TOUCHMOUSE: DESIGN AND IMPLEMENTATION OF A TOUCHSCREEN BASED PAPER-EQUIVALENT ELECTRONIC FORM

By

Luis A. León-Chacón

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

MASTER OF SCIENCE
in
COMPUTER ENGINEERING

University of Puerto Rico
Mayagüez Campus
2007

Approved by:

Manuel Rodríguez Martinez, Ph.D. Member, Graduate Committee	Date
Pedro Rivera Vega, Ph.D. Member, Graduate Committee	Date
Bienvenido Vélez, Ph.D. President, Graduate Committee	Date
Isabel Ríos, MBA. Representative of Graduate Studies	Date
Isidoro Couvertier, Ph.D. Chairperson of the Department	Date

ABSTRACT

Many regulated manufacturing companies use paper forms to collect daily production data thus becoming burdened with manual data entry and paper-intensive processes yielding considerable inefficiencies for managing and handling thousands of paper forms. By digitizing and automating form-based processes, they can operate efficiently, analyze information effectively and reduce user errors considerably. Badía-Reyes [1] proposed a Paper Equivalent Form framework (PEF) that manages digital forms while keeping a continuous and cryptographically certified audit trail of all changes. PEF provides a user interface rich in buttons and options designed to be used with a mouse or a pen.

Mice require a flat surface and they are inconvenient in certain scenarios such as clean rooms because they can trap dust and dirt or in public places where they can be lost or stolen by users. Touch screen technology can solve some of these inconveniences. However, user interfaces designed for mice and pens are often difficult to adapt to touch screens without considerable modification to the application. Some problems include visual obstruction on the screen by the user’s finger and the difficulty to manipulate small objects. These modifications often require dual versions of the application to be developed and maintained thus significantly increasing development time and costs.

We propose the TouchMouse, a new touch sensitive cursor for touch screens that can combine the advantages of touch screen technology with the high precision of a mouse or pen. By using the TouchMouse, there is no need to make complex modifications to an existing application in order to adapt it to touch screens. We conducted usability studies to calculate the relative efficacy of the TouchMouse versus the Mouse and versus paper forms and we evaluated the learning speed of using the TouchMouse. These studies evidence that users can manipulate electronic forms with the TouchMouse with comparable efficiency to the mouse. They also evidence that users can learn to master the TouchMouse with remarkable ease. The main advantages of the TouchMouse are allow manipulation of small objects, maintain the user gaze in the screen, do not need additional desk space and do not require complex modifications to application.

RESUMEN

Muchas compañías de manufactura usan formularios en papel para almacenar diariamente los datos de producción, sobrecargándose con entrada manual de datos y procesos intensivos que reducen la eficiencia para administrar miles de formularios en papel. Digitalizando y automatizando los procesos basados en formularios, ellas pueden operar eficientemente, analizar efectivamente la información y reducir considerablemente los errores de usuario. Badía-Reyes [1] propuso una aplicación para formularios electrónicos equivalentes al papel (PEF) que maneja formularios digitales y mantiene un historial cifrado de todos los cambios. PEF provee una interfaz rica en botones y opciones diseñada para ser usada con un ratón o un lápiz.

Los ratones requieren una superficie plana y son inconvenientes en ciertos escenarios tales como cuartos limpios porque pueden atrapar polvo y suciedad o en sitios públicos donde se pueden perder o ser robados por los usuarios. La tecnología de pantallas táctiles puede solucionar algunos de estos inconvenientes. Sin embargo, las interfaces de usuario diseñadas para ratones y lápices son a menudo difíciles de adaptar para pantallas táctiles sino se hacen grandes modificaciones en la aplicación. Algunos problemas incluyen obstrucción visual en la pantalla por el dedo del usuario y la dificultad para manipular objetos pequeños. Estas modificaciones a menudo requieren de versiones duales de la aplicación que tienen que ser desarrolladas y mantenidas dando ocasionando un crecimiento significativo de tanto el tiempo de desarrollo como de los costos.

Nosotros proponemos el RatónTáctil, un nuevo cursor táctil sensitivo para pantallas táctiles que puede combinar las ventajas de la tecnología de pantallas táctiles con la gran precisión de un ratón o un lápiz. Con el RatónTáctil no hay necesidad de hacer modificaciones complejas a una aplicación existente para poder adaptarla a pantallas táctiles. Nosotros realizamos estudios de usabilidad para calcular la eficacia relativa del RatónTáctil contra el Ratón y contra el Formulario en papel, y evaluamos la rapidez de aprender a usar el RatónTáctil. Estos estudios muestran que los usuarios pueden manipular formularios electrónicos con una eficiencia comparable a la del ratón. Ellos también muestran que los usuarios pueden dominar el RatónTáctil con una facilidad impresionante. Las principales ventajas del RatónTáctil son: permitir la manipulación de objetos pequeños, mantener la mirada del usuario en la pantalla, no requerir espacio adicional ni cambios complejos en la aplicación.

To my family: my mother Maria de Los Angeles, my father Luis, and my love Janeth.

ACKNOWLEDGMENTS

I would like to thank the Lord. I sincerely thank my family for their love. I thank my love Janeth for all these years of support and understanding. I wish to thank my advisor, Dr. Bienvenido Vélez for his advice, support, patience and friendship. Dr. Manuel Rodríguez and Dr. Pedro Rivera for their help and support.

I thank my friends Edith, Mariana, Iván, Abdiel, Angel and the rest of my partners of the Advanced Data Management Group (ADMG) for their continued support, their company during all those long days. Particular acknowledgment and gratitude to the PEF project partner, René, for all his work and support. I thank all the people who collaborated in the usability test and Dr. Rogelio Palomera and Manuel Jiménez for the workshops. I would also like to thank all other people not mentioned here who have helped me all the way through these years.

This work was supported by the UPRM Digital Government project, under the NSF Grant EIA-0306791.

TABLE OF CONTENTS

LIST OF FIGURES ... **VIII**

LIST OF TABLES ... **X**

1 INTRODUCTION ... **1**

2 PREVIOUS WORK ... **6**

2.1 ELECTRONIC FORMS ... 6
2.2 ELECTRONIC MEDICAL RECORDS ... 7
2.3 TOUCH SCREEN TECHNOLOGY ... 9
2.4 MOUSE TECHNOLOGIES ... 13
2.5 USER STUDIES ... 15

3 DESIGN OF THE TOUCHMOUSE ... **17**

3.1 OVERVIEW ... 17
3.1.1 PEF Project ... 17
3.1.2 TouchMouse ... 22
3.2 EVOLUTION OF THE TOUCHMOUSE ... 37
3.2.1 First TouchMouse ... 37
3.2.2 Second TouchMouse ... 38
3.2.3 Third TouchMouse ... 39
3.2.4 Fourth TouchMouse ... 40
3.2.5 Fifth TouchMouse ... 41
3.2.6 Sixth TouchMouse ... 42
3.3 IMPLEMENTATION OF THE TOUCHMOUSE. ... 43
3.3.1 Simple Algorithm ... 44
3.3.2 Extreme Points Algorithm. ... 49
3.3.3 Moving Average algorithm. ... 50
3.3.4 Score Algorithm. ... 53
3.3.5 Linear Regression Algorithm. ... 55

4 USER TESTING AND RESULTS ... **60**

4.1 INTRODUCTION ... 60
4.2 METHODOLOGY ... 60
4.2.1 Objective ... 60
4.2.2 Computing Infrastructure ... 61

4.2.3 Participants ... *65*
4.2.4 Experimental Design ... *66*
4.3 STATISTICAL ANALYSIS ... 67
4.3.1 Latin Square design ... *67*
4.3.2 Analysis of Variance (ANOVA) ... *70*
4.4 PILOT TEST RESULTS. ... 70
4.5 USER TEST RESULTS. ... 71
4.5.1 Initial tasks ... *71*
4.5.2 Training tasks ... *72*
4.5.3 Testing tasks ... *77*
4.6 COMPARATIVE RESULTS ... 83
4.7 RESULTS OF ALGORITHMS. ... 84
4.7.1 Simple Algorithm. ... *84*
4.7.2 Extreme Points Algorithm. ... *84*
4.7.3 Moving Average Algorithm. ... *85*
4.7.4 Score Algorithm. ... *85*
4.7.5 Linear Regression Algorithm. ... *85*
4.8 RESULTS OF QUESTIONNAIRES ... 86

5 CONCLUSIONS AND FUTURE WORK ... 89

5.1 RESEARCH CONCLUSION ... 89
5.2 FUTURE WORK ... 91

APPENDIX A: QUESTIONNAIRE "GENERAL INFORMATION" ... 97

APPENDIX B: "CONSENT FORM AND INFORMATION LETTER" ... 99

APPENDIX C: "ORIENTATION SCRIPT" ... 101

APPENDIX D: QUESTIONNAIRE "PEF CLIENT INTERFACE" ... 102

APPENDIX E: INITIAL TASKS ... 104

APPENDIX F: TRAINING TASKS ... 105

APPENDIX G: TESTING TASKS ... 110

LIST OF FIGURES

FIGURE 1. TOUCHMOUSE 4
FIGURE 2. PEF ARCHITECTURE [1]. 18
FIGURE 3. AN EXAMPLE OF AN ELECTRONIC FORM [1]. 21
FIGURE 4. ARROW POINTER OF THE TOUCHMOUSE. 22
FIGURE 5. NUMBERED POSITIONS FOR THE ARROW POINTER IN THE TOUCHMOUSE. 23
FIGURE 6. AUTOMATIC ARROW POINTER. 24
FIGURE 7. ROTATION AREA OF THE TOUCHMOUSE. 25
FIGURE 8. MOVE/POINT AREA OF THE TOUCHMOUSE. 25
FIGURE 9. TOUCHMOUSE OVER THE WELCOME SCREEN IN PEF [1]. 26
FIGURE 10. OPEN A FORM IN PEF [1] USING THE TOUCHMOUSE. 28
FIGURE 11. THE TOUCHMOUSE CHOOSING YEAR 2007 IN PEF [1]. 29
FIGURE 12. THE TOUCHMOUSE CHOOSING MONTH OCTOBER IN PEF [1]. 30
FIGURE 13. THE TOUCHMOUSE CHOOSING LINE 1 AND SHIFT 2 IN PEF [1]. 31
FIGURE 14. THE TOUCHMOUSE IN THE OPENED FORM IN PEF [1]. 32
FIGURE 15. TOUCHMOUSE OVER THE NEW FORM BUTTON IN PEF [1]. 34
FIGURE 16. TOUCHMOUSE SETTING A DATE FOR A NEW FORM IN PEF [1]. 35
FIGURE 17. NEW FORM IN PEF [1] CREATED USING THE TOUCHMOUSE. 36
FIGURE 18. FIRST DESIGN OF THE TOUCHMOUSE. 37
FIGURE 19. SECOND DESIGN OF THE TOUCHMOUSE. 38
FIGURE 20. THIRD DESIGN OF THE TOUCHMOUSE. 39
FIGURE 21. FOURTH DESIGN OF THE TOUCHMOUSE. 40
FIGURE 22. FIFTH DESIGN OF THE TOUCHMOUSE. 41
FIGURE 23. SIXTH DESIGN OF THE TOUCHMOUSE. 42
FIGURE 24. CONTIGUOUS POINTS. 44
FIGURE 25. SIMPLE ALGORITHM. 45
FIGURE 26. EXTREME POINTS. 49
FIGURE 27. EXTREME POINTS ALGORITHM. 50
FIGURE 28. ALL POINTS. 50
FIGURE 29. MOVING AVERAGE ALGORITHM. 52
FIGURE 30. DIRECTION TO SCORE. 53
FIGURE 31. SCORE ALGORITHM. 54
FIGURE 32. LINEAR REGRESSION MODEL. 56
FIGURE 33. LINEAR REGRESSION IN A SCREEN. 57
FIGURE 34. LINEAR REGRESSION ALGORITHM. 58

FIGURE 35. TOUCH SCREEN. 62
FIGURE 36. PAPER-BASED FORM. 64
FIGURE 37. PIXEL POSITIONS. 71
FIGURE 38. TRAINING TASK COMPARISON. 73
FIGURE 39. NORMAL PROBABILITY PLOT OF TRAINING TASKS. 75
FIGURE 40. TEST FOR EQUAL VARIANCES FOR TIME WITH THE TRAINING TASKS. 76
FIGURE 41. TESTING TASK COMPARISON. 78
FIGURE 42. NORMAL PROBABILITY PLOT OF TESTING TASKS. 79
FIGURE 43. TEST FOR EQUAL VARIANCES FOR TIME WITH THE TESTING TASKS. 80
FIGURE 44. TRAINING AND TESTING TASK COMPARISON. 83
FIGURE 45. RESULTS FROM QUESTIONNAIRES. 87

LIST OF TABLES

TABLE 1. ANGLES FOR EVERY POSITION OF THE ARROW POINTER. 47
TABLE 2. LATIN SQUARE DISTRIBUTION FOR THE TEST. 69
TABLE 3. LATIN SQUARE DISTRIBUTION WITH THREE USERS AND THREE TASKS. 69
TABLE 4. MEAN AND STANDARD DEVIATION FOR INITIAL TASKS. 72
TABLE 5. MEAN AND STANDARD DEVIATION FOR THE TRAINING TASKS. 72
TABLE 6. MEAN AND STANDARD DEVIATION FOR EACH DEVICE IN THE TRAINING TASKS. 73
TABLE 7. MEAN AND STANDARD DEVIATION FOR TESTING TASKS. 77
TABLE 8. MEAN AND STANDARD DEVIATION FOR EACH DEVICE IN THE TESTING TASKS. 77
TABLE 9. ONE-WAY ANOVA FOR THE TESTING TASKS – TIME VERSUS TASK. 80
TABLE 10. ANOVA FOR THE TRAINING TASKS – TIME VERSUS DEVICE. 81
TABLE 11. ANOVA FOR THE TRAINING TASKS – FACTORS AND LEVELS. 81
TABLE 12. ANOVA - GENERAL LINEAR MODEL: TIME VERSUS TASK, DEVICE, REPLICA. 82
TABLE 13. PERCENTAGE OF AVERAGE TIME TO PERFORM A TASK. 82
TABLE 14. IMPROVEMENT FROM TRAINING TO TESTING. 83
TABLE 15. RESULTS FROM QUESTIONNAIRES. 87

1 INTRODUCTION

Many regulated manufacturing companies use paper forms to gather data from processes, customers, employees, and vendors. They keep data such as audits, deviations, specifications, corrective/preventive actions, and general information. Consequently, paper forms have become the primary interface in many organizational processes, thus companies are burdened with manual data entry and paper-intensive processes. For example in the pharmaceutical industry to keep records of every process in paper-based forms is required by the Food and Drug Administration (FDA)[1]. One particular case in this industry is the packaging process, where every process deals with information about operators, shifts, lines, machines, number of units, productivity, and many others. However, problems appear when it is required to share and retrieve information; because that information is stored in papers that in some cases are stored away from the plant.

In that case, time consuming and complex manual tasks associated with managing and handling each paper form are reducing efficiency, productivity. To be effective, form-based processes should be able to adjust readily to different conditions for fulfilling a company's organizational needs. They should be efficient in getting general information and acceptance from everyone involved, and equipped to allow collaboration among several people or departments. By

[1] Food and Drug Administration. December 2007. http://www.fda.gov/

digitizing and automating form-based processes, organizations can operate more efficiently, reduce error rates, and improve the speed and quality of their services. Thus, with information stored digitally, it is possible to use tools for analyzing the information, as well as apply human computer interaction techniques to facilitate learning and reduce user errors.

