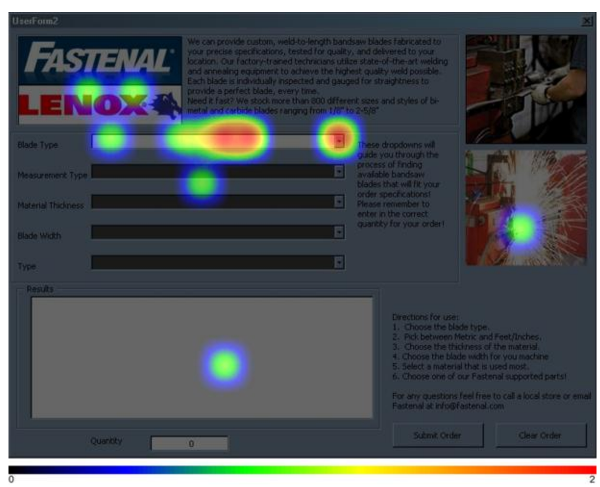


Figure 5. Participants fixation on the point they noticed first.

maintaining a separate database on the web of parts is smaller than the gains of using the new software. You will be able to know for certain that all users have the same database of information with the Web-style compared to pushing out new versions of the parts database in excel via email or other mediums. This in turn gives it much more scalability by editing the database online.

4.1.2 Discussion from Participants

Many users felt the Macro-style was overwhelming and had issues of placement of items. The biggest issue being the location of quantity field which was a required field to complete any order. We observed that it took many participants well over 30 seconds to find the location of quantity, purely because its position was masked as the second field for decimal inches. The feet and inches section had two text fields that allowed for easy non-conversion of length, but decimal was calculated from feet and inches meaning there was only one section. The second sections location then became quantity. Another complaint about the old GUI was the inability to easily search the large list of parts. To find the correct part, participants had to scroll. The shortcut of typing the first few characters was unavailable. The new Web-style isn't perfect either. Users felt confused when they had to choose several key factors of a blade. Although after completing the entire experiment most participants describe that they understood that if they were looking to find a part most of the key factors of the blade are things they knew which would remove confusion. They also felt it was overwhelming, but the availability of the instructions on screen allowed for easier order process.

5. CONCLUSION

Currently there is a lot of different GUI's available and many different types. Some use a lot of text fields and some use a lot of imagery to convey its purpose. People use many different ways to interface with an actual system to buy, sell and return product. From the two methods we studied (Macro and Web-style) in this paper, the results showed that the new was better when compared to its predecessor in every comparison field. Averages shown in Table 2 were up across the board. We observed that there is a great deal of benefit to move into a complete new Web-style program to replace the old. It offers a vast amount of scalability; it's extremely efficient since it's so simple. Therefore since its simple that translates into being effective with users. Meanwhile, to re-implement the old Macro style would save some time being that there is one in place but the costs of updating it is less than worth it.

5.1.1 Future Work

Though through these tests we've found several issues to be addressed in the new Web-style GUI. Tthe groundwork has been laid for a more intensive study into the BSBE. Future work could/should include a fast order pad to allow users who know all specifications to simply enter in the part they are looking for and order it. The addition of being connected to the web will allow for tests to conduct whether or not store and user information is transferred to completely reassure the new 4 step process takes out the delays between the old eight step processes. This opens the door for more testing into sequential UI's against non-sequential UI's in certain conditions to find when and where each are most effective.

Table 2. Base averages for key sections.

Averages based on a scale of 6 being best	Old	New
Efficiency	2.99	3.99
Speed	3.03	3.98
Usability	2.92	4.44
Difficulty	3.19	4.71

6. ACKNOWLEDGMENTS

Our thanks to those at Fastenal, and all the participating employees including, John Marter.

7. REFERENCES

[1] Lee, I. (2012, January 18th). Email Interview

[2] Halvorson, C. (2012, January 18th). Personal Interview

[3] T.-H. Chang and Y. Li. Using Graphical Representation of User Interfaces as Visual References. In UIST '11. ACM.

[4] Yanagida, T. Nonaka, H. Kurihara, M. Personalizing Graphical User Interface on Flexible Widget Layout. EICS'09, July 15–17, 2009, Pittsburgh, Pennsylvania, USA.

[5] Apple Inc. Apple human interface guidelines, 19, 2012. Available at http://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/AppleHIGuidelines/OSXHIGuidelines.pdf

[6] Gajos, K. Czerwinski, M. Tan, D. Weld, D Exploring the Design Space for Adaptive Graphical User Interfaces. May 23 AVI '06. ACM

[7] Oviatt, S. Human-Centered Design Meets Cognitive Load Theory: Designing Interfaces that Help People Think. MM '06. ACM

[8] Karsvall, A. Personality Preferences in Graphical Interface Design. NordiCHI October 19-23, 2002. ACM.

[9] Hartmann, J. Sutcliffe, A. De Angeli, A. Investigating Attractiveness in Web User Interfaces. CHI Proceedings. April 28-May 3, 2007. ACM

Vulnerability of Wireless Network Security due to Parallelized Brute Force Attacks

Matthew Wolfe
Winona State University – Computer Science Department
34487 Old Homer Road
Winona, MN 55987
(507)-313-9870
mjwolfe07@winona.edu

ABSTRACT

Wireless networks are becoming one of the most prominent methods for businesses and consumers to share digital information. While wireless networks are convenient, they also create a potential vulnerability for companies who choose to deploy them. The current security encryption schemes for wireless networks are susceptible to dictionary attacks as well as traditional attacks that compromise the 802.11 standard. This paper discusses the possibly of a malicious user's ability to gain unauthorized access to a wireless network using highly parallelized brute force dictionary attacks designed to run on a Graphics Processing Unit (GPU). In order to determine whether a GPU optimized attack would increase the probability of gaining unauthorized access, the attacks were run under three different configurations: serial code run on a single processor, multithreaded code on a quad-core processor, and GPU optimized code. In order to provide quantitative data for analysis, two different benchmarks were used. These benchmarks produced results that showed the GPU optimized code, while producing more overhead, would ultimately reveal a wireless network key faster than single and multiple core optimized code.

Keywords

WPA/WPA2, CUDA, brute force attack, wireless network security

1. INTRODUCTION

As the number of 802.11 wireless devices continues to grow, there is an increasing need to develop robust wireless security protocols. Initial protocols such as Wired Equivalent Privacy (WEP) and WiFi Protected Access (WPA/WPA2) do not provide the level of protection that is necessary to securely protect home and enterprise networks. These networks are susceptible to traditional attacks against the 802.11 standard [4]. In addition, the newest version of the WiFi Protected Access protocol, WPA2, can be compromised by pre-computing part of the authentication phase used by the protocol, and piping these results into a piece of software that checks each computed Pairwise Master key (PMK) against the captured wireless data [3].

Although these attacks have been proven to be effective against current wireless security protocols, brute force attacks could become more powerful if they were able to execute in parallel. High performance hardware has become readily available at the consumer level due to the generic programming capabilities of graphic processing units (GPUs). Languages such as CUDA, developed by NVIDIA, and OpenCL are able to take advantage of the multiple core vector architecture that the graphics cards provide. By developing algorithms suited for parallel execution, wireless encryption schemes could be broken in a fraction of the time it would take a serial processor to do [1]. Figure 1 depicts different hardware platforms and their peak output of pre-computed Pairwise Master keys per second.

In order to determine whether current wireless security schemes are viable options for future wireless security, a series of penetration tests have been conducted to validate security strength. An open source security distribution of Linux called Backtrack (http://www.backtrack-linux.org/about/) has been released and contains software designed specifically for wireless security testing. The software that was used in our experiments is called Pryit [2]. It is software that pre-computes the PMKs and stores them so that they can be compared to the wireless network data that has been captured [3].

The experiment phase of our project consisted of a brute force dictionary attack as well as a built-in benchmarking utility. Both of these tests were run on three different hardware configurations. The first hardware platform that we tested was a quad core processor with only one core enabled. Next, the three remaining cores on the quad-core processor were enabled thus changing the configuration into a multiprocessor. Finally, a GPU optimized module was loaded for Pyrit, and the attack was run on an NVIDIA GeForce 9400 GT video card in addition to four of the cores on the processor.

The use of the GPU accelerated program will allow a malicious user to gain unauthorized access to a wireless network faster than the quad-core program and the quad-core program will allow access faster than the single core system. By comparing the wall clock time of each attack, as well as the average PMKs per second using the benchmarking utility, the determination will be made as to whether the attacker would be able to come up to an arbitrary wireless network and gain access to it. Depending on the result of the attacks, it may be made clear that a 'zero information attack' is feasible or that the current encryption schemes are secure enough to prevent these types of attacks.