On the other hand when regulated organizations want to migrate the paper-based processes to digital media they most abide by strong regulations that attempt to guarantee the confidentiality of all the information stored digitally. One example is the Food and Drug Administration (FDA) 21 CFR Part 11 regulation [3]. The 21 CFR Part 11 states the “criteria under which the agency considers electronic records, electronic signatures, and handwritten signatures executed to electronic records to be trustworthy, reliable, and generally equivalent to paper records and handwritten signatures executed on paper”.

Badía-Reyes [1] proposed a Paper Equivalent Form framework (PEF). PEF receives data from forms and keeps an audit trail of data as it is filled out in electronic forms. When changing the value of a field in a form the user must provide proof of its identity. A certificate that recognizes the change is asked for a certification authority (CA), to be able to review the form at any time. PEF provides means to compare information, to verify changes, and/or to discover alterations and meets the requirements expressed in the 21 CFR Part 11.

PEF have an interface rich in buttons and options that requires for interaction to use an input device with high accuracy like a mouse or a pen. But, sometimes it is not feasible to have the area for a full computer solution like a desktop for a monitor, mouse and keyboard. Also, there are many situations where a mouse is inconvenient like public places, crowded places, and clean rooms because they can be lost or trap dust and dirt. Thus, by integrating the input device with the display, valuable workspace can be saved. In that case, a touch screen technology will be very useful, because with a touch screen interface, operators can supervise and manage difficult operations in real-time by simply touching the screen. Although touch screen technology has been around for years, the high cost of touch screen monitors has prevented this technology from being massively adopted. Today, a touch screen is no longer cost prohibitive allowing any company to take advantage from its benefits.

However, we found out that PEF forms were difficult to manipulate using touch screens. This is common in most touch screen applications where there is visual obstruction of some information on the screen by the user finger. Also, with the finger is difficulty to manipulate small objects. In fact, to make an application over a touch screen work properly, designers must ensure that the large surface of a finger user can point to a single place on the screen. To compensate this problem, designers must make the controls and buttons in the touch screen bigger than the finger width, thus it is required to make modifications in the interface or design and develop another application. This increases development time and costs since different versions of the same application must be designed and maintained.

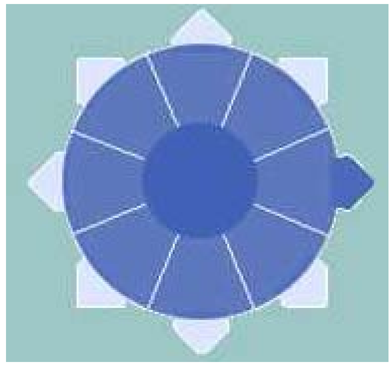

Figure 1. TouchMouse

Our thesis is the TouchMouse such as the one shown in Figure 1. The TouchMouse can combine the advantages of touch screen technology with the high accuracy of a mouse or pen device. By using the TouchMouse, there is no need to make any changes in an existing interface. This is great because we can have a single implementation of the interface for different pointing devices such as mouse, pen or the finger with the TouchMouse. A single implementation implies the TouchMouse can be used where it is not possible to make changes in applications, either because of the lack of the source code or because it is not cost-effective.

In summary of the TouchMouse offers the following advantages:

- The TouchMouse allows manipulation of small objects
- The TouchMouse maintains the user gaze in the screen
- The TouchMouse does not require additional desk space
- With the TouchMouse is not required to make modifications in the existing interface or application

In summary the Contributions of this research are:

- To provide another option to interface with electronic forms that support usability by using touch screen technology
- Designed the TouchMouse and implemented using the Java[2] [12] programming language and its GUI facilities
- Developed automatic arrow pointer algorithms for the TouchMouse
- Conducted a user study to evaluate the efficacy of using the TouchMouse to manipulate the same electronic forms designed for the mouse

Our experiments ensure that the TouchMouse offers high performance comparable to the mouse. Moreover, we discovered that users are able to learn and adapt to the new TouchMouse device with remarkable rate.

The rest of this document is organized as follows. Chapter 2 presents the previous work related to this research. Chapter 3 describes the design of the TouchMouse. Chapter 4 presents the methodology, tests, statistical analysis and results. Chapter 5 summarizes our findings and achievements and also presents some ideas for improving our current work. Finally, the Appendixes give supporting information used for the usability test.

[2] Java programming language. December 2007. java.sun.com

2 PREVIOUS WORK

This section presents relevant publications and previous work related to our research.

2.1 Electronic Forms

With the advances in technology and science, electronic forms are an accepted option to replace paper forms, because they help to save paper and reduce management and processing costs for a department. Also, paper forms require large amount of storage space and they can be destroyed, misplaced, or needlessly duplicated. Electronic forms store data digitally and they often facilitates sharing and replication of the information.

Futo [4] presents an example of electronic forms called A General Purpose Front-Office System for eGovernment Communications. It consists of three components: the form generator, the client module and the server module. Two kinds of communications are supported: communication based only on PIN code and password, and communication using also electronic signature. The form generator consists in a graphical design of the form and a design of the algorithms (relations between fields, calculation formulas); also, generation of the downloadable program and publication on the Internet (portal). The client module handles the filling the form and sending the data electronically via the Internet. The server module consists of the portal and processing system. The tasks of the portal are ensuring the authentication and authorization of the documents. The processing system checks the incoming documents and converts them to a predefined format. The converted data is then loaded into the back-office system.

Badía-Reyes [1] presents “A Framework for Auditable, Paper-Equivalent Electronic Forms using Data Logging, Secure Access Controls, and Electronic Certificates”. This framework for electronic forms is as trustworthy as physical paper and provides additional security procedures and data logging mechanisms that allow verifying the authenticity of the information stored within the forms and whether the data has been tampered with or not. This project will be more explained in session 3.1.1.

2.2 Electronic Medical Records

A typical example of electronic records is an Electronic Medical Record (EMR). EMR has been seen as the answer for solving the need for timely and location-independent access to comprehensive patient data. This data can be integrated with respect to type (clinicians notes, medical imaging, patient records, dose of some medication prescribed, time to take medications, charts, etc.) and time (a single patient centered record of each and every contact between health care service providers and the patient) [7] [21]. With the growing demand for greater coordination and cooperation between different health care services, these attributes became a powerful incentive in the acceptance of the EMR.

Some of the difficulties experienced in using the EMR as a medium for the achievement of service integration are the classification of organizations and the amount of services involved. For large organizations with complex information needs, achieving even low levels of integration can be difficult in practice, because they exhibit complexities related to scale, distinct roles, quantity of information, and distinct processes within the organization. Information exchange practices and systems are intrinsic in local work processes as well as wider patterns of communication and coordination. Efforts to modify practices and redefine roles and relationships may lead to resistance. Definitely, reservations about the consequences of electronic media on everyday medical work have been increased by numerous studies that point to the unique, multiple advantages of paper-based records for viewing, reviewing, annotating and amending data [7].

There are many works that present Electronic Medical Records as an opportunity to reduce costs and a powerful tool for sharing information. Wong [21] presented two simulation models to quantify the advantages of an electronic medication ordering, dispensing and administration process compared with the current manual process at an acute care academic health sciences center. The first model represented a manual system, and was validated against observed data. The second model represented their proposed electronic medication ordering, dispensing and administration system. Our user studies take some ideas from this work.

Hartswood [7] stated that the introduction of the electronic medical record (EMR) is widely seen by health care policy makers and service managers as a key step in the accomplishment of more efficient and integrated health care services. But, their study of inter service work practices revealed significant discrepancies between the presumptions of the role of the EMR in achieving service integration and the ways in which medical workers actually use and communicate patient information.

2.3 Touch Screen Technology

With the widespread availability of pen-based computers, such as the Tablet PC and pen based Personal Digital Assistants (PDA), touch screen technology is becoming ubiquitous and mainstream. User interface designers increasingly face the challenge of designing effective pen-based user interfaces for this class of devices [2]. We now describe some examples of interfaces designed for touch screen technology.

Electronic Cocktail Napkin [5] is a pen-based interface implemented on a Tablet PC. The interface acquires information about ambiguity and precision from freehand input, represents it internally, and echoes it to users visually and through constraint based edit behavior. Its goal was to support the kind of informal drawing that designers do on the back of an envelope or a cocktail napkin during conceptual design. To achieve that goal they explored a general purpose drawing environment that end users can make more specific through interface techniques used in a pen based collaborative design environment. It requires a pen connected to the Tablet PC.

Kurtenbach [13] presents the design of a GUI paradigm based on Tablets, Two hands and Transparency called T3. T3 was based on maximizing the amount of screen used for data application, reducing the amount of User Interface (UI) that distracts visual attention from the application data, and increasing the quality of input. To achieve these goals, they integrated the nonstandard UI technologies of multi-sensor tablets, tool glass, transparent UI components, and marking menus. Although, it does not requires a pen connected to the Tablet PC, it is only for drawing images.

Matsui [14] proposed a text input method for pen-based computers. He used dynamic queries on a dictionary and word prediction from context to filter the results from the dictionary and select a word with the pen. He used this idea instead of composing the word by entering characters one by one. His results show that with his method, the speed of text input was increased on pen-based computers and it becomes a viable alternative to use a keyboard on a screen. It requires a pen connected to the Tablet PC and it is only for inserting text.

Hinckley and Sinclair [8] demonstrated that by adding touch sensors to input devices it is possible to offer many alternatives for novel human interaction techniques. They presented the Touch Trackball and the Scrolling TouchMouse, which use sensors to detect contact from the users hand without requiring pressure or mechanical actuation of a switch. They also demonstrated how the capabilities of these devices can be matched to an inherent interaction technique; first one is the On-Demand Interface, which uses the passive information captured by touch sensors to fade in or fade out portions of a display depending on what the user is doing; a second technique uses explicit, intentional interaction with touch sensors for enhanced scrolling.

They presented some similarities between Touch Sensors and Touch Tablets [8]:

- No moving parts.

- No mechanical intermediary to activate them.
- Operation by feel: Touch sensors can be arranged into regions that act like a physical template on a touch tablet.
- The user can feel the touch-sensing regions without looking at the device or at the screen. This can reduce the time that would be required to switch between devices or widgets on the screen.
- Feedback: Touch sensors differ from traditional push buttons in the type and quantity of feedback provided. While with a mouse button, the user does not feel or hear a different click when a touch sensor is activated.

Although, it does not require a pen connected to the PC and it does not use a touch screen, it requires a Touch-Trackball and Scrolling TouchMouse. The Touch-Trackball and Scrolling TouchMouse are hardware implementations where the Touch-Trackball is a modified Kensington Expert Mouse[3] and the Scrolling TouchMouse is a modified Microsoft IntelliMouse Pro[4].

McClard and Somers [15] presented a work where a cable company gave families pen-based tablet computers with a wireless connection to their high-speed data network. They were trying to understand how web access from a tablet changes the way people use the Internet. They used ethnographic and usability methods for understanding how tablets should be incorporated into

[3] Kensington Expert Mouse. December 2007. http://us.kensington.com/html/2200.html

[4] Microsoft IntelliMouse Pro. December 2007. www.microsoft.com/hardware/mouseandkeyboard

household activities. Also, they used the information to specify user requirements for these devices. Their results show that participants viewed the tablet: comfortable, portable and as a toy for fun and playing, but the PC was viewed as less comfortable as a tool. These results suggested that correctly designed Internet applications on portable devices filled a special role in daily life, because they allowed access to information and communication anywhere and anytime. Their system requires a pen connected to the Tablet PC.

Tracking Menu [2] consists of a cluster of graphical buttons the cursor can be moved within the menu to select and interact with any item and the menu always stays under the cursor. They examine one tracking menu design in detail, reporting usability studies and their experience integrating tracking menus into a commercial application for a Tablet PC. It requires a pen connected to the Tablet PC.

Yee [22] presented the combination of a touch screen and a tablet display to support asymmetric two-handed interaction in which the preferred hand used a stylus and the non-preferred hand operated the touch screen. The author described a method for tracking the independent motions of both hands and his results showed that when the non-preferred hand performed elementary actions and the preferred hand performed complex actions then users felt comfortable, natural, and efficient to work with both hands. It requires a pen connected to the Tablet PC.

Po, Fisher and Booth [18] presented an evaluation of four cursor orientations and a neutral orientation of the cursor in a circular menu selection task. Mouse interaction on a desktop, pointer interaction on a large screen, and pen interaction on a Tablet PC were evaluated. Authors provide some advice for the appropriate use of cursors in various input and display configurations and their results propose that choosing suited cursors is quite important for pointer interaction. It requires a pen connected to the Tablet PC.

Hover Widgets [6] is a technique for enhancing the capabilities of pen-based interfaces. Hover Widgets were implemented in the tracking state by using the pen movements above the display surface. The tracking state was used to monitor the current position of the cursor and it sensed the pen location while the pen was proximal to the interaction surface. By using the tracking state movements, Hover Widgets created a command layer which was distinct from the input layer of a pen interface. It requires a pen connected to the Tablet PC.

The TouchMouse presented in this research only requires the touch screen and it is an all software implementation. Although, TouchMouse does not need a pen, mouse or any special input device, the user can use his/her finger, nail or any object to use the TouchMouse. The TouchMouse is explained in section 3.

2.4 Mouse Technologies

Mechanical Mice require that the mouse be set on a flat surface. The distance and the speed of the rollers inside the mouse determine how far the mouse cursor moves on the screen depending on the software configuration; it relies on the traction between the mouse ball and the rollers. Optical Mice require a special mouse pad which has a grid pattern. A sensor inside the mouse determines the movement by reading the grid as the mouse passes over it while emitting a light from an LED or sometimes a laser. This type of mouse is much more accurate than the ordinary optical mechanical mouse. One drawback to an optical mouse is they can have problems in bright lights. New Optical Mice do not have the disadvantages of earlier mice and are capable of being utilized on any surface. In comparison to the traditional Optical-Mechanical mouse, the Optical is a much better solution for a computer mouse.

Optical-Mechanical Hybrid Mice consists of a ball which rolls a wheel inside the mouse. This wheel contains a circle of holes and or notches to read the LED by a sensor as it spins around when the mouse is moved. This mouse is much more accurate than the mechanical mouse. This mouse is now the most commonly used mouse with PC and Macintosh computers.

Inconveniences of the computer mouse

Regardless of the fact that computers are becoming more frequently embedded into everyday activities, when a person has no previously experience with a computer mouse then it is necessary to train that person. The training is to overcome the deficiency in the ability to master the use of this interface device.

- One problem for any new user arises when the monitor of the computer is in the vertically plane and the mouse is on a horizontal table, so that the difference between the two planes is 90°. This causes an orientation difficulty for new computer users in some cases.
- The physical distance between the computer mouse, which must be clicked to activate elements in the computer, and the pointer on the

screen, which represents the location of the mouse, can cause difficulty in orientation.

- A computer mouse reports relative motion since the mouse can be picked up and placed down on another location without any change in position of the designation on the computer screen. This also produces confusion for new computer users
- When a novice user needs to move the mouse in the screen but physically he/she can not continue moving the mouse because there is no more space on the desktop, it is possible that he/she could not guess it is necessary to lift the mouse and put in another location and then continue the movement
- Additionally, it is possible that a new computer user could attempt to move the mouse by lifting it from the surface and try to move it
- Mice must be placed on a flat
- Mice are inconvenient for use in public places such as kiosks since they can be stolen or inadvertently taken away

2.5 User Studies

Usability is defined as the ease of use and acceptability of a system for a particular class of users carrying out specific tasks in a specific environment [9]. Usability problems predicted by evaluation techniques are valuable in system development. Usability studies helps to identify and rectify usability deficiencies existing in computer based applications and their supplemental material [10][17][19].

Well designed user studies ensure the creation of products that are easy to learn and use, are satisfying to use and provide utility and functionality for their users [17][19].