Proceedings of the 12th Winona Computer Science Undergraduate Research Seminar, April 25, 2012, Winona, MN, US.

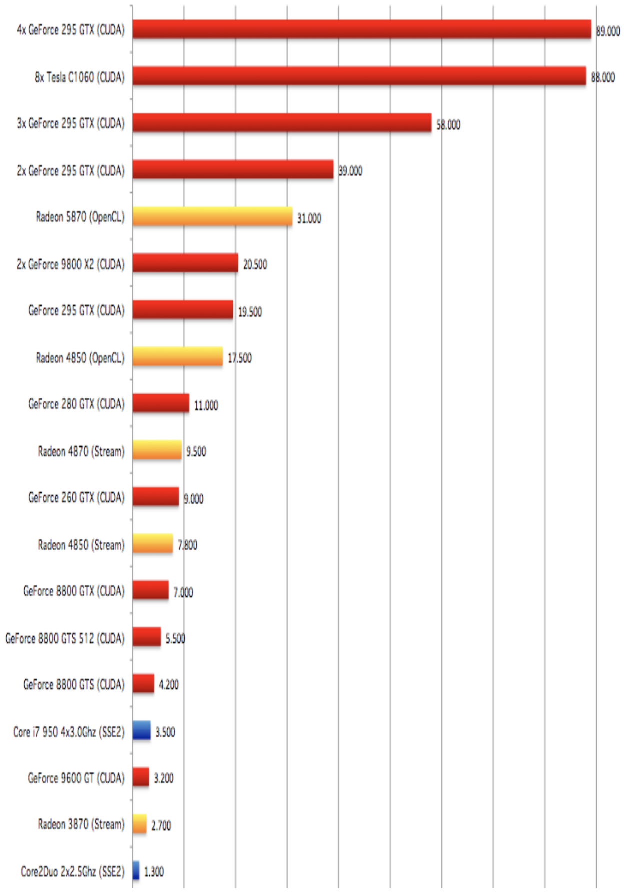


Figure 1. Pyrit performance on different hardware configurations [3]

2. METHODOLOGY

In order to complete the experimentation to prove the hypotheses, the necessary tools and equipment for developing a program for specific use on the GPU had to be acquired. This included basic knowledge of the C programming language as well as the CUDA extensions. This provided the testers adequate information to analyze the Pyrit code and determine how each different hardware configuration would change the function of the program. In addition, knowledge of the architecture of a graphics card was also necessary to attain high performance. The special hardware that was acquired is an NVIDIA GeForce 9400 GT pci-express graphics card. This card was added into a custom PC configuration. Full technical specifications of both the graphics card and CPU are shown in Figures 2 and 3.

In addition to the programming languages and paradigm used, several pieces of software were used during testing. The software used for the project came in a self-contained distribution of Linux called Backtrack 5. This distribution contains the proper wireless card drivers necessary for replaying and injecting packets into an unconnected network. It also contains Pyrit and its CUDA module, in addition to the CUDA toolkit and corresponding development environment. Although Backtrack 5 can operate from a 'live CD,' a physical installation of the operating system was used in order to write configuration files to the disk.

Cuda Cores	16
Graphic Clock (MHz)	550 MHz
Processor Clock (MHz)	1400 MHz
Memory Clock	400 MHz
Standard Memory Configuration	512 MB
Memory Interface Width	128-bit
Memory Bandwidth (GB/s)	12.8

Figure 2. NVIDIA GeForce 9400 GT Technical Specifications [5]

Brand	AMD
Series	Phenom II X4
Socket Type	AM3
Multicore	Quad-core
Core Size	45 nm
Frequency	3.2 GHz
L2 Cache	4 x 512 KB

Figure 3. AMD Phenom II X4 840 Technical Specifications [6]

Before the actual experimentation could be started, the test system had to be configured. This included physically installing the NVIDIA graphics card into the system as well as installing Backtrack 5 to a physical disk. Backtrack provides a standard wizard to guide a user through installing the operating system.

In order to provide enough data for analysis, the attacks were repeated ten times for each platform. After running the password recovery attack, the benchmarking utility was run ten times on each platform as well to provide enough result data. This test was run using the command *pyrit benchmark*. This benchmark creates a set of dummy data and records the number of Pairwise Master Keys the processing unit can compute per second.

When the test system has been properly configured, the first step of the password recovery experiment is to obtain the sample wireless data that is provided by Pyrit. This data contains the WPA-PSK encrypted four-way handshake that is initialized when a client attempts to connect to a wireless access point as seen in Figure 4 [2]. For each different hardware platform, the same captured data will be used because it will aid in controlling the variables between each test. This data also contains a dictionary of plain-text passwords that Pyrit uses to compute the PMK. The included dictionary contains 4091 passwords.

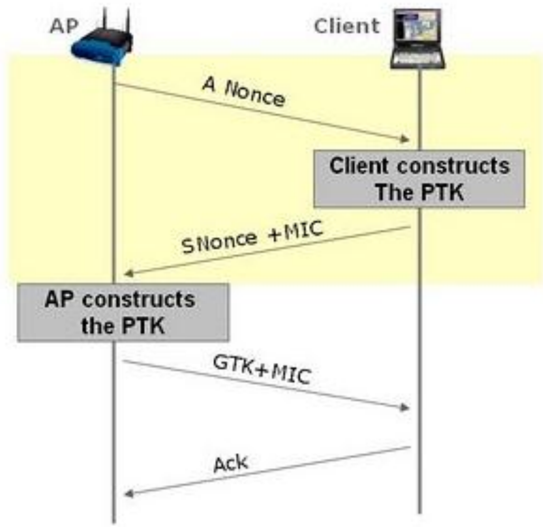


Figure 4. WPA-PSK Four-way Handshake [9]

Once the four-way handshake has been captured, the first of the attacks can be run against the data. Pyrit needs to run on the single core processor in order to test the serial version of the software. Disabling three of the four cores that the CPU operates does this. Since the user knows the password in this scenario, we can ensure that it is included in the dictionary of passcodes that Pyrit is configured to use. Once the pass begins, Pyrit will iterate through the entire list of passcodes in an attempt to match a code with the PMK that was used during the four-way handshake. At this time, a stopwatch needs to be started in order to use the wall clock time as a comparison for each different acceleration attack. After the program has found the passcode, stop the timer and record the time.

The next step in the experiment is to repeat the attacks but change the hardware platform on which Pyrit runs. The next attack is accelerated on the quad-core processor with all four cores enabled. In order to enable the cores of the CPU, the user must restart the computer and enter the BIOS menu. Depending on the type of motherboard, this can be found in different locations. A sample menu is shown in Figure 5.

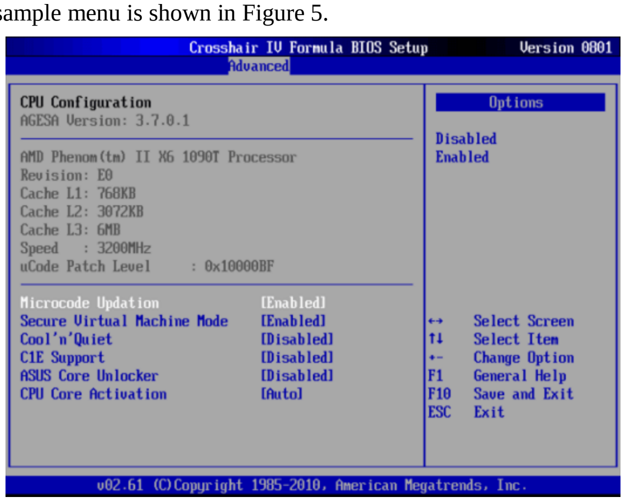


Figure 5. CPU Core Activation shown in the BIOS Menu [7].

Once that is complete, the attack runs in the exact same manner as the first attack. Finally, the GPU accelerated attack was run. For this test, the all four cores of the CPU are enabled and the special GPU module of Pyrit has to be loaded [8]. This allows for one core of the CPU to run the special packages required to send data and instructions to the GPU for generic programming purposes.

3. RESULTS

The raw data results are shown in the following tables. The first test was a benchmarking utility that comes standard with Pyrit (Figure 6). The second test used captured data from a wireless network that is publically available to Pyrit users (Figure 7). This data is then attacked using a dictionary of 4091 keys. In order to quantify the second test, a stopwatch was started at the beginning of the run and stopped once the program found the Pairwise Master key.