A pilot test is a preliminary test procedure with a few user subjects. During pilot testing is possible to find if the instructions for some tasks are incomprehensible to the users or that they misinterpret them. It can also be used to refine the experimental procedure and to clarify the definitions of various things that are to be measured. The pilot test subjects may be chosen for convenience among people who are easy available to the experimenter [17][19].

3 DESIGN OF THE TOUCHMOUSE

This chapter begins with a brief description of the Paper Equivalent Forms (PEF) framework and proceeds to describe the evolution and implementation of the TouchMouse.

3.1 Overview

We used The Paper Equivalent Form Framework (PEF) [1] as case study to identify potential usability problems, interaction differences, advantages, and disadvantages of two interaction techniques: the Mouse and our TouchMouse.

3.1.1 PEF Project

The Paper Equivalent Form framework (PEF) [1] receives data from forms and keeps an audit trail of data as it is filled out in electronic forms. When changing the value of a field in a form the user must provide proof of its identity. A certificate that recognizes the change is asked for a certification authority (CA), to be able to review the form at any time. This framework provides means to compare information, to verify changes, and/or to discover alterations.

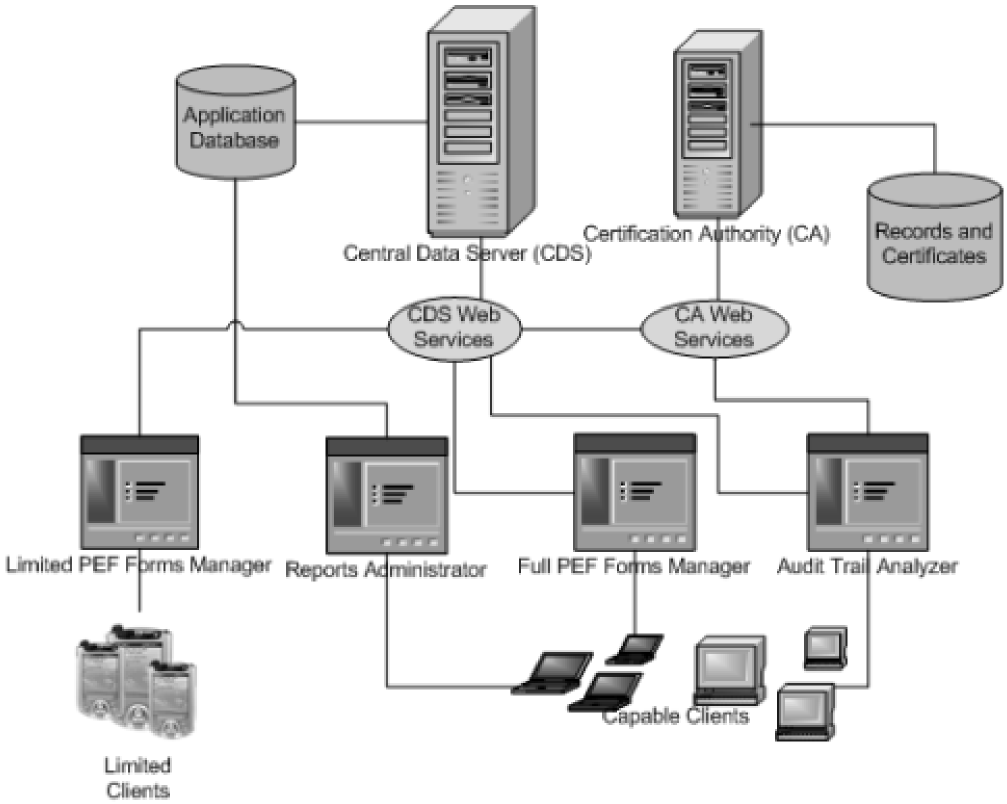

Figure 2. PEF Architecture [1].

We now briefly describe the main components of the PEF framework shown in Figure 2. A more detailed description of Badía-Reyes' work is in [1].

3.1.1.1 Central Data Server

The Central Data Server (CDS) provides a centralized repository of all forms and it is also in charge of coordinating requests from the clients. Whenever a client accesses the framework to create a new form or to update an existing one, it will make a request to the Central Data Server who will consult the database and conduct the appropriate action based on the request. The CDS uses XML Web services to handle requests of the clients. These services provide the following functionality [1]:

- Create and verify the validity of a user in the database authenticated by a password or by a fingerprint.
- Create forms or templates in the database.
- Store a new certificate in the Database when a field in a form changes.

3.1.1.2 Certification Authority

The Certification Authority (CA) is a reliable, external server who has the functionality of being able to review the data on the Application Database by means of an electronic certificate using cryptographic algorithms. For this reason, the Certification Authority has a database that contains the information on the certificates that have been received from the Form Manager of PEF [1].

The Certification Authority also uses the XML Web services to handle requests of the Central Data Server. These services provide the functionality to create a certificate for a record based on the characteristics and the value of the record.

3.1.1.3 Forms Manager

The PEF Forms Manager (FM) allows the user to handle the forms and enter data [1]. The full and limited FM in PEF refers about the equipment that the user can use to interact with the framework; capable clients are users working with computers or workstations with high processing capacity, limited clients are those who are using portable equipment with a limited processing capacity or with small screens such as PDAs.

In addition, the Forms Manager is in charge of the user access methods and of controlling security. The user logins to the PEF Forms Manager by an authentication form. After login, the user can use the Forms Manager to access the forms, to open, to change data, to create templates and forms. When a user changes the value of any field, the framework requests the user to confirm or cancel the change. If the user confirms it, the record is sent to the Central Data Server, which will make a request to the Certification Authority to produce a certificate for the record. After the Certification Authority sends the certificate to the Central Data Server, the record, along with its certificate, can be stored in the Application Database. A registry of changes made for each field stays in each form. The date, the time, the author, and the value of each record are kept in a perpetual audit trail.

3.1.1.4 Audit Trail Analyzer

The Audit Trail Analyzer component is used to review the data on the Application Database. It can do it by comparing the data from the forms against the data stored in the Certification Authority. It validates that the records and their related certificates stored in the Application Database match, that is it refers to the information has not changed by means others than from the Forms Manager [1].

3.1.1.5 Example

Figure 3. An example of an electronic form [1].

Figure 3 shows an example of a form for a packaging process created using PEF [1]. We use the data from Appendix F for this example. The form includes date of creation, line and shift. The upper part of the form shows number of reference, name of product, lot number, number of units and name of the machine. The graph in the center of the form shows the performance achieved by the operators for every hour in the shift. At the bottom of the form is the information about different events from packaging that occurred during the shift that lasted more than 10 minutes. In section 3.1.2.3, we explain how to use the TouchMouse to manipulate this form.

3.1.2 TouchMouse

The TouchMouse is a new type of interactive cursor designed for touch screens. The TouchMouse facilitates direct manipulation of small objects which are very difficult to manipulate using a finger. Now, we present a brief description of the TouchMouse, its features and how to use it.

3.1.2.1 Anatomy of the TouchMouse

In this section we describe the main visual components of the TouchMouse.

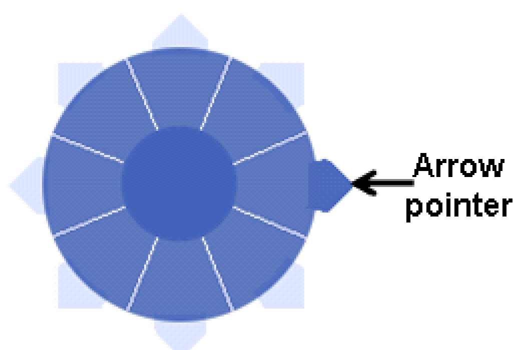

Figure 4. Arrow pointer of the TouchMouse.

Figure 4 shows the arrow pointer of the TouchMouse used as a reference point to interact with graphical user interfaces (GUI). Thus, when the user wants to interact with a component in a GUI, he/she just needs to move the TouchMouse until arrow pointer touches that component. You can click by tapping the Move/Point area.

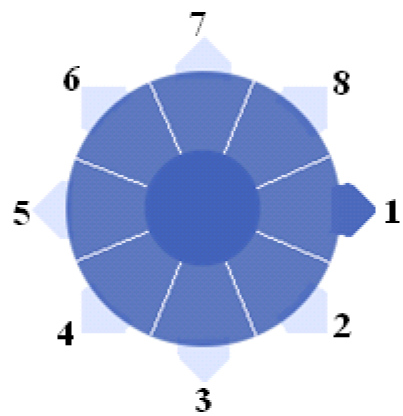

Figure 5. Numbered Positions for the arrow pointer in the TouchMouse.

Figure 5 shows the 8 available spaces numbered around TouchMouse's crown. The arrow pointer can be in any of these 8 places. Only one of them can contain the arrow pointer at the same time. The TouchMouse uses an automatic arrow pointer algorithm, which repeatedly updates the position of the arrow pointer in the TouchMouse's crown based on the movement of the TouchMouse on the screen.

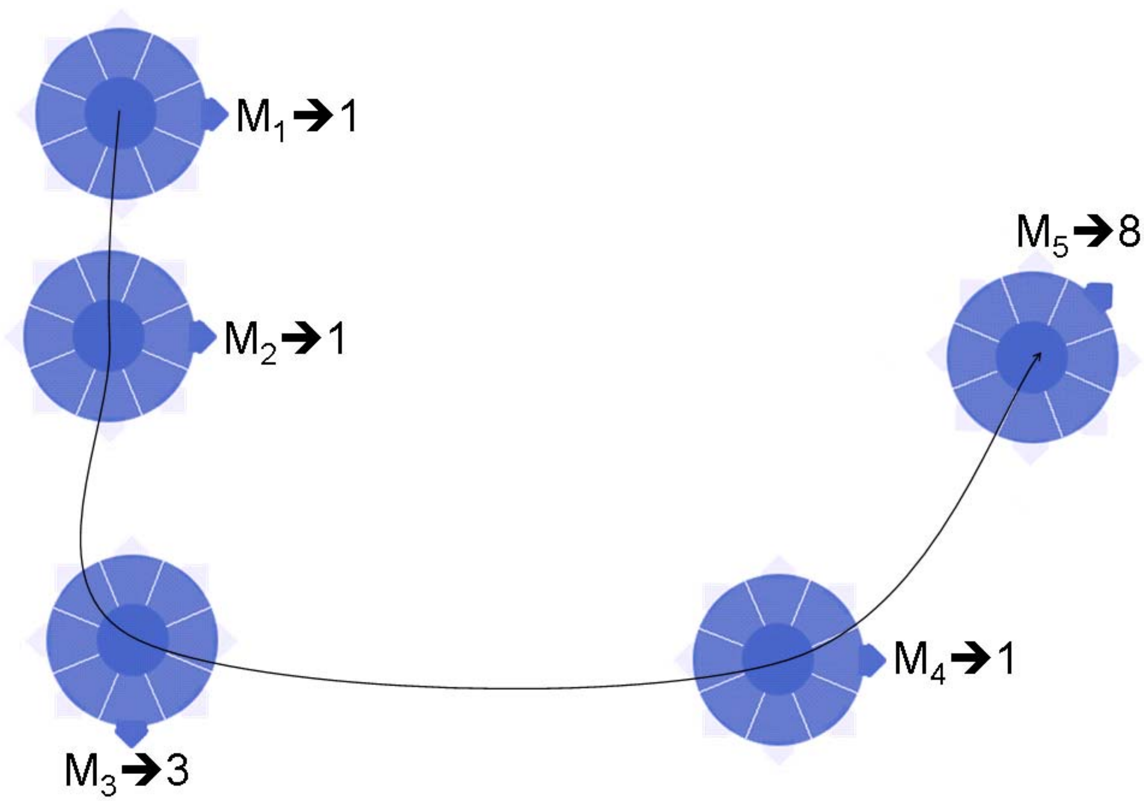

Figure 6. Automatic Arrow pointer.

Figure 6 shows the TouchMouse with the arrow pointer changing positions in the TouchMouse's crown. When the TouchMouse moves short distances the algorithm maintains the position of the arrow pointer. For instance, if the TouchMouse is moved from M1 to M2, the arrow pointer stays in position 1. This feature helps the user performs small adjustments without worrying about changes in the position of the arrow pointer.

This algorithm only changes the arrow pointer for longer distances. For example, if the TouchMouse is moved from M2 to M3, the arrow pointer changes from position 1 to position 3. If the TouchMouse is moved from M3 to M4, the arrow pointer changes from position 3 to position 1. If the TouchMouse is moved from M4 to M5, the arrow pointer changes from position 1 to position 8. There is more information about automatic arrow pointer algorithms in section 3.3.

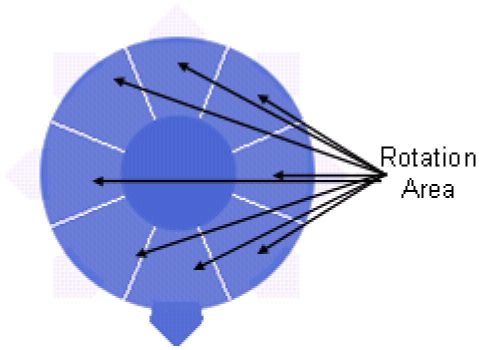

Figure 7. Rotation Area of the TouchMouse.

Figure 7 shows the rotation area. Although, the TouchMouse has an automatic arrow pointer algorithm that changes the position of the arrow pointer based on the TouchMouse movement, the user can use the rotation area to manually change the position of the arrow pointer to any space in the TouchMouse's crown. Thus, the user can adjust the position of the arrow pointer where he/she considers necessary.

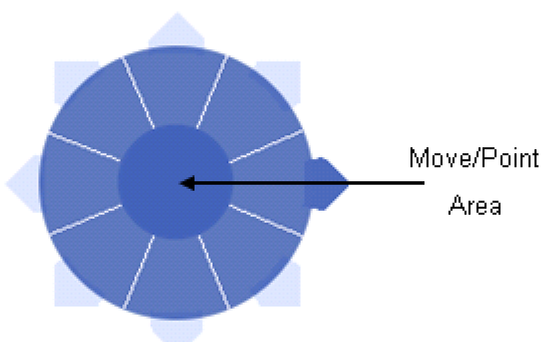

Figure 8. Move/Point Area of the TouchMouse.

Figure 8 shows the click and move area. This is the Move/Point area that the user can use to change the position of the mouse on the screen. When the user puts his/her finger in this area it is possible to change the TouchMouse position by moving the finger in the screen while keeping contact between the finger and the screen.

Thus, when the user lifts his/her finger from the screen the TouchMouse will stop moving. To use the click in the TouchMouse, the user just need to tap in the Move/Point area. Thus, to click the button you have to lift your finger from the TouchMouse then touch the Move/Point area in the TouchMouse and return to lift the finger.

Transparency of the TouchMouse

An important feature of the TouchMouse is the semi transparent color. It means, if a user is moving the TouchMouse over an application, he/she is able to see what is behind the mouse. Thus, it is possible to avoid obscuring an important part of the screen.