As Figure 6 shows, the GPU accelerated benchmark resulted in the most PMK processed per second. On average it computed approximately 6% more PMKs per second than the quad-core program and over 75% more PMKs than the single core program. Figure 6 also shows that the single core program was about one quarter the speed of the quad-core program. This is an intuitive result since the single core program had one-fourth the processing capacity as the quad-core program. The benchmarking utility test tends to support both of the hypotheses claimed earlier in the paper.

The password recovery test results are shown in Figure 7. This test produced results that do not support the hypotheses claimed. In this test, the quad-core program performed faster than the GPU accelerated program. These results will be discussed at length in the Discussion section of the paper. All three platforms performed consistently throughout the experiment. The data shows no abnormal runs or any obvious signs of an improper test.

Run #	Single Core	Quad Core	GPU Acceleration
1	705.8	2871.5	3041.1
2	706.4	2878.9	3036.3
3	706.6	2832.4	3015.6
4	706.5	2894.5	3040.2
5	706.1	2880.1	3015.6
6	706.6	2797.0	3038.9
7	706.5	2881.3	3040.3
8	706.5	2880.5	2998.6
9	706.2	2877.2	3031.8
10	706.3	2876.9	3013.6

Figure 6. Pairwise Master Keys per Second (Integrated Benchmark)

Run #	Single Core	Quad Core	GPU Acceleration
1	7.29	4.25	6.07
2	6.69	3.32	6.19
3	6.62	3.43	6.33
4	6.69	3.72	6.61
5	6.65	3.5	5.93
6	6.38	3.72	6.49
7	6.21	3.32	6.31
8	6.61	3.3	5.9
9	6.62	3.79	6.4
10	6.73	3.7	6.69

Figure 7. Password Recovery of Captured Network Data (time in seconds)

4. DISCUSSION

After conducting the two different tests, several conclusions can be made. Figure 6 clearly shows that both of the hypotheses are supported when performing the integrated benchmark test. The single core program operated at the slowest PMKs per second, followed by the quad-core processor, and finally the GPU accelerated program.

However, the password recovery test showed results that were unsupportive of the first hypothesis. While this test did show that the single core program would operate slower than the quad-core program, it also showed that the quad-core program could recover the password faster than the GPU program. This could be explained by several factors. First, since the GPU accelerated code is run on the GPU, it means that all of the data and instructions have to travel across the PCI-express bus before they can be executed. While that time is likely minimal, it would contribute to the overall time difference. This transmission of data and instructions would include doing memory copy and allocate operations. The dictionary size used in the experiments was quite small, and thus the GPU program likely took a performance hit since the amount of overhead stayed the same even though it moved a small amount of data to the GPU. If the size of the dictionary was larger, the GPU would be able to access a greater amount of data, and the overhead to data ratio would not be as high.

Another reason the GPU accelerated program performed worse than the quad-core program in the password recovery test is likely due to the fact that the amount of passcodes being parsed was not large enough to make up for the time to set up the GPU. The quad-core program was able to execute as soon as it loaded the passcodes into memory. On the other hand, the GPU program had to do the same steps, and then load the passcodes into the GPU memory, essentially doubling the amount of memory accesses. If the number of passcodes in the dictionary were larger, the GPU would be able to make up for those memory accesses by computing the PMKs faster.

Through this experiment, it has been determined that while GPU acceleration decreases the time it takes to recover a password, wireless security protocols are still secure enough to withstand ‘zero information attacks.’ If a business or consumer deploys the maximum key length in their security plan, a malicious attacker would not be able to compromise their network without having a significant amount of time to pre-process the Pairwise Master Keys.

5. ACKNOWLEDGMENTS

Our thanks to Professor Gerald Cichanowski and Minrui Zhang for supervising our project.

6. REFERENCES

[1] "GPU Cracks WPA, WPA2 - WIFI Security (and More) Cracked Using Elcom Distributed Password Recovery Software and NVIDIA GPUs." *Ubergizmo*. Web. 20 Jan. 2012. <http://www.ubergizmo.com/2008/10/wifi-security-and-more-cracked-using-elcom-distributed-password-recovery-software-and-nvidia-gpus/>.

[2] "The Twilight of Wi-Fi Protected Access « Pyrit." *Pyrit*. Web. 02 Feb. 2012. <http://pyrit.wordpress.com/the-twilight-of-wi-fi-protected-access/>.

[3] "Pyrit." *Google Code*. 25 Apr. 2010. Web. 02 Feb. 2012. <http://code.google.com/p/pyrit/>.

[4] Uecke, Jacob, and Hal Berghel. "WiFi Attack Vector." *Digital Village* 48.8 (2005). Print.

[5] *“Nvidia GeForce 9400 GT”* NVIDIA. Web. 30 Mar. 2012 <http://www.geforce.com/hardware/desktop-gpus/geforce-9400-gt/specifications>.

[6] *AMD Phenom II X4 840 - HDX840WFK42GM*. CPU World. Web. 30 Mar. 2012. <http://www.cpu-world.com/CPUs/K10/AMD-Phenom%20II%20X4%20840%20-%20HDX840WFK42GM.html>.

[7] *High-End Mainboards on AMD 890FX: Asus Crosshair IV Formula and Gigabyte GA-890FXA-UD7. Page 4.* Xbit Laboratories. Web. 1 April 2012. <http://www.xbitlabs.com/articles/mainboards/display/asus_crosshair_formulaiv-gigabyte-ga890fxa_ud7_4.html>.

[8] Yang, Chau Chee. "BackTrack 5: Accelerate Pyrit with NVIDIA GPU Processor." *Chau Chee Yang Technical Blog*. 11 Nov. 2012. Web. 01 Apr. 2012. <http://chee-yang.blogspot.com/2011/11/backtrack-5-accelerate-pyrit-with.html>.

[9] Wifidot, “Beware Road Warriors! WPA2 Honeypot APs Might Haunt You.“ Wireless-E-cure. 22 Oct. 2010. Web. 11 Apr. 2012. http://wifidot11.blogspot.com/2010/10/beware-road-warriors-wpa2-honeypot-aps.html

Privacy Issues of Antivirus Apps for Smartphones

Neema Shakya
Dr. Gerald W. Cichanowski, Dr. Mingrui Zhang
WSU Computer Science Department
Winona, MN
682-559-6646
nshakya08@winona.edu

ABSTRACT
There are thousands of antivirus applications available for the Android market. Most of these applications help assure protection of smartphones but they require access to a cellphone's private data and sensitive information. Often users do not know the type of data that is being used. Hence, the safety of data has become a critical issue when it comes to smartphone applications. We take a set of 30 antivirus applications easily available from the Android market and test them using the static and dynamic analysis tools ComDroid and TaintDroid. Based on the analysis, we found that almost all the applications have access to user's private information and contain at least a possible vulnerability.

Categories and Subject Descriptors
D.2.5 [**Software Engineering**]: Testing and Debugging

General Terms
Security

Keywords
Android, antivirus apps, Privacy, TaintDroid, Comdroid, mobile phone security, dynamic analysis, static analysis

1. INTRODUCTION
With the advances in mobile computing, smartphones are becoming widely accepted around the world. The increase in the use of smartphones definitely increases the rate of mobile applications being downloaded from the market. Several of these apps obtain data from remote cloud and use sensor devices like GPS receivers, cameras, microphones, etc. Many users are not aware of how these apps use a phone's private sensitive data. Hence, securing users' private information from third party applications has been a serious problem.

Android is an open source Linux based platform and is used in more than 200 million devices [1]. In early 2011, users have downloaded more than 2.4 billion apps to their Android phones. The number went up to 10 billion on December of 2011 as shown in Figure I. Hence, it becomes very hard to determine if all of these apps are trustable.

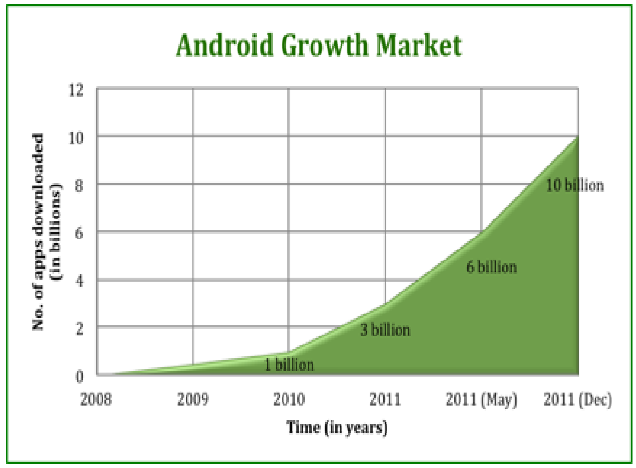


Figure 1. Growth in App consumption for Android Market [3]

A quick search for an antivirus app in an Android market results in the display of almost 1,500 antivirus applications [3]. Most of them are free but a few add extra features for a nominal cost. The antivirus apps promise the overall security of the android phones but hardly a few of these apps detect the malicious software [8]. They gain access to sensitive information like address books, GPS locations, SMS, phone numbers and so on. Some of the antivirus apps have permissions to the change the setting of system tools and also delete or uninstall any android apps.

We take a set of 30 antivirus application easily available from Android market and test them using static and dynamic analysis tools. Static analysis checks possible execution paths in a program code for errors. These tools are used for detecting errors such as buffer overflows, null pointer dereferences, use of uninitialized variables, etc. [7]. We use a static analysis tool named Comdroid developed in the University of California in 2011 [5]. On the other hand, dynamic analysis of a software is performed during its actual execution on a real or virtual device. We use TaintDroid, a dynamic information flow-tracking tool, for our testing.

TaintDroid is a dynamic taint tracking system, which is an extension to the Android phone platform [6]. The primary goal of TaintDroid is to detect the flow of sensitive information through third party applications. This system marks data from the private-

Proceedings of the 12th Winona Computer Science Undergraduate Research Seminar, April 25, 2012, Winona, MN, US.

sensitive sources and whenever these data are transmitted or used by any apps, it tracks down the app and destination where the data is sent. The tracked information is then stored in a log file and presented to the user. Hence, the real-time feedback helps users to detect bad apps and secure sensitive information. TaintDroid was built not long ago by combined efforts of the Intel Lab, Penn State and Duke University [9]. This open source software is easily available online.

Comdroid is a static analysis tool that analyzes Android applications to detect applications communication vulnerabilities [5]. Android's application communication model divides applications into components and provides a message passing system so that component can communicate within and across application boundaries. Android uses Intent, which is a message that declares a recipient and optionally includes data, for its message passing system. These intents are used for both inter-application and intra-application communication and are also used by the operating system as event notifications to the apps [5]. Comdroid analyses the Intent, permission maps and components of the mobile applications. It issues a warning when it detects an implicit intent or whenever a component might be susceptible to an attack. The warning includes the specific method or activity that contains the possible vulnerability along with the line number in the source code and the type of data used.

This research project involves an intense study on the functionality of these two tools and the tests on the set of 30 antivirus apps. The previous research on these tools tested only one antivirus application, hence, this research paper analyses a wide variety of antivirus applications developed for Android phones [5] [6].

2. HYPOTHESIS

In general terms, antivirus software is used to protect the computer systems from any kind of malicious activity and threats. The purpose stays the same even for smartphones or tablets. However, in the context of smartphones, antivirus applications access user's sensitive information and pass them to a third party application.

Do these applications really need access to the locations and SMS content of smartphones? What other sensitive data are being used that the user is unaware of? Is this jeopardizing the privacy of a user? Can users trust these applications?

The permission required by antivirus apps to gain access to a phone's data harms user's privacy.

3. METHODOLOGY

We collected a set of over 30 antivirus applications from the Android Market that is mentioned in Appendix A. These applications were selected based on their ratings and popularity. We conducted static analysis and dynamic testing of the antivirus applications using the tools TaintDroid and ComDroid. The experiments were performed on Google Nexus S running Android version 2.3.4. The log reports generated by each tool were analyzed and based on the reports, the behaviors of antivirus applications were determined. The analysis included the use of users' privacy data and the vulnerability of application communication.

The table in Appendix A contains a list of antivirus apps that were used in our research project. Twenty-one antivirus apps have 4.5 rating out of 5 stars, seven apps have 4 stars and only two apps have 3.5 stars. The most popular antivirus among our thirty apps is the free version of AVG Antivirus, which has more than 220,000 downloads [3]. Antivirus Free by Creative Apps, Avast! Mobile Security, Dr. Web Antivirus Light, Norton Antivirus and Security, and NQ Mobile Security & Antivirus are the other most downloaded antivirus apps in the market. More information about these apps can be found in Appendix A.

3.1 Dynamic Analysis

Each of the above-mentioned antivirus applications went through our dynamic testing tool TaintDroid. All of these applications have access to majority of user's private data. The applications use the mobile device's address book (Contact Provider), phone number, location, accelerometer and SMS. The location used by these apps includes GPS based location, Net–based location and last known location. GPS-based location is the location information offered through the actual device. The GPS system provides the exact position such as latitude, longitude and altitude of the phone on the Earth [10]. On the other hand, Net-based location is the location based on the user roaming networks [10]. This location is determined by using cell tower and Wi-Fi signals when a user moves from one network to other [2].

As shown in table 1, five antivirus applications sent out the IMEI number of the device to the app's content server. The IMEI is a unique identification of a mobile phone and is used to prevent a stolen handset from accessing the cellular network [6]. Two apps transmitted the ICC-ID number, which is a unique SIM card serial number, to the server. None of the apps mentioned the use of this number in their permission section. Two of the antivirus apps send the device IMEI number every time they start up. As shown in table 1, all the 30 applications transmitted the address book (content provider), phone number, locations (both GPS-based and Net-based), accelerometer and SMS data to the application server.

Table 1: Dynamic Analysis Report (TaintDroid)

Applications	Address Book	Phone Number	Location	Accelerometer	SMS	IMEI	ICCID
Aegislab Antivirus	✓	✓	✓	✓	✓		
Aircover - BlueSprig	✓	✓	✓	✓	✓		
Android Antivirus	✓	✓	✓	✓	✓		
Antivirus For Android	✓	✓	✓	✓	✓	✓	
Antivirus Free -Creative Apps	✓	✓	✓	✓	✓		
Avast Mobile	✓	✓	✓	✓	✓		
AVG Antivirus	✓	✓	✓	✓	✓		
Bitdefender Mobile	✓	✓	✓	✓	✓		
Blackbelt Antivirus	✓	✓	✓	✓	✓		
Bluepoint Antivirus	✓	✓	✓	✓	✓		
Clutch Mobile Security	✓	✓	✓	✓	✓		
Comodo Antivirus Free	✓	✓	✓	✓	✓		
Dr. Web Light	✓	✓	✓	✓	✓		
FastScan Antivirus	✓	✓	✓	✓	✓		
Footprint Antivirus	✓	✓	✓	✓	✓		
G Data Mobile Security	✓	✓	✓	✓	✓		
Guardx Antivirus	✓	✓	✓	✓	✓		
Kinetoo Mobile Security	✓	✓	✓	✓	✓		
Mcafee Antivirus	✓	✓	✓	✓	✓	✓	
MT Antivirus	✓	✓	✓	✓	✓		
MYAndroid Protection Antivirus	✓	✓	✓	✓	✓	✓	✓
Norton Mobile	✓	✓	✓	✓	✓		
NQ Mobile Security & Antivirus	✓	✓	✓	✓	✓		
Perfect App Protector	✓	✓	✓	✓	✓		
Privacy Antivirus - Appriva	✓	✓	✓	✓	✓		
SecureBrain Antivirus	✓	✓	✓	✓	✓		
Snap Secure Antivirus	✓	✓	✓	✓	✓		
Trustgo Antivirus	✓	✓	✓	✓	✓		
Webroot Secure Anywhere	✓	✓	✓	✓	✓	✓	✓
Zoner Antivirus	✓	✓	✓	✓	✓	✓	

Table 2: Type of Vulnerabilities

S.N.	Type of Vulnerability	Description
1.	Broadcast Theft/ Sniffing	Application sends an implicit without requiring the receivers to have signature permission; a malicious receiver can easily attack the broadcast Intent and data in the Intent can be leaked.
2.	Activity Hijacking	Application starts an activity implicitly; a malicious activity can intercept the Intent and provide its own UI to the user; data in the Intent can leak and the attacker can steal any input in the UI
3.	Service Hijacking	Application starts a Service implicitly; a malicious service can intercept the service request, steal data content in the Intent and inject false responses back into the sending components
4.	Malicious Broadcast Injection	Receiver is public; a malicious Intent can send an implicit/ explicit Intent to this component; the component trusts Intent data, takes actions or changes state of application
5.	Malicious Activity Launch	Activity is public; a malicious Intent can send an implicit/ explicit Intent to this component; the component trusts Intent data, takes actions or changes state of application
6.	Malicious Service Launch	Service is public; a malicious Intent can send an implicit/ explicit Intent to this component; the component trusts Intent data, takes actions or changes state of application
7.	Action Misuse	Application sends an implicit Intent with a unique action to an internal component; exposes both the Intent and receiving components
8.	Protected System Broadcast w/o action check	A component receives only protected broadcasts, but the component is still publicly accessible to all other application by explicit Intents