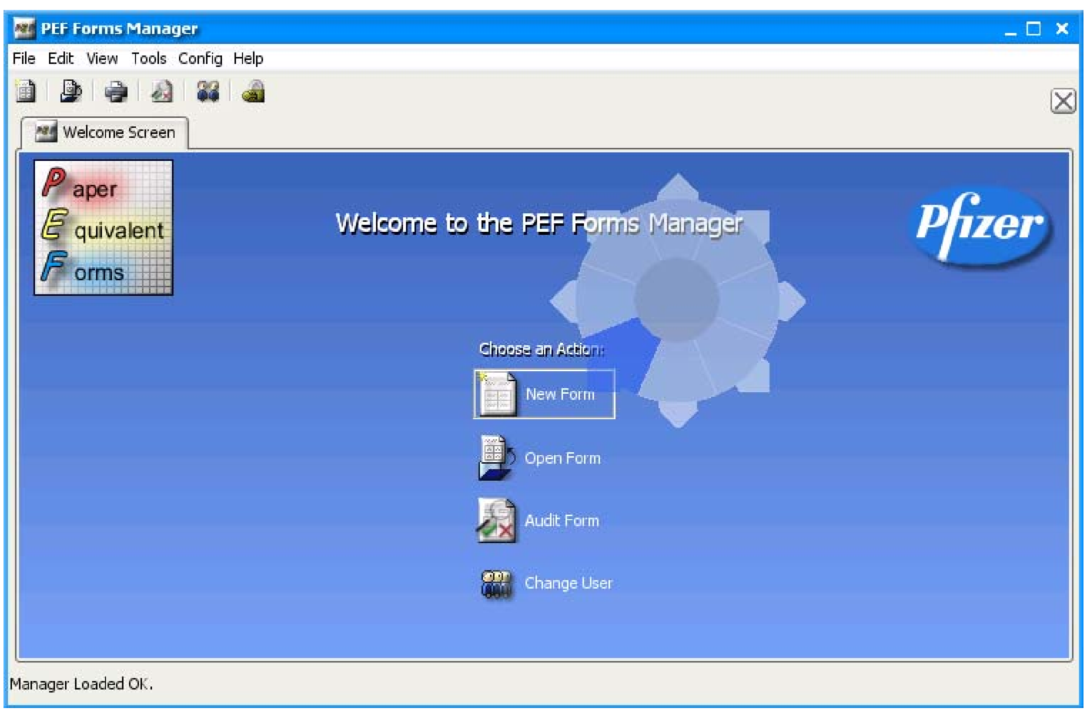

Figure 9. TouchMouse over the Welcome Screen in PEF [1].

Figure 9 presents the TouchMouse over PEF. There is a more detailed description on how to use the TouchMouse and PEF in section 3.1.2.3.

3.1.2.2 Advantages of the TouchMouse

The TouchMouse overcomes some of the limitations of touch screens:

- First, the user's finger can not obscure an important part of the screen because there is a small distance between the arrow pointer and the finger. Thus the user finger never obscures where he/she wants to click.

- Second, with the user finger it is difficult to point at targets that are smaller than the finger width. The human finger as a pointing device has very low "resolution" compared to a computer mouse. The TouchMouse and a computer mouse have the same resolution as pointing devices.

3.1.2.3 How to use PEF forms with the TouchMouse

In this section, we explain how to open an existing PEF form and how to create a new form by using the TouchMouse as the input device. If you want to open an existing form you must use the PEF Forms Manager. When the mouse is above the Open Form button the interface will change showing that it is active. You have to click on it for opening a form. To open a PEF form you have to use the associated information of the form like date, line and shift. This information is used to retrieve the created form. Let's assume that we want to open a form with the following parameters:

- Date: October 22, 2007.
- Line: 1.
- Shift: 2.

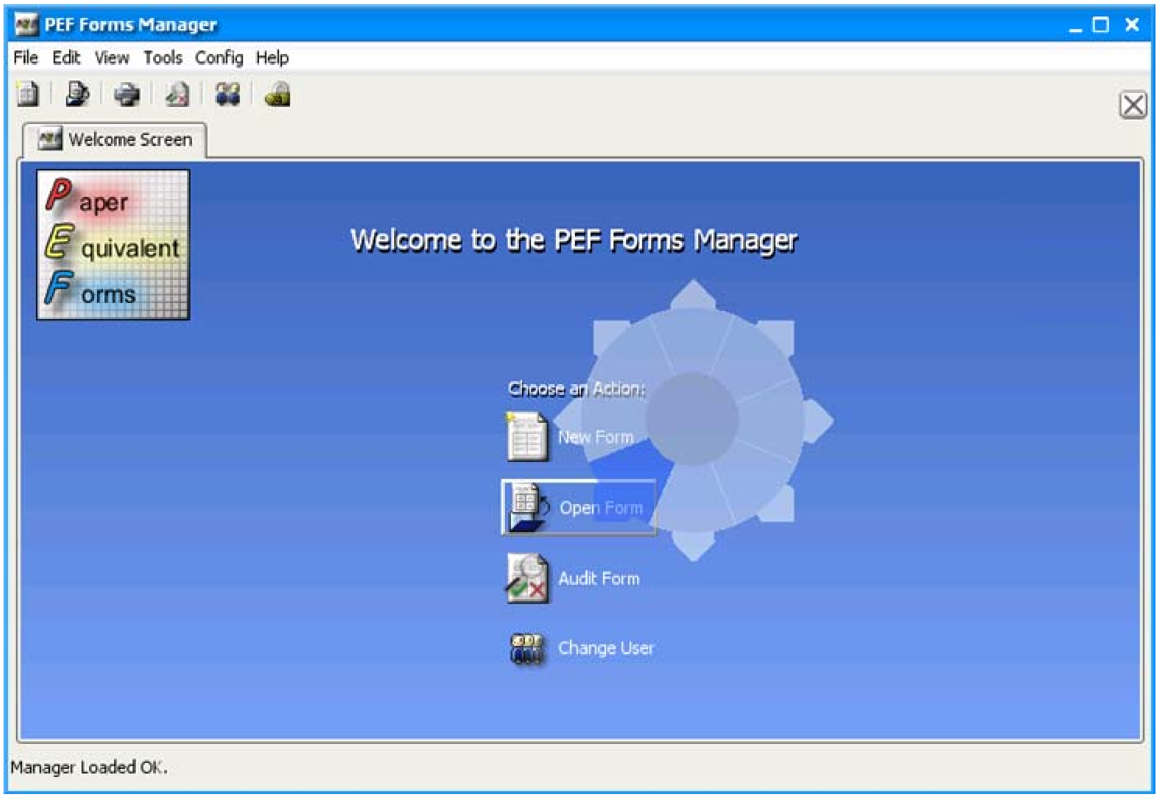

Figure 10. Open a Form in PEF [1] using the TouchMouse.

We need to click in the "Open Form" button. First we move the TouchMouse by placing the finger in the Move/Point area and dragging it to the open form button. The finger must keep touching the screen while moving the TouchMouse. Then we point the arrow of the TouchMouse in the direction of the button as shown in Figure 10. To click the button you have to lift your finger, and then touch the Move/Point area in the TouchMouse again. Thus, when the user taps the Move/Point area then the TouchMouse fires a click event.

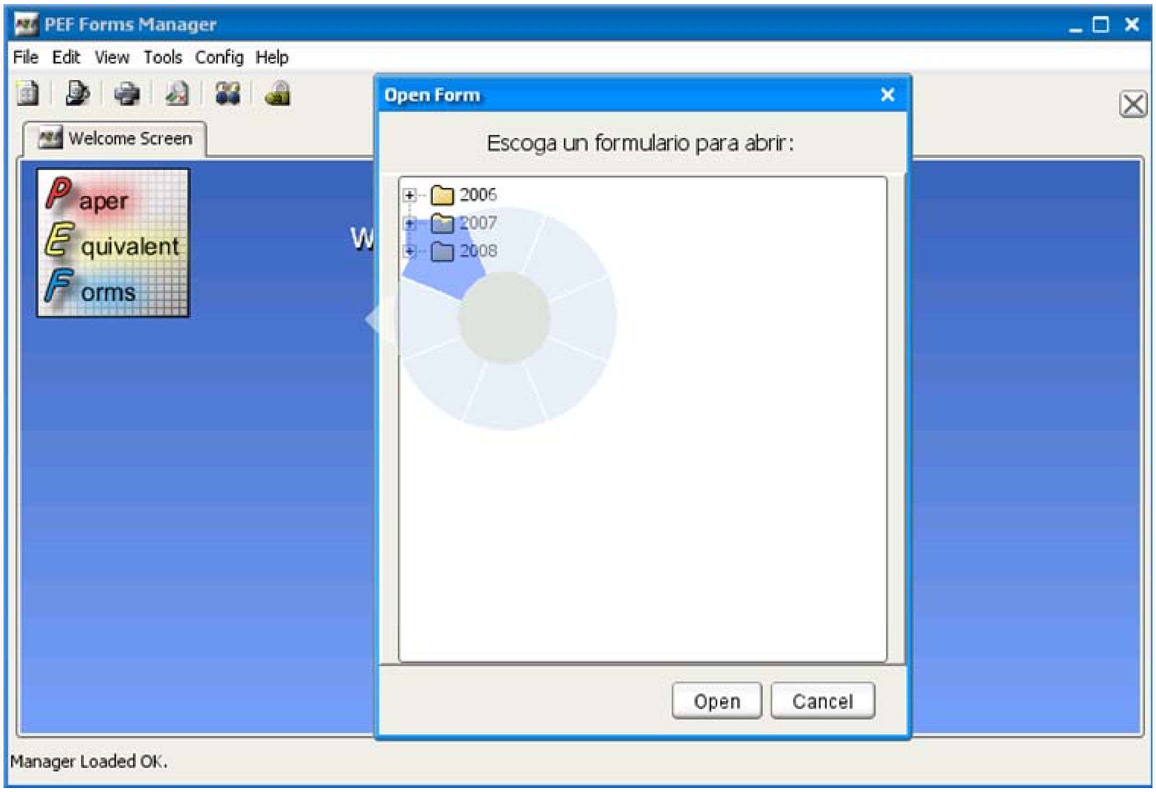

Figure 11. The TouchMouse choosing year 2007 in PEF [1].

PEF application used a tree distribution to organize the existing forms. Thus, when a user wants to use this interface to open a form, he/she finds the year in the first level, month in the second level, day in the third level, line in fourth level and shift in fifth level. Figure 11 shows the TouchMouse choosing the year 2007.

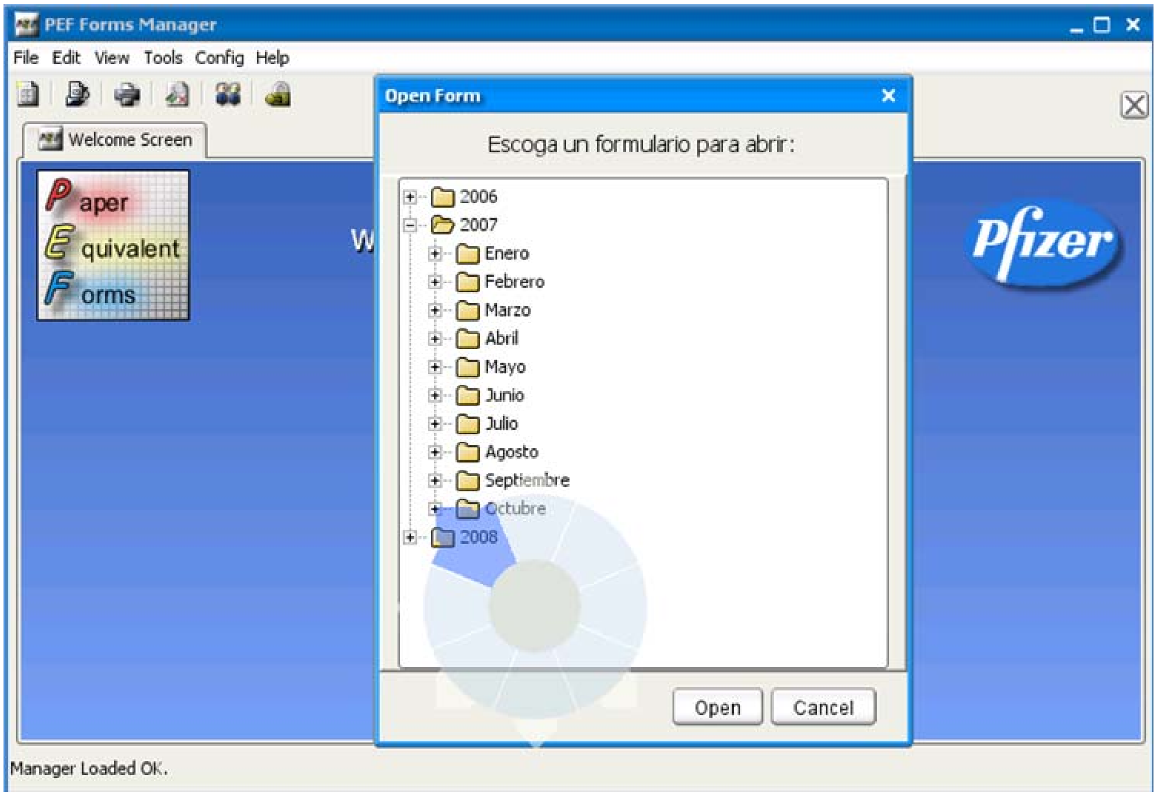

Figure 12. The TouchMouse choosing month October in PEF [1].

Once you click on any year then the interface will show all the months with existing forms. You need to move the TouchMouse near the month you want. In this case the arrow pointer should be over October. The Move/Point area of the TouchMouse must be used to choose this month. Figure 12 shows the TouchMouse choosing October.

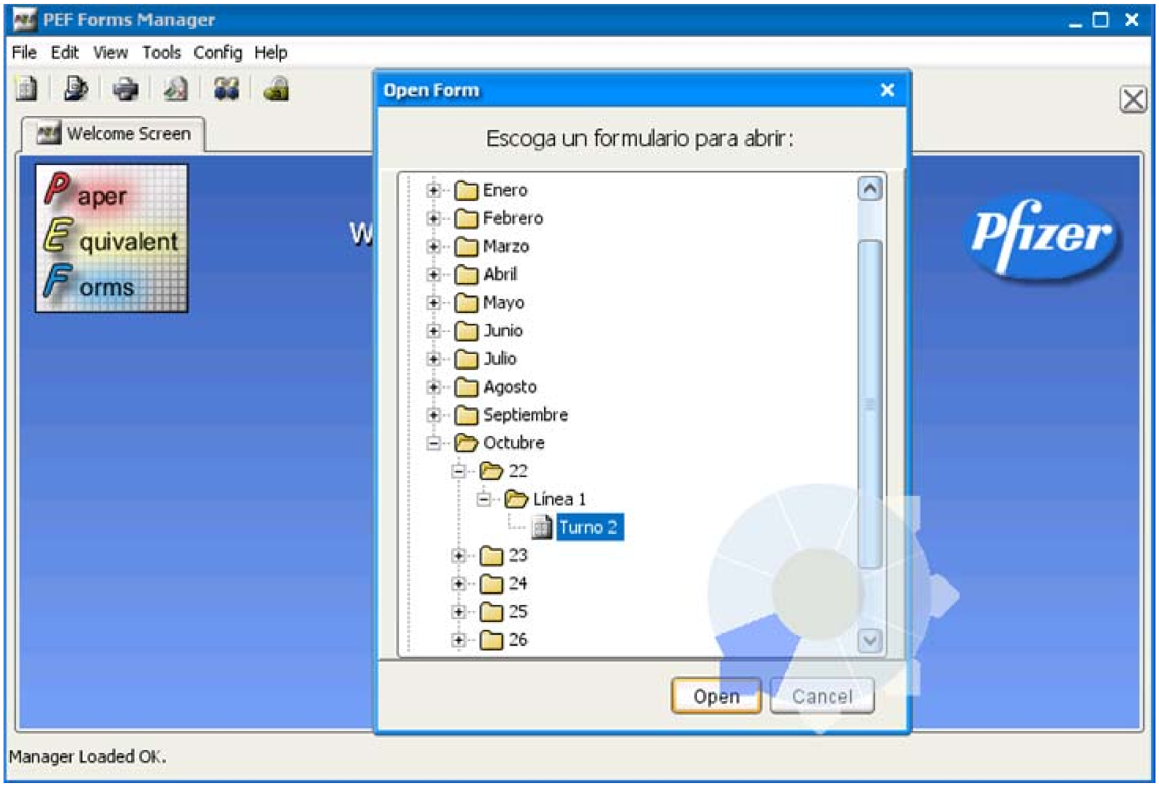

Figure 13. The TouchMouse choosing Line 1 and Shift 2 in PEF [1].

Once you click on any month the interface will show all the days with created forms. You need to move the TouchMouse near the day you want and click on it. Thus, the interface will show all the lines with created forms. After you choose the line then the interface will show all the shifts with created forms. Figure 13 presents the TouchMouse for choosing Line 1 and Shift 2.

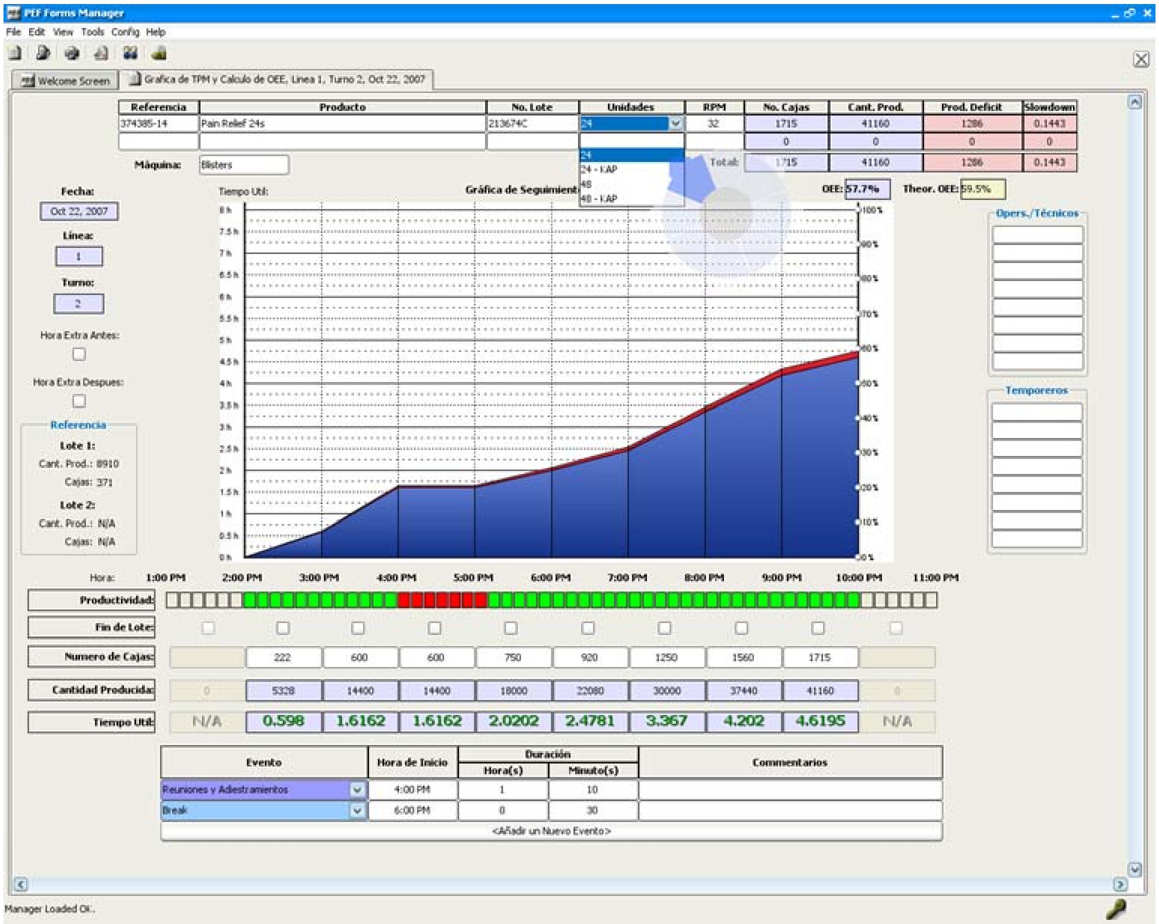

Figure 14. The TouchMouse in the opened form in PEF [1].

Figure 14 shows the form for October 22, 2007, Line 1 and Shift 2. We used the data from Appendix F to fill this form. The upper part of the form shows number of reference, name of product, lot number, number of units and name of the machine. Besides, to analyze performance based on the information in the form, you have the revolutions per minute (RPM) of the manufacturing machine. The graph in the center of the form shows the production achieved by the operators for every hour in the shift. On the right there are slots to put the name of the operators working within that shift and line.

Below the graph, there is the part that corresponds to the productivity data. It shows the hours for a shift. There is a group of 6 boxes to indicate the productivity; each one represents a time frame of ten minutes and has three colors red (no productivity, an event happened), green (productivity) and white (no information available). It is possible to have two lots in the same shift, so you can put a checkmark in the box corresponding to the end of a lot. The number of boxes packaged on each hour; when a number is inserted then the system will calculate the produced quantity and the useful time. At the bottom of the form is the information about different events that occurred during the shift and lasted more than 10 minutes. Each event has name, begin time, duration in hours and minutes and description. We can add as many events as we want only if we do not go beyond the maximum time for the shift. This is only one of many validations that PEF conducts internally to minimize enter of wrong or inaccurate data.

For every new form there is some information associated like date of the form, a line and a shift. This information is used to create a new form. Suppose that you want to use the TouchMouse for creating a new form with the next parameters:

- Date: October 29, 2007.
- Line: 1.
- Shift: 1.

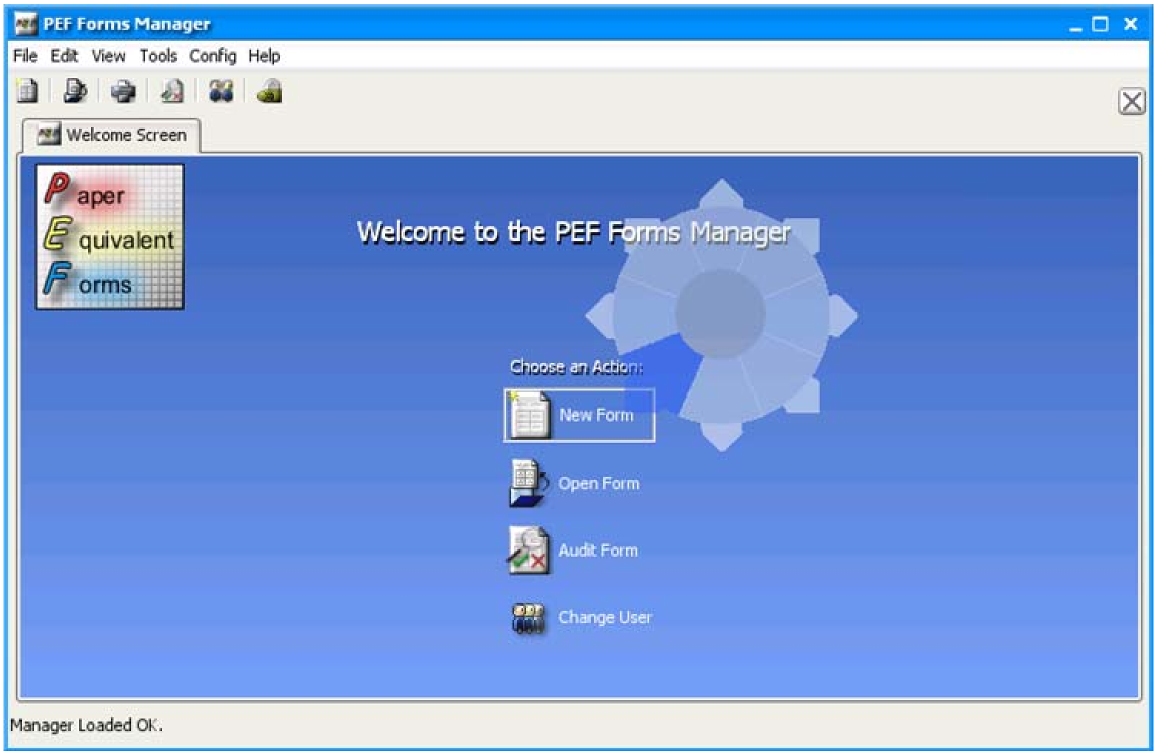

Figure 15. TouchMouse over the New Form button in PEF [1].

We need to click in the "New Form" button. First we move the TouchMouse by putting a finger in the Move/Point area and move the finger near the new form button. Then we point the arrow of the TouchMouse in the direction of the button. To change the arrow position the finger must be put in the rotation area near the place where the arrow point should be as shown in Figure 15. When the finger touches any part of the rotation area the arrow pointer will change its position to that place.

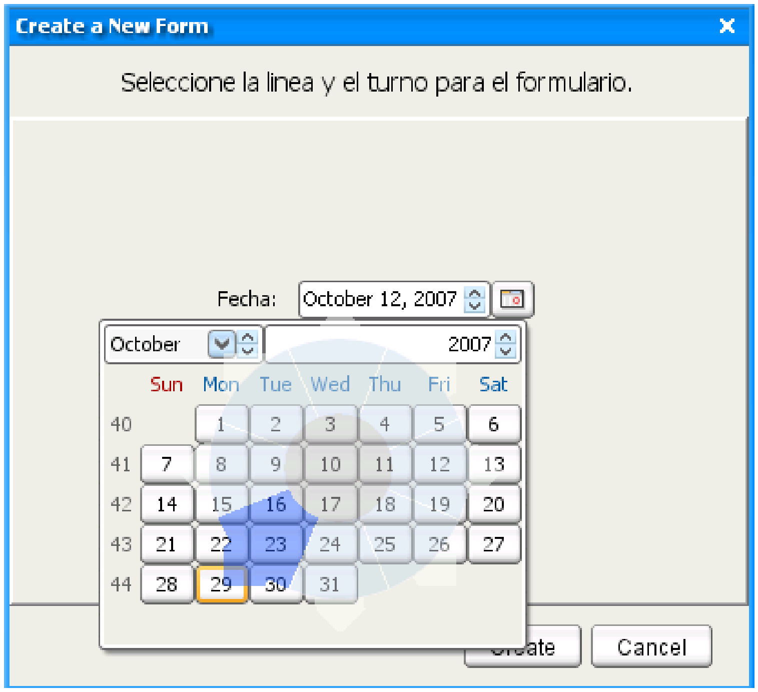

Figure 16. TouchMouse setting a date for a new form in PEF [1].

Figure 16 shows the TouchMouse selecting October 29, 2007 as date for the new form in the calendar. When you need to use the TouchMouse to interact with a button you use the Move/Point area to change the TouchMouse location until one of the arrows in the TouchMouse's crown is over the button. If the arrow pointer is in another position in the TouchMouse's crown you can use the Rotation area to change the position of the arrow pointer. If the arrow pointer is over the button you can use the Move/Point area to click the button. Remember, to click the button you have to lift your finger and then touch the Move/Point area in the TouchMouse and lift the finger again.

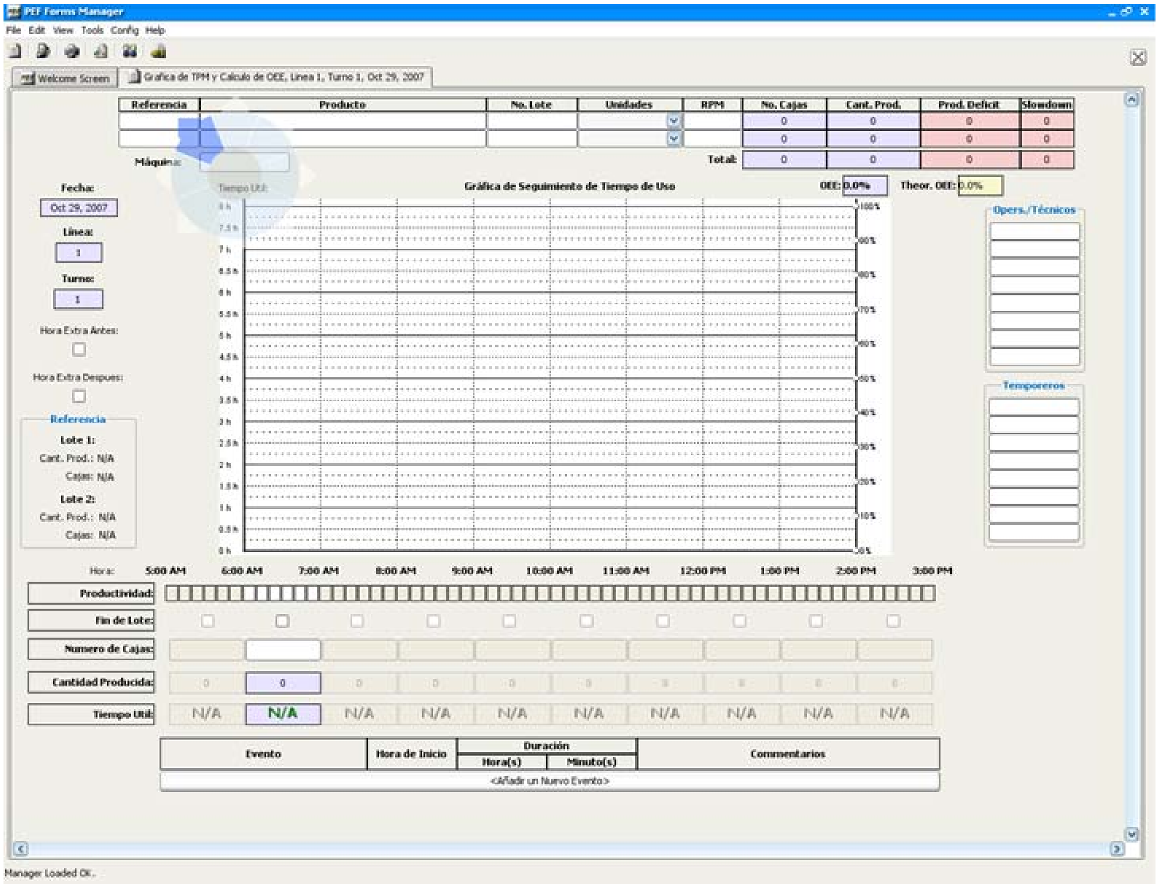

Figure 17. New Form in PEF [1] created using the TouchMouse.

Figure 17 presents the empty form created in PEF. This window shows an empty form with all the necessary information for a packaging process. The TouchMouse can be used to interact with any component of this interface even if the component is much smaller than the finger width. Thus, you can use the Move/Point area to change the TouchMouse location until one of the arrows in the TouchMouse's crown is over a component in the interface. The component can be clicked by lifting your finger and then tapping the Move/Point area in the TouchMouse.

3.2 Evolution of the TouchMouse

In this section, we present a brief description of the evolution of the TouchMouse and its algorithms.

3.2.1 First TouchMouse

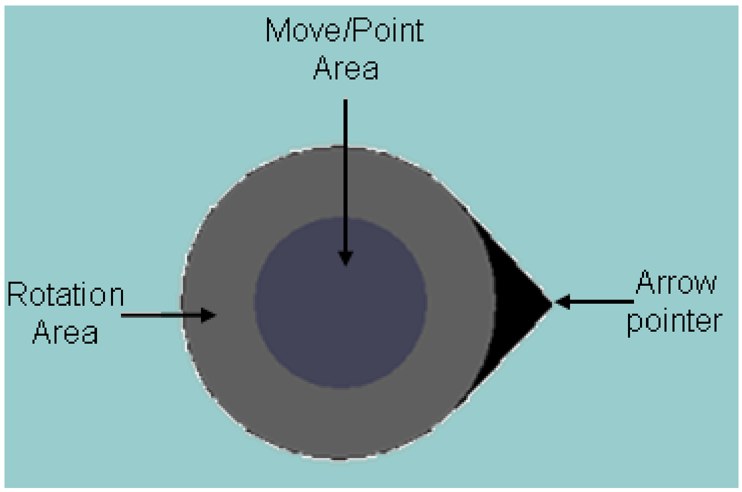

Figure 18. First design of the TouchMouse.