Table 3: Static Analysis Report (ComDroid)

Applications	Action Misuse	Activity Hijacking	Sniffing	Malicious Broadcast Injection	Malicious Service Launch	Service Hijacking	Malicious Activity Launch	Protected System Broadcast w/o action check
Aegislab Antivirus		8						
Aircover - BlueSprig	11	18	11	6		1		
Android Antivirus		1		1				
Antivirus For Android		2					1	
Antivirus Free		27	1				1	
Anrivirus Free - AVG	7	22	13	12		3		
Avast! Mobile Security	5	12	8	15		1	3	
Bitdefender Security	5	12	7	6		3		
Blackbelt Antivirus	4	4	1	7	1	4		
Bluepoint Antivirus		19		5		2		
Clutch Mobile Security		13		3		2		
Comodo Antivirus	30	20	17	24				2
Dr. Web Antivirus Light	3	4		6	1	3		
FastScan Antivirus		2						1
Footprint Antivirus		6		2				
G Data Mobile Security		7		2				2
GaurdX Antivirus		22						
Kinetoo Mobile Security		3		1				1
McAfee Antivirus	16	37	7	8	6	17		
MT Antivirus	2	8	1	2				
MYAndroid Protection Antivirus		19		1				
Norton Antivirus	39	21	17	15		13	13	
NQ Mobile Antivirus		19		1				
Perfect App Protector		2		6		1		
Privacy Antivirus Appriva	5	12	4	12				1
SecureBrain Antivirus		2		3				
Snap Secure + Antivirus	23	6	27	10		1	1	1
Trustgo Antivirus & Mobile Security		17	2	8		1	1	
Webroot Secure	10	33	2	15		10	9	
Zoner Antivirus		15	1	7				
Total	**160**	**393**	**119**	**178**	**8**	**62**	**29**	**8**

3.2 Static Analysis

The log report generated from Comdroid show the possible weak spots in the Android applications. The weakest spots in this analysis mean those areas in the source code of the application that contain a possible vulnerability. Comdroid indicates these weak spots as warnings. The warning will also include type of intent, sender and receiver of the intent and the location of where these intents are used. In section 1, the way in which Intents are used for inter-application communications was explained. When the intents are sent to the wrong applications, data contained in them are leaked. Table 2 explains the eight major types of vulnerability that are found in our analysis results.

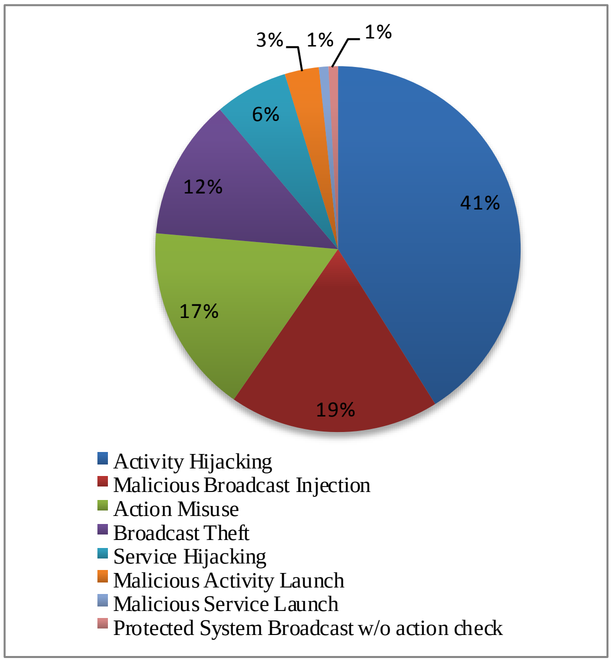


Figure 2. Breakdown of warnings [Comdroid Results]

The numbers in the table 3 above represent the number of weak codes, which can be easily exposed. According to table 3, Norton Antivirus & Security antivirus app topped the ranking list with 118 warnings. The most popular app AVG antivirus has 57 vulnerabilities and 22 possible chances of displaying a fake UI for inputs. McAfee and Webroot Secure antiviruses show maximum activity hijacking warnings. Around 20% of the antivirus apps have more than 50 warnings.

Figure 2 above represents the percentage of the vulnerability from our results. Almost 41% of the warnings were activity hijacking. Hence, the antivirus apps have a greater chance of a malicious activity providing a phony UI and gathering user input as mentioned in table 2. 19% of the vulnerability represented malicious broadcast injection, 17% of the warnings were action misuse and 12% were sniffing or broadcast theft. There were only a few malicious service launch and protected system broadcast without action check vulnerabilities.

4. CONCLUSION

Most antivirus applications for Android phone own full control of the mobile devices. All the 30 antivirus apps we reviewed required access to sensitive data like phone state, identity, SMS and so on. Dynamic analysis showed about 16% of the applications transmitted user private data like IMEI and ICCID numbers to their servers. The designing phase of any android app mentions limiting the use and transmission of private and sensitive data of a phone [2]. The use of unique ID and phone number breaches security and privacy of users.

Static analysis exposed the possibility of attack and weak spots in the antivirus packages we studied. This weak spot could easily lead to leaking of a user's private data. The use of these testing tools is effective in analyzing the reliability of the apps and use of private sensitive data. Hence, the antivirus apps that are developed for protection of a smartphone jeopardize the privacy of a user.

5. ACKNOWLEDGEMENT

We would like to thank our advisors Dr. Gerald Cichanowski and Dr. Mingrui Zhang for their support and advice regarding the research. We thank the WSU Computer Science Department for providing necessary funds to print the posters. We would also like to thank the members of Google discussion board of TaintDroid, and Erika Chin and Adrienne Porter Felt of ComDroid for helping us solve issues regarding the tools used in this project.

6. REFERENCES

[1] Android. http://www.android.com/ 2 Feb, 2012.

[2] Android Developers. http://developer.android.com 15 Apr, 2012.

[3] Android Market. https://play.google.com/store 5 Apr, 2012.

[4] Android – Open Source Project. http://source.android.com 2 Feb, 2012.

[5] Chin, E., Felt, A. P., Greenwood, K. and Wagner, D. 2011. Analyzing inter-application communication in Android. In *Proceedings of the 9th international conference on Mobile systems, applications, and services, MobiSys 2011*, Washington, D.C., June 2011, ACM, New York, NY, 239-252.

[6] Enck, W., Gilbert, P., Chun, B. G., Cox, L. P., Jung, J., McDaniel, P. and Sheth, A. N. 2010. TaintDroid: An information-flow tracking system for real-time privacy monitoring on smartphones. In *Proceedings of the 9th USENIX conference on operating systems design and implementation, OSDI 2010*, Vancouver, BC, Canada, October 2010, USENIX Association Berkeley, CA, 1-6.

[7] Nagappan, N. and Ball, T. 2005. Static analysis tools as early indicators of pre-release defect density. In *Proceedings of the 27th international conference on software engineering, ICSE 2005*, St. Louis, Missouri, May 2005, ACM, New York, NY, 580-586.

[8] Pilz, H. and Schindler, S. 2011. Are free Android virus scanners any good? AV Test. 10 Nov. 2011. http://www.av-test.org/fileadmin/pdf/avtest_2011-11_free_android_virus_scanner_english.pdf 22 Mar, 2012.

[9] TaintDroid 2.3. 2012. TaintDroid 2.3 Build. https://sites.google.com/site/taintdroid23/home 3 Mar, 2011.

[10] Talukder, A. K. and Yavagal, R. R. 2007. *Mobile Computing: Technology, Applications, and Service Creation.* The McGraw-Hill Corporation, Inc., New York, NY.