The first design of the TouchMouse (Figure 18) used a square window in Java. The TouchMouse was round shaped implemented using transparency. The transparency was simulated by taking a screen shot and using it as background and then drawing the TouchMouse on top of it. The rotation area was contiguous thus the arrow pointer can be placed in any orientation.

When the user change the screen with another window then it was necessary to take another screen shot; this make the response time of the TouchMouse very slow and the user can see the movement of the TouchMouse as discrete. It was difficult to calculate the new position of the arrow and draw it for clicking with the mouse.

3.2.2 Second TouchMouse

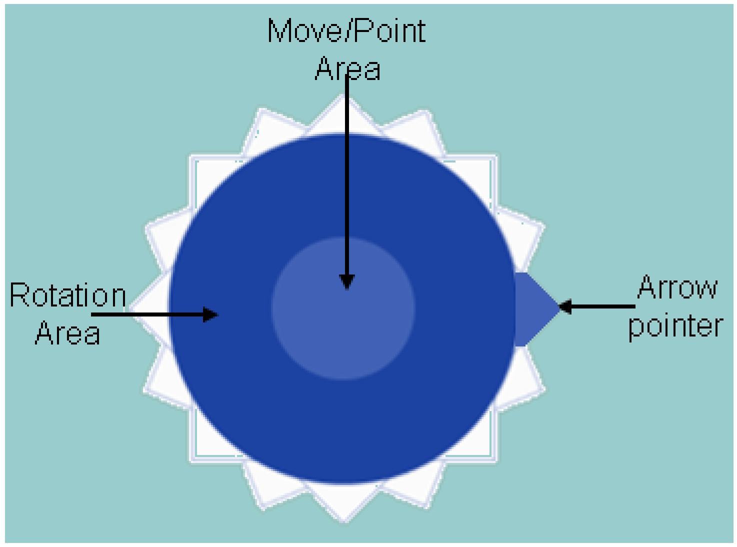

Figure 19. Second design of the TouchMouse.

Figure 19 shows the second generation of the TouchMouse. In this case, we defined a limited number of arrow pointer positions. We improved the simulated transparency by taking fewer screen shots to draw the TouchMouse. The TouchMouse with 16 arrow pointer positions makes easier to calculate the positions. The angle difference between two contiguous arrow pointers is 22.5 degrees.

Regardless of the improved transparency, the problems with the response time and the appearance of discrete movement persisted. The user could not see what was under the image of the TouchMouse.

3.2.3 Third TouchMouse

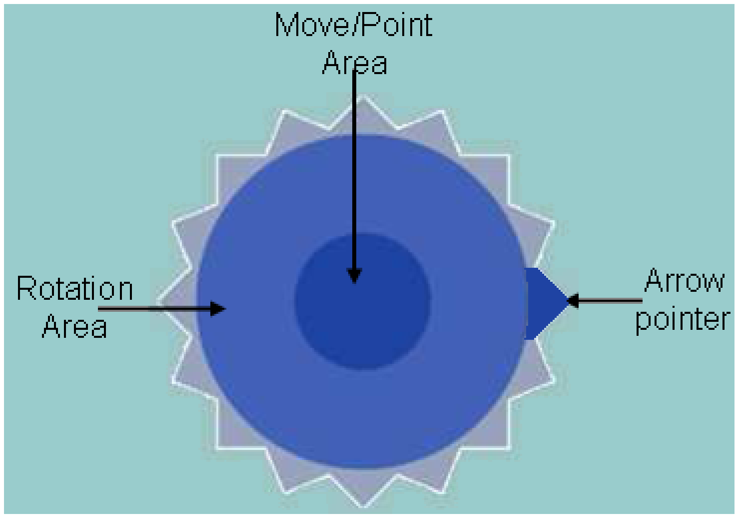

Figure 20. Third design of the TouchMouse.

Figure 20 shows the third design of the TouchMouse. Transparency in this version was made with Java Native Interface (JNI)[5]. JNI is a programming framework that allows Java code running in the Java virtual machine to call and be called by native applications and libraries written in other languages, such as C, C++ and assembly. It was possible to see behind the mouse with the improvement in the transparency; also, the response time was considerably improved.

The response time of the TouchMouse was still too slow and the movement was still perceived as discrete.

[5] Java Native Interface (JNI). December 2007. http://en.wikipedia.org/wiki/JNI

3.2.4 Fourth TouchMouse

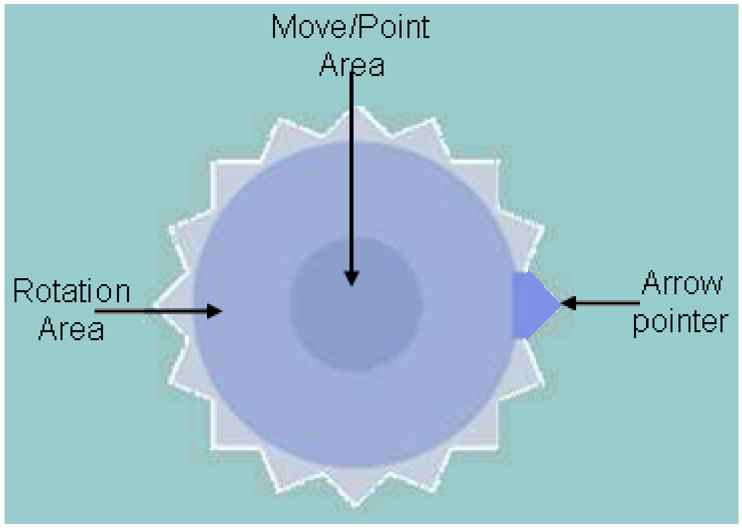

Figure 21. Fourth design of the TouchMouse.

Figure 21 shows the fourth design of the TouchMouse. For this version, we implemented a C++ program that defines a transparent window; this helped to speed up the transparency process. We used lighter colors in the TouchMouse; it was possible to see clearer behind the mouse and the response time was improved.

In many cases, it is necessary to change the location of the arrow pointer. We came out with the idea of implementing an automatic arrow pointing algorithm to help the user when the TouchMouse is been moved. It is a little difficult to select the area that corresponds to each arrow; it is necessary to try several times before we put the arrow pointer in the right position. It is hard for a user to distinguish what the touch area is for each arrow pointer resides.

3.2.5 Fifth TouchMouse

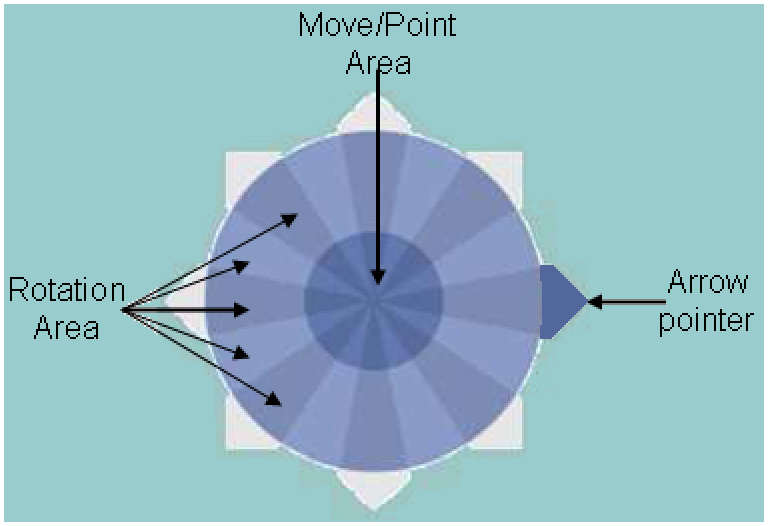

Figure 22. Fifth design of the TouchMouse.

Figure 22 shows the fifth generation of the TouchMouse. We put lines and alternated colors in the Rotation Area of the TouchMouse to distinguish the specific area to select the location of the arrow pointer. This helped the user to use his/her finger to change the arrow pointer more accurate. We tried some algorithms to change the arrow pointer based on the movement of the TouchMouse. The algorithms used in this section are:

- Simple Algorithm.
- Extreme Points Algorithm.
- Moving Average Algorithm.

This means that when the user is moving the TouchMouse in the screen there is an algorithm updating the position of the arrow pointer based on the direction of the movement. Thus, the user does not need to touch the rotation area for changing the position of the arrow pointer. The algorithms are explained in section 3.3.

We found out that it is more difficult to select the arrow in the blue areas than gray areas. This behavior could be explained because we are taking every change in the finger position and because every pixel has only 8 contiguous pixels. Thus, the algorithm to calculate the direction for the arrow from contiguous points causes this problem.

3.2.6 Sixth TouchMouse

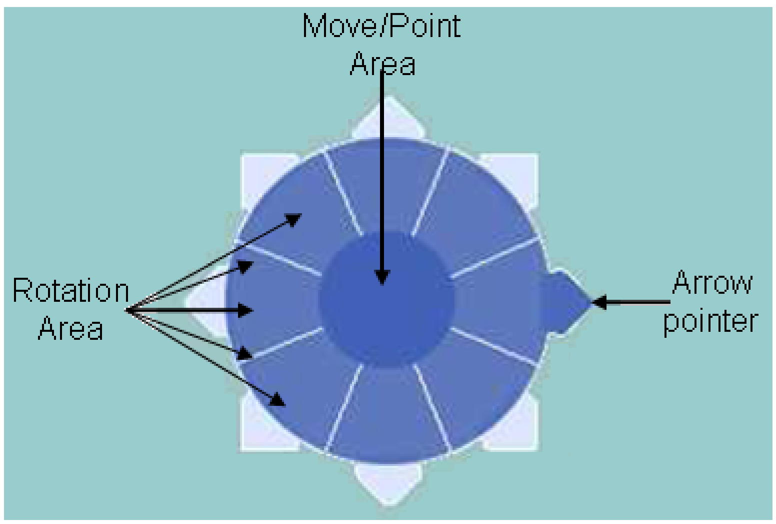

Figure 23. Sixth design of the TouchMouse.

Figure 23 shows the final design of the TouchMouse. In this version, we reduced the number of arrows from 16 to 8; this is because every pixel has exactly 8 contiguous neighbors' pixels and it is easier to calculate and locate the arrow in an accurate way. We clearly marked the area to change the direction of the arrow pointer using dividing lines.

We developed the Score algorithm to select the arrow pointer in the most accurate position. This was based on the movement of the TouchMouse when the user was dragging it. We developed another algorithm based on linear regression to make the automatic arrow; we put a bound of 135 points to calculate the new pointer position. This limit is because we found that the automatic arrow is useful with long distance but makes more difficult to position the mouse when there is a short distance. The algorithms are explained in section 3.3.

These algorithms helped to get a better performance of the TouchMouse. Thus, the arrow is fixed for shorter movements and automatic for longer movements. At this point we still did not like the fact that when a user moves the TouchMouse very fast on the screen, it looked like the TouchMouse was following the finger but not exactly behind.

3.3 Implementation of the TouchMouse.

When we were testing the TouchMouse we discovered a relation between the TouchMouse movement and the location of the arrow pointer. Thus, we designed some algorithms to improve the performance of the TouchMouse. These algorithms were intended to help users to use fewer actions in the TouchMouse by changing automatically the direction of the arrow pointer. This means that when the user is moving the TouchMouse in the screen there is an algorithm updating the position of the arrow pointer based on that movement. Thus, in many cases the user does not need to touch the rotation area for changing the position of the arrow pointer.

3.3.1 Simple Algorithm

We wanted the arrow pointer to change with every movement of the TouchMouse. Thus, when the user touched the Move/Point Area in the TouchMouse we captured the finger position in the screen and we used it as reference point.

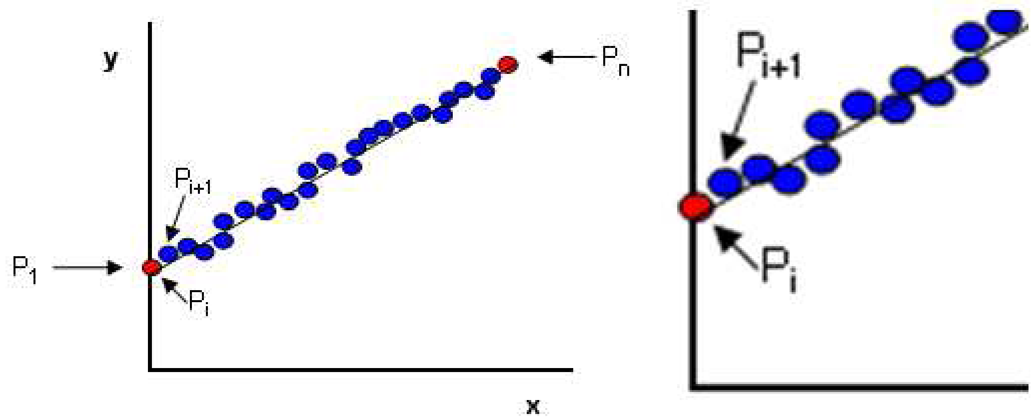

Figure 24. Contiguous points.

Figure 24 shows points representing coordinates of the TouchMouse while moving from P_1 to P_n and two contiguous points P_i and P_{i+1}. Whenever we captured a new finger position different from the reference point we used these two points to calculate the angle of the movement. We used the angle to match the closest location of the arrow pointer and then we updated the arrow pointer. We used the last point as reference point and repeated the process when a new position was detected.

```
Capture the initial finger position.
Translate position into screen coordinates.
If current finger position is different from the initial position
Then
        Calculate direction of the arrow.
        Make initial position = current position
Else
        Continue reading finger position.
```

Figure 25. Simple Algorithm.

Figure 25 presents a pseudo-code of the Simple Algorithm. This algorithm was our first approach for the automatic arrow pointer.

For instance, if you moved the TouchMouse and you captured these two coordinates (75, 50) and (200,300) then to calculate the position of the arrow pointer you must find the angle. There are some steps you must follow:

First, calculate the difference between the final position and initial position.

Final position = $(X_f, Y_f) = (200{,}300)$

Initial position = $(X_i, Y_i) = (75{,}50)$

Difference is:

$(X, Y) = (X_f - X_i, Y_f - Y_i) = (200 - 75{,}300 - 50) = (125{,}250)$

Second, calculate the hypotenuse by using the Pythagorean Theorem showed in Equation 1:

$$h^2 = x^2 + y^2 \qquad (1)$$

$h^2 = x^2 + y^2 = 125^2 + 250^2$

$h = \sqrt{(125^2 + 250^2)} = 279.508$

Third, use the inverse trigonometric functions to calculate the angles. Equation 2 presents the inverse equation of the sine function and Equation 3 presents the inverse equation of the cosine function.

$$AlphaSin = \arcsin\left(y/h \right) \tag{2}$$

$$AlphaCos = \arccos\left(x/h \right) \tag{3}$$

Where, $AlphaSin$ is the angle from the inverse function of the Sine function and $AlphaCos$ is the angle from the inverse function of the Cosine function.

$$AlphaSin = \arcsin\left(y/h \right) = \arcsin\left(250/279.508 \right) = 1.107 radians$$

$$AlphaCos = \arccos\left(x/h \right) = \arccos\left(125/279.508 \right) = 1.107 radians$$

Fourth, Equation 4 presents how to change from radians to degrees and adjust the angle when it is necessary.

$$Alpha = 180 * \frac{AlphaCos}{PI} \tag{4}$$

Where, Alpha is the angle in degrees and Pi is the mathematical constant equal to a circle's circumference divided by its diameter. This example has no need to adjust the angle because AlphaCos and AlphaSin are the same.

$$Alpha = 180 * \frac{AlphaCos}{PI}$$

$$Alpha = 180 * \frac{1.107}{PI} = 63.4349$$

Arrow pointer position	Initial Angle	Final Angle
1	-22.5	22.5
2	22.5	67.5
3	67.5	112.5
4	112.5	157.5
5	157.5	202.5
6	202.5	247.5
7	247.5	292.5
8	292.5	337.5

Table 1. Angles for every position of the arrow pointer.