Appendix A

Antivirus Applications	Vendor	Version	Ratings	Web Links
AegisLab Antivirus Free	AegisLab	1.0.6	4.5 / 1,050	http://www.aegislab.com/Solution/AegisLabAntivirusFree.php?id=6
Aircover - Bluespring	BlueSprig, Inc.	1.0.2	4.5 / 48	http://www.bluesprig.com/aircover.html
Android Antivirus	Android Antivirus	1.6.5	4.5 / 1,938	https://play.google.com/store/apps/details?id=and.anti&feature=search_result#?t=W251bGwsMSwxLDEsImFuZC5hbnRpIl0.
Antivirus For Android	Dala Apps	1.3	4.5 / 212	https://play.google.com/store/apps/details?id=com.antivirusforandroid&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5hbnRpdmlydXNmb3JhbmRyb2lkIl0
Antivirus Free	Creative Apps	1.3.4	4.5 / 51,850	http://zrgiu.com/blog/2011/01/antivirus-free/
Antivirus Free - AVG	AVG Mobliation	2.10.1	4.5 / 219,749	http://www.avgmobilation.com/products
Avast! Mobile Security	AVAST Software	1.0.1892	4.5 / 26,113	http://www.avast.com/en-us/free-mobile-security
Bitdefender Mobile Security & Antivirus	Bitdefender	1.1.692	4.5 / 3,059	http://www.bitdefender.com/solutions/mobile-security-android.html
Blackbelt Antivirus	BlackBelt Antivirus	2.5.0002	4 / 79	http://www.blackbeltdefence.com/product/BlackBelt-AntiVirus
Bluepoint Antivirus	BluePoint Security, Inc.	4.0.25	4 / 631	http://www.bluepointsecurity.com/presentationlayer/pages/mobileedition.aspx
Clutch Mobile Security	Clutch Mobile, Inc.	1.1.11	4 / 29	https://www.clutchmobile.com/
Comodo Antivirus	Comodo Security Solutions	1.2.1831 4.4	4.5 / 714	http://www.comodo.com/
Dr. Web Antivirus Light	Doctor Web, Ltd	7.00.0	4.5 / 51,638	http://download.drweb.com/android
FastScan Antivirus	K-TEC Inc.	1.1.6	4.5 / 13	https://play.google.com/store/apps/details?id=jp.ktinc.fastscan365&feature=search_result#?t=W251bGwsMSwxLDEsImpwLmt0aW5jLmZhc3RzY2FuMzY1Il0.
Footprint Antivirus	Footprint Media	1.2	3.5 /15	https://play.google.com/store/apps/details?id=net.footprintmedia.aav&feature=search_result#?t=W251bGwsMSwxLDEsIm5ldC5mb290cHJpbnRtZWRpYS5hYXYiXQ..
G Data Mobile Security	G Data Software AG	23.4.190 38	4.5 / 914	http://www.gdata.de/
Gaurdx Antivirus	QStar	2.3	4.5 / 3,079	https://play.google.com/store/apps/details?id=org.qstar.guardx&feature=search_result#?t=W251bGwsMSwxLDEsIm9yZy5xc3Rhci5ndWFyZHgiXQ..
Kinetoo Mobile Security	Kaspersky Lab	9.10.108	4 / 1,898	http://usa.kaspersky.com/downloads/free-home-trials/mobile-security
McAfee Antivirus	McAfee	2.0.2.445	3.5 / 2,601	https://www.mcafeemobilesecurity.com/
MT Antivirus	KissDroid	2.0.8	4.5 / 3,197	https://play.google.com/store/apps/details?id=com.hot.free.defence.main&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5ob3QuZnJlZS5kZWZlbmNlLm1haW4iXQ..
MYAndroid Protection Antivirus	MYMobileSecurity LTD	4.6.12.68	4 / 2,362	http://www.mymobilesecurity.com/

Norton Antivirus & Security	Norton Mobile	2.5.0.398	4.5 / 23,690	http://us.norton.com/
NQ Mobile Security & Antivirus	NetQin Mobile Inc.	6.0.06.20	4.5 / 28, 274	http://www.nq.com/
Perfect App Protector	Morrison Smart Software	5.0.4	4.5 / 3,815	http://morrison-soft.blogspot.com/
Privacy Antivirus – Appriva	Moobila Corporation		4.5 / 84	http://www.appriva.com/
SecureBrain Antivirus	SecureBrain	1.0.0.2	4 / 46	http://www.securebrain.co.jp/
Snap Secure + Antivirus	Exclaim Mobility, Inc.	7.28	4 / 325	http://www.snapsecure.net/
Trustgo Antivirus & Mobile Security	TrustGo Mobile Inc.	1.0.4	4.5 / 777	http://www.trustgo.com/
Webroot Security & Antivirus	Webroot Inc.	2.6.0.2217	4.5 / 2,582	http://www.webroot.com/En_US/consumer-products-mobile-security-android-phone.html
Zoner Antivirus	Zoner, Inc.	1.3.1	4.5 / 4,887	http://www.zonerantivirus.com/clanek/android

Lung Cancer Survival Predication with Symptom Cluster of Fatigue, Dyspnea and Cough

Yichen Jiang, Mingrui Zhang, Gerald W. Cichanowski
Department of Computer Science
Winona State University, Winona, MN, USA
yjiang09@winona.edu

ABSTRACT

The study is to show that inclusion of self-reported quality of life symptoms may improve the predication accuracy of lung cancer patients' survival. We analyzed 2405 lung cancer patients' clinical information and developed statistical models to predict patients' survival. The model used patients' self-reported quality of life symptoms including fatigue, dyspnea and cough. The data was collected among lung cancer patients who enrolled in an eight-year prospective longitudinal study at Mayo clinic. Patients' quality of life symptoms were reported within the 1st year of diagnosis.

Keywords
Machine learning, Cox model, Quality of life, Lung cancer, Symptom Cluster.

1. INTRODUCTION

In treating cancers, one of the important facts concerning both physicians and patients is the patients' survival time after cancer diagnosis. To improve the prediction of survival, different statistic models have been developed. A recent research was conducted at Mayo clinic on more than 2400 lung cancer patients. This study analyzed patients' self-reported symptoms, employment status, and physical activity reported in the Baecke questionnaire, and his/her overall quality of life using nested Cox and generalized linear multilevel mixed models. It shows that fatigue and dyspnea are strongly associated with poor clinical outcomes in lung cancer survivors [1].

Computer software is the product that software professionals build and then support over the long term [6]. A Survival Probability Prediction Architecture (SPPA) was developed previously. It consists of a repository of statistical models uploaded by researchers on cancer treatments, and a web-based user interface for physicians and patients [2]. We included this model in the SPPA and developed the role-based user interface.

User interface in the field of human computer interaction is the space where interaction between humans and machines occurs. A role-based view is a concept that implements different views of a database based on the role a user plays. Role-based concept is wildly used in many fields. A popular one is the role based access control (RBAC) [7]. Different user groups will view limited information from the database based on the roles they have. With role-based design, different user interfaces are generated targeting on different user groups. While a physician inputs a patient's clinical information in choosing an optimal treatment, the patient wants to assess how the quality of lives impacts her treatment.

To see if patient self-reported quality of life (QoL) symptoms fatigue, dyspnea and cough within first year after diagnosis may help in improving the prediction accuracy of a patient's survival, we have replicated the model, translated it into a software tool and validated it.

Hypothesis – The inclusion of patient self-reported quality of life symptoms within first year after diagnosis improves the patient's survival prediction.

This paper describes the replication of quality of life (QoL) model in R and the model accuracy measurements. We describe the data collection and QoL model replication in Section 2.1 and 2.2. We discuss the accuracy measurements in Section 2.3. We present the measurement results in Section 2.4 and we present the model based interface for QoL model design and its software implementation in Section 2.6. Then we conclude on Section 3.

2. METHODOLOGY

2.1 Data collection

We acquired 2405 lung cancer patients' data. Patients enrolled in an eight-year prospective longitudinal study with analyzable data. Since 1999, the Epidemiology and Genetics of Lung Cancer Research Program maintains a database for lung cancer patients who agreed to participate in the study and were evaluated at the Mayo Clinic, Rochester, MN. Patients are mailed self-report study instrument within six months of their diagnosis and subsequently on a yearly basis. [3].

Table 1 shows the demographics of reported symptoms of fatigue, cough and short of breath (SOB) in the first year after the diagnosis. The 50% of subjects reported fatigue; 23% of subjects reported cough and 40% reported SOB, respectively. Among all lung cancer patients, 40% of them are NSCLC (Non-small cell lung cancer) and in stage I. 8% are in stage II. 29 % of patients are in stage III or limited-stage SCLC (Small cell lung cancer), 23% in stage IV or extensive stage of SCLC.

Proceedings of the 12th Winona Computer Science Undergraduate Research Seminar, April 25, 2012, Winona, MN, US.

Table 1. Date Demographic

Category	Distribution (%)
Fatigue	50
Cough	23
SOB	40
Male	52.27
Age<=35	0.64
35<Age<=50	8.94
50<Age<=70	55.72
70<Age<=80	27.38
Age>80	7.32
LC stage	
I NSCLC	40
II NSCLC	8
III NSCLC, or limited SCLC	29
IV NSCLC, or extensive SCLC	23

2.2 Replication of Statistical Model

In order to validate the models we developed, we preserved one-tenth of the whole data set, total of 200 patients. Two statistical Cox models were built on the medical records of 2205 patients. The cox model [4], also known as Cox proportional hazards model, is one of the survival models used in statistical analysis of patients' survival. The model provides an estimate of the hazard or risk of death for an individual with provided prognostic variables.