Table 1 presents the range in degrees for every position of the arrow pointer. All the positions wrap around 360 degrees.

Fifth, we use Table 1 to match the Alpha angle (63.4349) with a position of the arrow pointer. Thus, the position of the arrow pointer is 2.

Sixth, update the position of the arrow pointer.

As another example, if you moved the TouchMouse and you captured these two coordinates (200,300) to (75, 50) then to calculate the position of the arrow pointer you must find the angle. First, calculate the difference between the final position and initial position.

Final position = $(X_f, Y_f) = (75{,}50)$

Initial position = $(X_i, Y_i) = (200{,}300)$

Difference is: $(X, Y) = (X_f - X_i, Y_f - Y_i) = (75 - 200{,}50 - 300) = (-125{,}-250)$

Second, calculate the hypotenuse by using the Pythagorean Theorem showed in Equation 1:

$$h^2 = x^2 + y^2 = (-125)^2 + (-250)^2$$

$h = \sqrt{\left((-125)^2 + (-250)^2\right)} = 279.508$

Third, use the inverse functions of the trigonometric functions presented in Equation 2 and Equation 3 to calculate the angles.

$$AlphaSin = \arcsin\left(y/h\right) = \arcsin\left(-250/279.508\right) = -1.107 radians$$

$$AlphaCos = \arccos\left(x/h\right) = \arccos\left(-125/279.508\right) = 2.034 radians$$

Where, AlphaSin is the angle from the inverse function of the Sine function and AlphaCos is the angle from the inverse function of the Cosine function.

Fourth, use Equation 4 to change from radians to degrees.

$$Alpha = 180 * \frac{AlphaCos}{PI} = 180 * \frac{2.034}{PI} = 116.565$$

Where Alpha is the angle in degrees. If $AlphaCos * AlphaSin < 0$ then the angle is out of phase. This example needs to use Equation 5 to adjust the angle.

$$Alpha = 360 - Alpha \quad (5)$$

Thus, $Alpha = 360 - 116.565 = 243.435$

Fifth, use Table 1 to match the Alpha angle (243.435) with a position of the arrow pointer. As a result, the position of the arrow pointer is 7.

Sixth, update the position of the arrow pointer.

3.3.2 Extreme Points Algorithm.

The use of two contiguous points caused a lot of variability in the arrow pointer. Therefore we decided to capture a specific number of finger positions in the screen and put them in a queue.

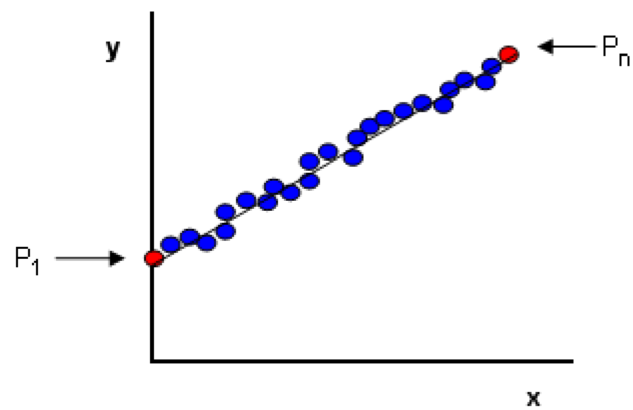

Figure 26. Extreme Points.

Figure 26 shows points representing coordinates of the TouchMouse while moving from P_1 to P_n. We used only the first and last point in the queue to calculate the angle of the movement. Then, we used the angle to match a specific arrow position and then we updated the arrow pointer.

```
Define an n, where n is the size of a queue.
Create a counter = 0.
Capture the initial finger position.
Translate position into screen coordinates.
Add the position to the queue.
While dragging
        Increase counter.
        If counter > n
        Then
                Take initial and last position in the queue.
                Calculate direction of the arrow.
                Update arrow.
                Clear the queue.
        Else
                Continue capturing finger position.
                Translate position into screen coordinates.
                Add the position to the queue.
End-while.
Clear the queue.
```

Figure 27. Extreme Points Algorithm.

Figure 27 presents the pseudo-code for the Extreme Points Algorithm. This algorithm was designed to avoid the problem with contiguous points.

3.3.3 Moving Average algorithm.

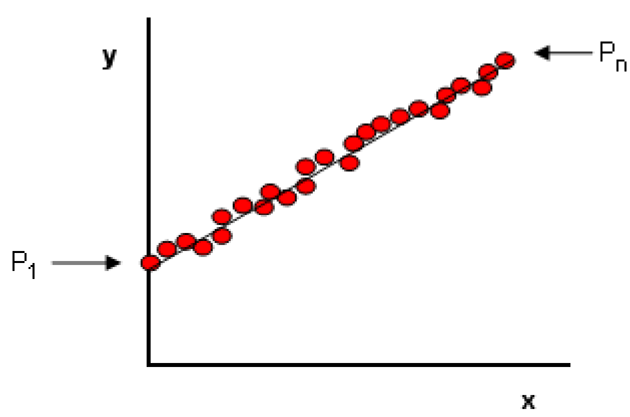

Figure 28. All points.

We discovered that using information with only two points often resulted in a wrong prediction of the final arrow direction intended by the user. Then we decided to use all the n points captured in the movement of the TouchMouse to have a more accurate automatic arrow pointer as shown in Figure 28.

We used the definition of moving average[6] used in statistics to develop another algorithm. Given a sequence $(a_i)_{i=1}^{n}$ an n-moving average is a new sequence $(s_i)_{i=1}^{N-n+1}$ defined from the a_i by taking the average of the subsequences n terms. Equation 6 defines the sequence of coordinates. Where, a_i is a coordinate that represents the finger position on the screen, when the user is dragging the TouchMouse.

$$s_i = \frac{1}{n} * \sum_{j=i}^{i+n-1} a_j \tag{6}$$

Equation 7 presents the sequence for the 2-moving average.

$$s_2 = \frac{1}{2} * (a_1 + a_2, a_2 + a_3, ..., a_{n-1} + a_n) \tag{7}$$

Equation 8 presents the sequence for the 3-moving average.

$$s_3 = \frac{1}{3} * (a_1 + a_2 + a_3, a_2 + a_3 + a_4, ..., a_{n-2} + a_{n-1} + a_n) \tag{8}$$

---

[6] Moving average. December 2007. http://en.wikipedia.org/wiki/Moving_average

```
Define an n, where n is the limit for the new sequence.
Define an x, where x is the size of a queue.
Create a counter = 0.
Capture the initial finger position.
Translate position into screen coordinates.
Add the position to the queue.
While dragging
        Increase counter.
        If counter > n
        Then
                Create the n-sequence.
                Calculate direction of the arrow.
                Update arrow.
                Clear the queue.
        Else
                Continue capturing finger position.
        Translate position into screen coordinates.
        Add the position to the queue.
End-while.
Clear the queue.
```

Figure 29. Moving Average Algorithm.

We used a sequence S_n to select the location of the arrow pointer based on the average of the angles calculated from the coordinates a_i and a_{i+1} of S_n. We made some tests to find an appropriate n for this algorithm. The best result for this algorithm was with n = 100. But there was no significant improvement in the accuracy in the automatic arrow pointer compared with the other algorithms. Figure 29 presents a pseudo-code of the Moving Average Algorithm.

We used the Extreme Points algorithm to avoid the problem with contiguous points and there was an improvement when we were moving the

TouchMouse. Although the arrow pointer was better with this algorithm, the arrow pointer was not totally accurate.

3.3.4 Score Algorithm.

Since, there was no significant improvement in the accuracy of the arrow pointer with the Moving Average algorithm we decided to use the most apparent direction. When we used two contiguous points there was too much variability in the arrow pointer. But if we used the angle from two contiguous points and match a specific position of the arrow pointer then it was possible to have a counter for every position of the arrow pointer.

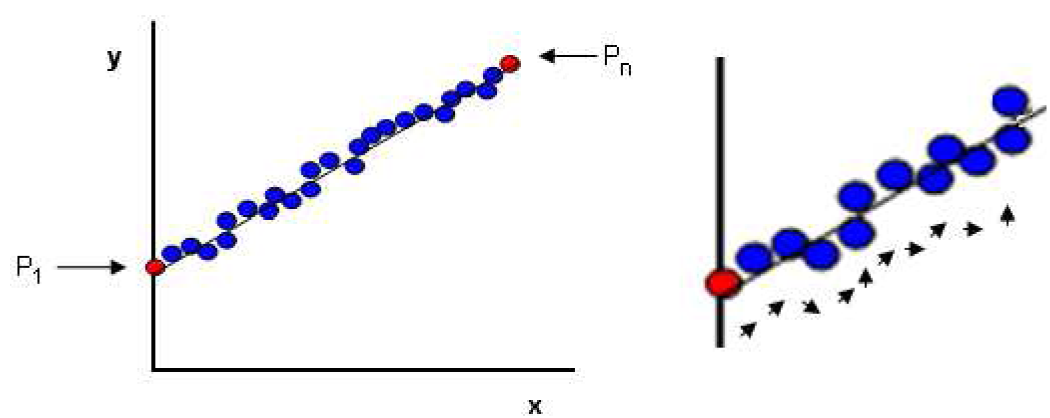

Figure 30. Direction to score.

Figure 30 shows points representing coordinates of the TouchMouse while moving from P_1 to P_n and every direction while moving from one point to another.

```
Define an n, where n is the goal.

Create an array to keep a counter for each possible position of the arrow.

Reset the array of counters = 0.

Capture the initial finger position.

Translate position into screen coordinates.

While dragging

        Capture the finger position.

        Translate position into screen coordinates.

        Calculate the angle of the movement based on the last two values.

        Increase arrow counter that corresponds to the angle.

        If any counter > n

        Then

                Calculate direction of the arrow based on the angle.

                Change the position of the arrow pointer.

                Clear the array of counters.
End-while.

Clear the array of counters.
```

Figure 31. Score Algorithm.

Figure 31 presents the pseudo-code of the Score Algorithm. The Score algorithm consisted of assigning the arrow pointer to the position that reaches the largest counter or “Score”. This score indicates that it was probably because the user was trying to go in the direction that corresponds to that position. The best result for this algorithm was with $Score = n = 80$. This means that the location of the arrow pointer will be change as soon as the first counter reaches $n = 80$

This algorithm was designed to avoid the problem with the variability of the TouchMouse movement and to take advantage of the information from many points along the movement path.

3.3.5 Linear Regression Algorithm.

Each movement path could be seen as a trend line. The trend line represents the general direction in which the user wants to move the TouchMouse. Thus, a trend line could simply be drawn through a set of position points, their position and slope can be calculated using statistical techniques such as linear regression.

We used the definition of linear regression[7] used in statistics to develop this algorithm. We have a set of coordinates (x_i, y_i). Those coordinates represent finger positions when the user is dragging the TouchMouse.

---

[7] Linear regression. December 2007. http://en.wikipedia.org/wiki/Linear_regression

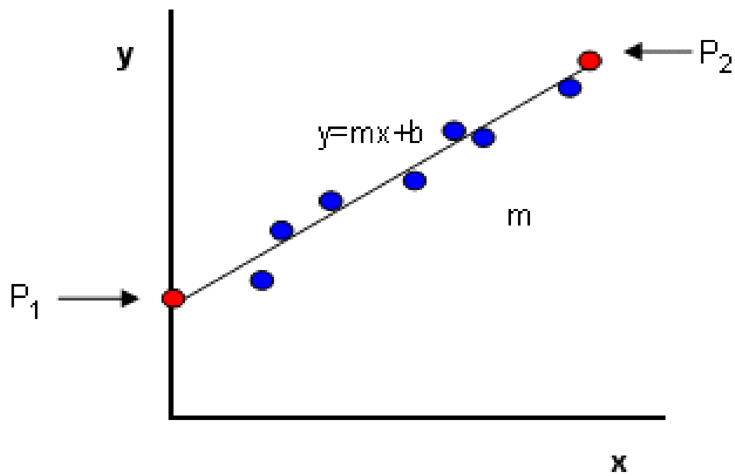

Figure 32. Linear Regression Model.

If the user is trying to move the TouchMouse from position p_1 to position p_2 (Figure 32) then we have reason to believe that there exists a linear relationship between the variables x and y. So, we can plot the data and draw a "best-fit" straight line through the points. Of course, this relationship is governed by the familiar equation $y = mx + b$. We can then find the slope, m, and y-intercept, b, for the data.

It is also possible to determine the correlation coefficient, r, which gives us a measure of the reliability of the linear relationship between the x and y values. A value of r = 1 indicates an exact linear relationship between x and y. Values of r close to 1 indicate excellent linear correlation. If the correlation coefficient is relatively distant from 1, the predictions based on the linear relationship are less reliable. Given a set of data (x_i, y_i) with n screen points, the slope, y-intercept and correlation coefficient, r, can be determined using the following equations:

$$m = \frac{n * \sum(x * y) - \sum x * \sum y}{n * \sum(x^2) - \sum(x)^2} \quad (1)$$

$$b = \frac{\sum y - m * \sum x}{n} \quad (2)$$

$$r = \frac{n * \sum(x * y) - \sum x * \sum y}{\sqrt{\left(\left(n * \sum(x^2) - \sum(x)^2\right) * \left(n * \sum(y^2) - \sum(y)^2\right)\right)}} \quad (3)$$

We apply these formulas to the set of data points; the point (0, 0) is located at the top-left corner of the screen. Linear regression gives the slope m and a function $y = mx + b$ that describes the movement.

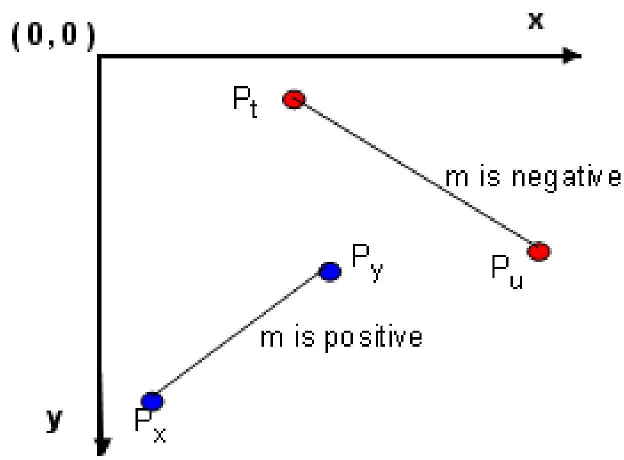

Figure 33. Linear Regression in a screen.

As shown in Figure 33, if you are moving from point p_t to p_u you have a straight line with a negative slope m but if you are moving from point p_u to p_t you have the same straight line with the same negative slope m. Similarly, if you are moving from point p_x to p_y you have a straight line with a positive slope m but if you are moving from point p_y to p_x you have the same straight line with the same positive slope m .

```
Define n, where n is the size of the queue.
Create a counter = 0.
Capture the initial finger position.
Translate position into screen coordinates.
Add the position to the queue.
While dragging
        Increase counter.
        If counter > n
        Then
                Take initial and last position in the queue.
                Calculate direction of the movement.
                Calculate m, b and r to create the function y=mx+b.
                Use r to measure the reliability of the function.
                if r > 0.95
                        Calculate the angle from the function.
                        Update the arrow pointing.
                        Clear the queue.
                Else
                        Delete the first element in the queue.
                        Continue capturing finger position.
                        Translate position into screen coordinates.
                        Add the position to the queue.
        Else
                Continue capturing finger position.
                Translate position into screen coordinates.
                Add the position to the queue.
End-while.
```

Figure 34. Linear Regression algorithm.