Out of the two models, the first model is the baseline model. It is built on patients' demographics includes gender and age, lung cancer stage, cell types and treatments. The second model, QoL model, includes all the variables used in the first model, symptom cluster and comorbidities. Comorbidity in a medicine condition is either the presence of one or more disorders (or diseases) in addition to a primary disease or disorder, or the effect of such additional disorders or diseases. The comorbidities are measured with the value: cocancer (lung cancer diagnosed with other primary cancers), colung (lung cancer diagnosed with lung related diseases) and coother (Lung cancer diagnosed with all other diseases). A symptom cluster is defined as: fatigue + SOB, fatigue + cough, SOB + cough, fatigue, SOB, or cough.

Both models are implemented using the software R [5]. R is a free software environment under GNU license for statistical computing and graphics. For each patient, each model produces a survival curve.

```
background <- coxph( Surv(time,vitalstatus) ~ stage  +
        gender  +  age  +  celltype + Surgery +
        Radiation + Chemotherapy
            , data=dataf[modelD,])
sxcl <- coxph( Surv(time,vitalstatus) ~ stage  +  gender +
        age  +  celltype + Surgery + Radiation +
        Chemotherapy + Cocancer + colung + coother +
        Fatigue + Coughing + SOB
            , data=dataf[modelD,])
```

Figure 1. R code for construct the model

Figure 1 shows the R programs used for constructing the two Cox models. They use *coxph()* and *Surv()* functions available in the survival package of R. The *Surv()* function is to create the survival object and the *coxph()* function is to build the Cox model. Background object is created for the baseline model; sxcl object is for the QoL model. Variables used in the background model are age, gender, lung cancer stage, cell type and treatment (surgery, radiation, chemotherapy). The QoL model uses additional patient comorbidities, and symptoms (fatigue, cough and SOB).

2.3 Accuracy measurement

We tested the accuracy of the Cox models on 200 reserved patients. Two methods are developed to measure the accuracy; they are depicted in Figure 2. In Figure 2, the *x*-axis is the patient survival time in years and *y*-axis is the survival probability. While the green curve is generated by the baseline model, the black curve is generated by QoL model. The actual vital status for this subject is indicated by an isolated dot A. The subject is deceased at year 3.12 after diagnosis. Point C indicates that the survival rate predicted by the QoL model at year 3.12 is about 0.58; point B indicates the survival rate of 0.67 predicted by the baseline.

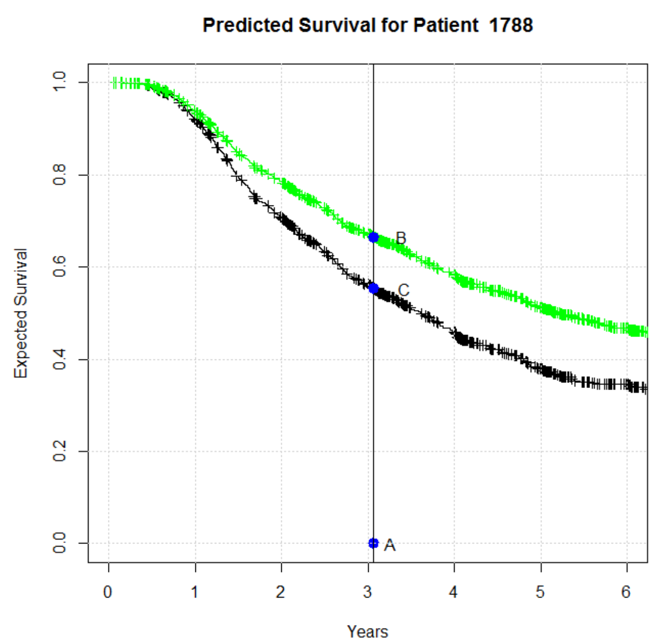


Figure 2. Predicted survival curves for one subject

The first method (value method) measures the value margin between the predictions made by two survival models for each testing subject. In Figure 2, the value is AC - AB, where AC is the difference between actual status and predication of QoL model, AB is the difference between actual status and the prediction of baseline model. In this example, AC-AB is-0.553-(-0.665) =0.112. As the value of AC-AB is positive, we can say that for this test case, QoL model is more accurate.

The second method uses binary coding to assess which model is more accurate. Since the prediction made by QoL model is closer to subject's actual status, we consider QoL model wins (set to 1), baseline model loose (set to 0).

2.4 Result Analysis

```
      qolDistance backgroundDi no.   qol-back
 [1,]    76.68603     73.32572 141  3.3603104
 [2,]    69.78844     67.98407 135  1.8043632
 [3,]    70.19479     66.96648 137  3.2283127
 [4,]    82.09851     79.41561 143  2.6828946
 [5,]    72.70084     70.51441 147  2.1864323
 [6,]    81.08286     77.32407 146  3.7587904
 [7,]    70.09088     69.72990 136  0.3609731
 [8,]    73.72594     70.23561 138  3.4903357
 [9,]    74.74677     73.21190 140  1.5348726
[10,]    74.48385     72.64639 141  1.8374660
[11,]    70.50029     68.64564 139  1.8546578
[12,]    73.11811     69.93932 152  3.1787867
[13,]    63.15395     63.90158 131 -0.7476318
[14,]    64.46960     62.13761 138  2.3319957
[15,]    70.50507     68.65187 132  1.8532072
[16,]    67.83789     64.80882 131  3.0290736
[17,]    75.94756     74.09929 147  1.8482701
[18,]    78.57391     75.50974 151  3.0641674
[19,]    69.20904     68.27139 127  0.9376482
[20,]    75.03858     74.34848 139  0.6901007
> distanceMethod  #qol-background distance. negative is expected
[1] 188.9816
```

Figure 3. Result from first method

Both assessment methods are implemented in R. Each method was repeated 20 times. In each time, 200 testing cases trials were used.

Figure 3 shows the result from the first method. Column *qolDistance* is the sum of the difference between actual statues and prediction of QoL model on all testing cases. Column *backgroundDi* is the sum of the difference between actual statues and the prediction of baseline model. The third column is the actual testing cases are used, and the fourth shows the comparing result. A positive value indicates that the baseline model is more accurate and a negative value indicated that the QoL model is more accurate.

For example, the first row in the Figure 3 shows that the summed differences of QoL model on 200 test cases is 76.69. The summed differences of baseline model are 73.33. The actual test cases used are 141. Other test cases are ignored due to insufficient data information. In comparing to the baseline model, the QoL model is further deviated from the actual status than baseline model by 3.36

The last line of the figure is the sum result of 20 runs. Each run use a 200 testing cases trail. It shows that QoL model is far from the actual status than baseline model about 188.98. It means that the baseline model is closer to the subject actual status than QoL model.

```
      qolBinary backgroundB
 [1,]        57          84
 [2,]        53          82
 [3,]        56          81
 [4,]        59          84
 [5,]        60          87
 [6,]        72          74
 [7,]        69          67
 [8,]        53          85
 [9,]        63          77
[10,]        65          76
[11,]        66          73
[12,]        72          80
[13,]        69          62
[14,]        59          79
[15,]        54          78
[16,]        47          84
[17,]        78          69
[18,]        59          92
[19,]        62          65
[20,]        61          78
> binaryMethod #qol-background binary. positive is expected
[1] -1284
```

Figure 4. Result for second method

The last two columns in Figure 4 show the results of the second assessment method. The qolBinary column shows how many times QoL model wins, and the *backgroundB* column shows how many times a baseline model wins. For example, the first row in the Figure 4 shows that among 200 test cases QoL model has 57 test cases gets more correct prediction, while the baseline model get 84 test cases. There are 59 test cases (200-57-84=59) ignored due to missing information among patients.

The last line of the figure shows the sum results for 20 runs. It shows that the baseline model gets 1284 more correct predictions than the QoL model.

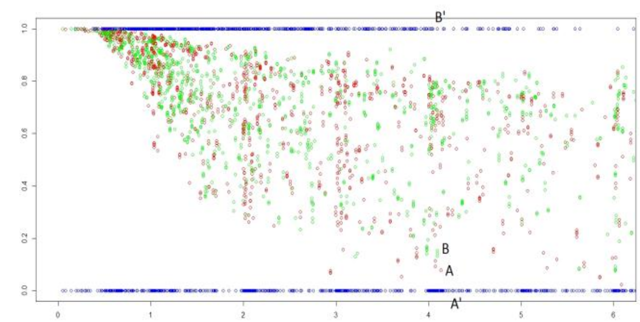


Figure 5. Graphic representation of value measurement result

Figure 5 shows the graphic representation of the results from the first assessment method. A blue dot indicates the actual status of a testing subject, red dot indicates the QoL model predicts more accurately than the baseline model and a yellow dot is the opposite. For instance, a red dot A predicts a survival rate of 0.08 at year 4.34. It is more accurate than the baseline model. The test subject's actual status is represented with blue dot A, who died at year 4.34. In another example, a yellow dot B represents that a testing subject's survival rate about 0.17 at year 4.12 predicted by the baseline model. It is more accurate that the QoL model. This testing subject's actual status is representing by blue dot B, alive at year 4.12.

2.5 Software Development

For further validation by patients, physician, we have integrated the QoL model into an existing web-based lung cancer survival prediction tool. User interfaces (Figures 6-8) were designed based on the roles a user plays. In the first screen (Figure 6), a user inputs patient's age, gender, smoking history, cancer cell type, stage, grade, treatment and if she wants to report self-report symptoms. If she chooses to report symptoms, an interface is displayed to allow the user to enter comorbidities and rate the symptoms using scales from 0 to 10. The scale 0 is the worst, and 10 is the best (Figure 6).

Then the software predicates a patients' survival using the QoL model, and displays her survival probability in the first five years in both graphical and tabular forms. Figure 7 is the graphical view of the patients' survival. It shows the survival curve for each symptom, the background curve without symptom cluster and a curve with symptom cluster. Patients can show/hide each curve by choosing the corresponding checkbox. In the tabular view (Figure 8), it shows the survival predications and their confidence intervals corresponding to each curve displayed in the graphical view.

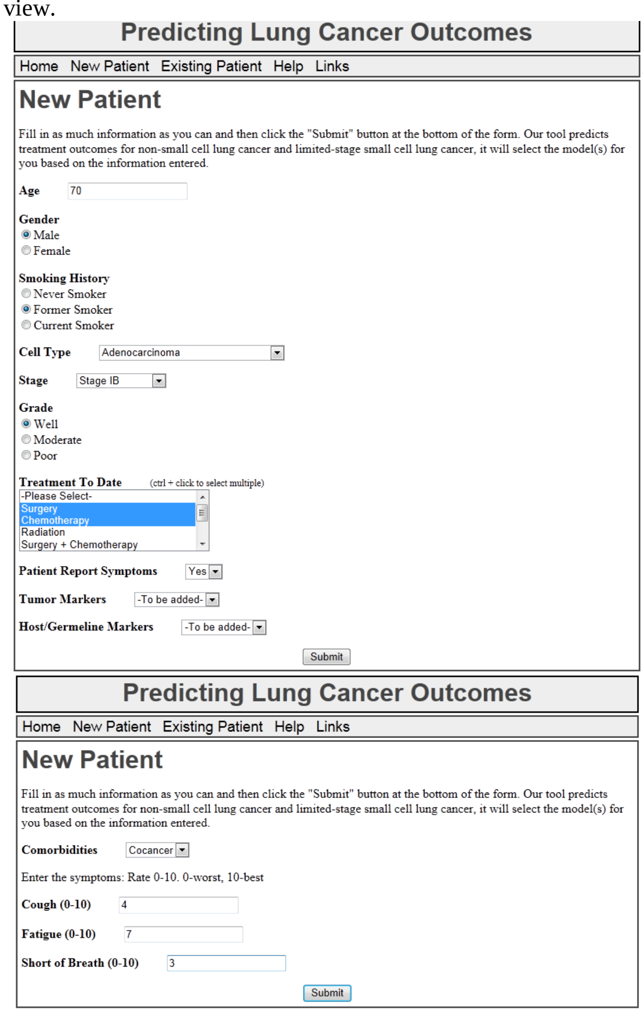


Figure 6. User input interface for QoL model

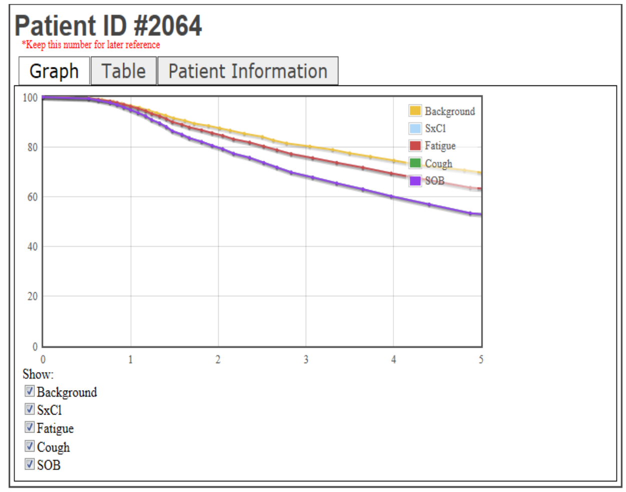
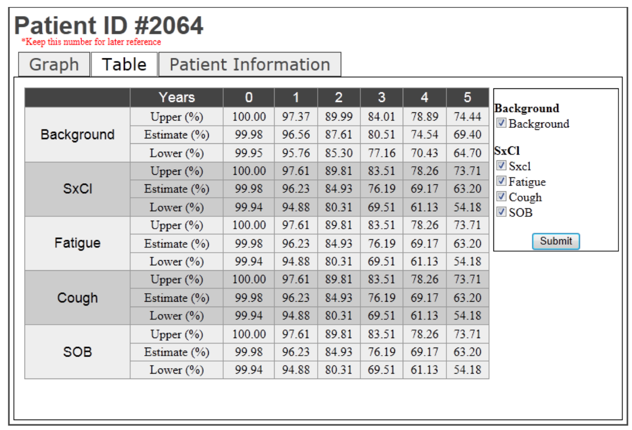


Figure 7. The graphical view of prediction



	Years	0	1	2	3	4	5
Background	Upper (%)	100.00	97.37	89.99	84.01	78.89	74.44
	Estimate (%)	99.98	96.56	87.61	80.51	74.54	69.40
	Lower (%)	99.95	95.76	85.30	77.16	70.43	64.70
SxCl	Upper (%)	100.00	97.61	89.81	83.51	78.26	73.71
	Estimate (%)	99.98	96.23	84.93	76.19	69.17	63.20
	Lower (%)	99.94	94.88	80.31	69.51	61.13	54.18
Fatigue	Upper (%)	100.00	97.61	89.81	83.51	78.26	73.71
	Estimate (%)	99.98	96.23	84.93	76.19	69.17	63.20
	Lower (%)	99.94	94.88	80.31	69.51	61.13	54.18
Cough	Upper (%)	100.00	97.61	89.81	83.51	78.26	73.71
	Estimate (%)	99.98	96.23	84.93	76.19	69.17	63.20
	Lower (%)	99.94	94.88	80.31	69.51	61.13	54.18
SOB	Upper (%)	100.00	97.61	89.81	83.51	78.26	73.71
	Estimate (%)	99.98	96.23	84.93	76.19	69.17	63.20
	Lower (%)	99.94	94.88	80.31	69.51	61.13	54.18

Figures 8. The tabular view of prediction

3. CONCLUSION

Both assessments show that the QoL model is less accurate than the baseline model. Results from the first assessment show that the baseline model without QoL symptoms is more accurate. Results from the 2nd assessment also show that the baseline model makes 1284 more correct predictions than the QoL model.

The self-reported symptoms may not accurately reflect patients' actual status of the symptoms. The symptoms evaluation value can be varied for different patients and patients' status. Therefore, it may affect the accuracy of the QoL model.

4. REFERENCES

[11] Chen, J., et al. "Prognostic factors for limited stage small cell lung cancer: A study of 284 patients." *Lung Cancer* (2010): vol. 67, pp. 221-226.

[12] Zhang, M., et al. "Integrating R Models with Web Technologies." *International Conference on Health Informatics*. Porto, Portugal, 2009.

[13] Cheville, A., et al. "Fatigue, dyspnea, and cough comprise a persistent symptom cluster up to five years after diagnosis with lung cancer." *Journal of Pain and Symptom Management* (2011): 202-212.

[14] Cox, David R. "Regression Models and Life-Tables." *Journal of the Royal Statistical Society* (1972): Series B (Methodological) 34(2): 187-220.

[15] Fox, John and Robert Andersen. "Using the R Statistical Computing Environment to Teach Social Statistics Courses." *Department of Sociology, McMaster University* (2006).