Figure 34 presents the pseudo-code for the linear regression algorithm. Linear regression can not give the direction of the movement. Thus, we used the extreme points' idea explained in section 3.3.2 to get an approximated direction and we combined this information with the linear regression approach. If the direction from the extreme points indicates that direction is down to top then we add 180 degrees to the calculated angle from linear regression to get the real direction. We chose an r value of 0.95 to have a 95% of confidence in the linear regression results.

4 USER TESTING AND RESULTS

The purpose of this chapter is to describe and present the results of the usability test on the TouchMouse with a Paper Equivalent Form application (PEF) [1].

4.1 Introduction

One of the most fundamental usability measuring methods is User Testing [17][19]. This method is an important part of the software development cycle that addresses user interface design. More often, it provides direct information about potential problems related to the user interaction with the interface.

4.2 Methodology

4.2.1 Objective

The main objective of the usability test presented here was to identify potential usability problems, interaction differences, advantages, and disadvantages of the TouchMouse. One of the most important issues is to calculate the relative efficacy of the TouchMouse versus the Mouse versus paper form. Also, evaluate the learning speed of how to use the TouchMouse.

A pilot test was performed before the usability testing to find if the instructions for the test task were comprehensible and correctly interpreted by the subjects. It was also useful to find if there were deficiencies in the test plan. As a framework for the interface we will use the same PEF application presented in previous chapters.

4.2.2 Computing Infrastructure

The study presented in the next sections was conducted using a paper-based form and two computer based systems to interact with PEF [1]. The computer based systems were:

- A Dell Precision PWS 380, Intel Pentium D 2.8 GHz, 2 GB of RAM, Nvidia Quadro 540 with 128MB and Windows XP operating system. This system is for the Central Data Server (CDS). The CDS provided centralized storage and data replication as well as coordinate the different requests by the clients [1].
- A Dell Precision PWS 380, Intel Pentium D 2.8 GHz, 2 GB of RAM, Nvidia Quadro 540 with 128MB and Windows XP operating system. This system is for the Certification Authority (CA). The CA enabled the functionality of auditing the data on the Application Database by producing a certificate for every record, using the necessary cryptographic algorithms [1].

4.2.2.1 Mouse System

The components of the Form Manager System (Mouse version) are a Personal Computer (PC), an optical mouse and a LCD monitor.

- PC: Dell XPS M1210, 2.16GHz Intel Core 2 Duo-T7200 2GHz; 2GB DDR2 SDRAM PC5300 667MHz; Nvidia GeForce Go 7400 256MB; with Windows XP Home operating system.

- The LCD monitor was a Dell E196FP with an active matrix - TFT LCD, 19 inches, pixel pitch 0.294 mm, faceplate coating Antiglare with hard-coating 3H, response Time 8ms typical, resolution 1280 x 1024 at 75 Hz.
- The pointing device used for this version was an optical mouse: Dell Optical USB 2-Button Wheel Mouse by Logitech.

4.2.2.2 TouchMouse System

The components of the Form Manager System (TouchMouse version) are a Personal Computer (PC), a touch screen and the TouchMouse.

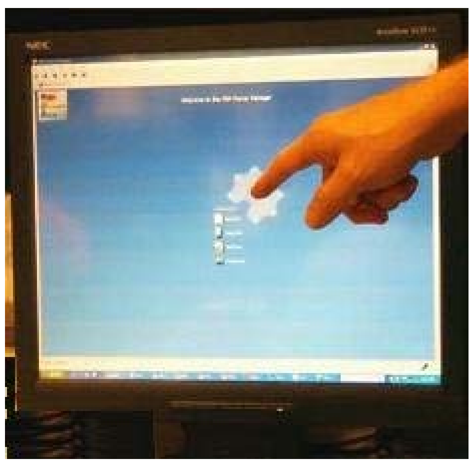

Figure 35. Touch Screen.

- PC: Dell XPS M1210, 2.16GHz Intel Core 2 Duo-T7200 2GHz; 2GB DDR2 SDRAM PC5300 667MHz; Nvidia GeForce Go 7400 256MB; with Windows XP Home operating system.

- The LCD monitor used for this version was a touch screen NEC[8] AccuSync LCD 72V[9]; as pointing device we used the TouchMouse. This LCD monitor is a 17" flat-panel with integrated capacitive touch screen technology by 3M[10]. Figure 35 shows the touch screen.

This screen has anti-glare properties, scratch resistant top coat provides smooth, easy glide touch surface with a good reliability for:

- Surface obstructions
- Chemical resistance
- Liquid resistance
- Suitable for extreme environmental conditions and outdoor applications
- Unaffected by most surface damages
- Shock and vibration resistance
- Input flexibility – operates with gloves, fingers, pens.

---

[8] NEC. December 2007. http://www.nec.com/

[9] AccuSync LCD 72V. December 2007. http://www.necdisplay.com/Products/Series/?series=fe775501-5090-4163-8cda-b3a638aba52d

[10] 3M. December 2007. http://www.3m.com/

4.2.2.3 Paper Form System

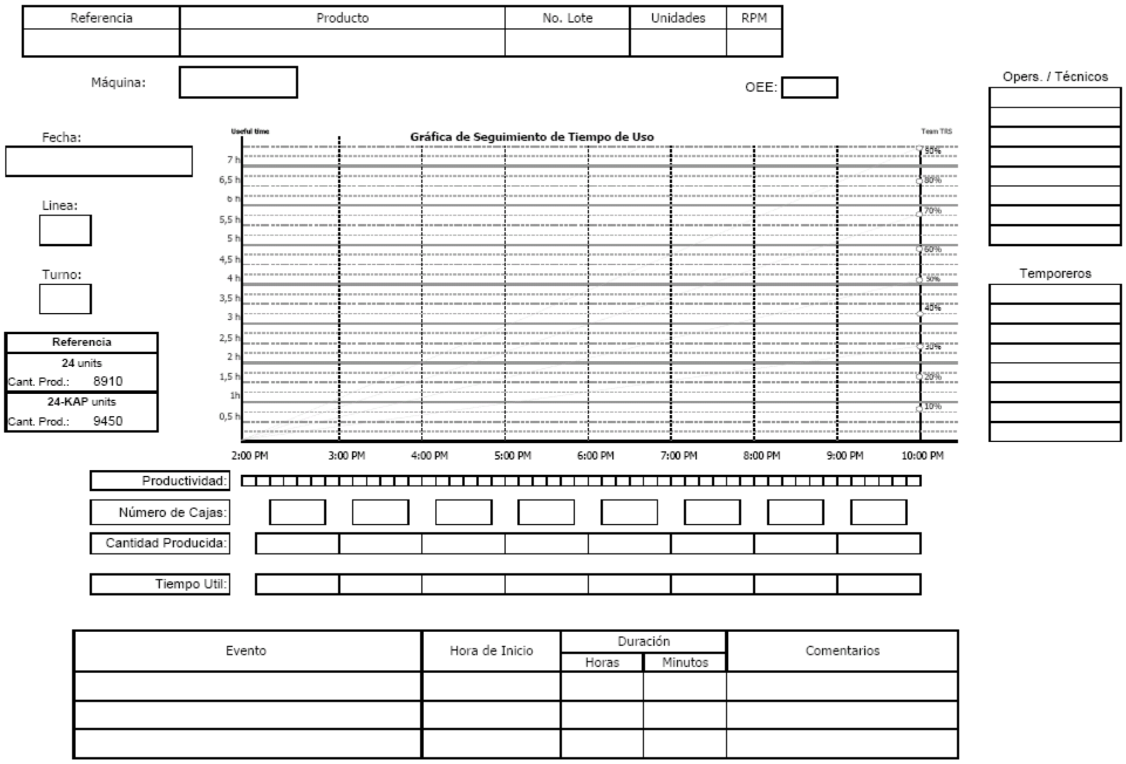

Figure 36. Paper-based Form.

Figure 36 presents the Paper Form for a packaging process with all the information needed for the usability test. This is a simplified version taken from the form in PEF. In section 3.1.2.3 you can find a brief explanation of the information in a PEF form. The upper part of the form shows number of reference, name of product, lot number, number of units and name of the machine. Besides, to analyze performance based on the information in the form, you have the revolutions per minute (RPM) of the motor. The graph in the center of the form shows the performance achieved by the operators for every hour in the shift.

Below the graph, is the section corresponding to the productivity data. It shows the hours for a shift. There is a group of 6 boxes to indicate the productivity; each one represents a time frame of ten minutes and should be filled with any of three options:

- No productivity, an event happened (×)
- Productivity (√)
- No information available ( )

The subject must put the number of boxes packaged for each hour; when the subject writes a number of boxes then he/she must calculate the produced quantity and the useful time. At the bottom of the picture is the information about the events different from packaging occurred during the shift that took more than 10 minutes. The events have event name, begin time, duration in hours and minutes and a description.

4.2.3 Participants

For the Pilot test we chose four people, two women and two men from the Advanced Data Management Group from the Electrical and Computer Engineering from University of Puerto Rico at Mayaguez. For the usability test we chose 21 (11 women and 10 men) students of the University of Puerto Rico at Mayaguez. They were selected from students with a major other than Software or computers. Their ages ranged from 19 to 29 years (Mean=24.4 years). Their experience with computers is more than five years. All of them without any experience with touch screen monitors. On average, they used computers on their studies for 3 hours per day.

4.2.4 Experimental Design

Before beginning the test, participants were asked to fill out a questionnaire about their work and computer experience (See Appendix A). They were asked to sign a Consent Form (See Appendix B). An orientation script (See Appendix C) was read to them. The script explained the objectives of the test, participants' role in the test, and test development. It also pointed out that their participation was voluntary and that they could stop the test whenever they wanted to. As a final point, the script clarified that all information about any user, as well as their participation, will remain anonymous.

A brief explanation on how to use the Paper Equivalent Form system was given to each of the participants before performing the test on each of the systems. The brief explanation of the three versions took an average of 8 minutes. In this explanation, participants were given details on how to work with each of the functions and user interfaces of the system.

Using the new TouchMouse is hard the first time, and it is really difficult to use it correctly during the first few minutes. So, users must be trained in the use of the TouchMouse because it is a new interaction technique. This has to be done before they begin to use the new interface, so the test will not be dominated by the effects of the users' effort with the new interaction technique and to gain information about the usability of the system [17][19]. Three elementary tasks were given to the participants with the purpose of training them in use the TouchMouse (See Appendix E).

After completing this introduction, they were asked to perform 6 tasks. The first three are for an additional training with the three devices and the process; the final three are for the real testing. The tasks are explained in appendixes F and G. In addition, participants were asked to rate their level of satisfaction with each system, and to indicate which system they would prefer for doing their tasks. Finally, they were asked to mention things they found easy and/or difficult to use in each version, and to comment on any particular thing they wished to, regarding each alternative.

4.3 Statistical Analysis

We used three tasks for each of the following activities: training for the TouchMouse, training for general training and testing the usability of the system.

4.3.1 Latin Square design

A Latin Square[11] design is an example of an incomplete block design where there is a single treatment and two blocking variables, each with the same number of levels [16]. Only a single treatment is applied within each combination of blocking variables. The Latin Square design is used to eliminate two nuisance sources of variability; this refers to systematically allow blocking in two directions.

[11] Latin square. December 2007. http://en.wikipedia.org/wiki/Latin_square

Thus, the rows and columns actually represent two restrictions of randomization, users and tasks. The model for the Latin Square is:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \varepsilon_{ijk} \tag{4}$$

Where:

- μ is the population mean.
- α_i is the main effect of the treatment i, i=1,…,n.

$$\sum_{i=1}^{n} \alpha_i = 0 \tag{5}$$

- β_j is a main effect of the first blocking variable j , j = 1, . . . , n.

$$\sum_{j=1}^{n} \beta_j = 0 \tag{6}$$

- γ_k is a main effect of the second blocking variable k, k = 1, . . . , n.

$$\sum_{k=1}^{n} \gamma_k = 0 \tag{7}$$

- ε_{ijk} is the subplot (individual) error distribution.

Latin square designs are reasonable choices when it is impossible to use each treatment level for the same combination of blocking levels. For example, if each task was performed with the devices in the same order, the treatment effect of a task would be confounded with the effect due to the order in which the devices were given. In this work, there are three devices (treatments) labeled Mouse, Paper and TouchMouse. The blocking variables refer to the "row effect" and the "column effect".

Mouse	Paper Form	TouchMouse
Paper Form	TouchMouse	Mouse
TouchMouse	Mouse	Paper Form

Table 2. Latin Square Distribution for the Test.

The Latin Square design is presented in Table 2. Each device appears in each row and in each column exactly once.

Task / User	**Task A**	**Task B**	**Task C**
User 1	Mouse	Paper Form	TouchMouse
User 2	Paper Form	TouchMouse	Mouse
User 3	TouchMouse	Mouse	Paper Form

Table 3. Latin Square Distribution with three users and three tasks.

Table 3 shows that we can test three users and three devices with three different tasks. This helps avoid a bias that we could be caused if we put all devices in the same order for all users, which would give priority to any of them.

4.3.2 Analysis of Variance (ANOVA)

We use statistical techniques such as the Analysis of Variance (ANOVA)[12] for determining if differences exist between three or more "groups", One-way ANOVA[13] and the associated F test [16]; The F test and subsequent ANOVA methodology involves the determination of differences for one group with multiple (typically, three or more) variations, as well as one variable, compared to multiple groups. We want to use the testing results to find the validity of our null hypothesis that the average time to accomplish a task does not vary significantly from a Mouse, a Paper Form and the TouchMouse.

4.4 Pilot Test results.

We used the pilot test to refine the questionnaire and tasks, because it is difficult to produce an unambiguous and unbiased test. Through the pilot test we learnt new ideas for improving the tasks, questions and gained sense of how long it takes for a subject to take the test. Thus, the pilot test demonstrated that we should:

- decrease the contrast for all the inactive arrows
- use a Stop-watch on the screen to start, end and repeat tasks. Also, to save the information in a Comma Separated Value file (CSV); this will help

[12] Analysis of variance. December 2007. http://en.wikipedia.org/wiki/ANOVA

[13] One-way ANOVA. December 2007. http://en.wikipedia.org/wiki/One-way_ANOVA

to have the information ready to open for a spreadsheet program to calculate, analyze and present the data

- improve the transparency of the TouchMouse
- redefined borders and use lighter colors
- added line separators to define the area for each arrow pointer
- maintain the location of the arrow pointer of the TouchMouse when the dragged movement is short
- make the task more comprehensible by a user
- change the number of arrows from 16 to 8

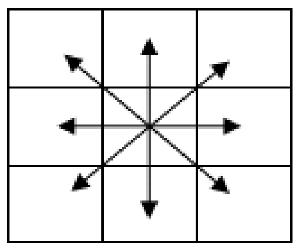

Figure 37. Pixel positions.

Figure 37 shows that there are only 8 contiguous positions for each pixel, this helped to improve the performance of the TouchMouse because with only 8 positions it is possible to increase the accuracy of the automatic arrow pointer.

4.5 User Test Results.

4.5.1 Initial tasks

All the users in this test have more than five years of experience with a computer mouse and all of them have been using papers in their lives, but none of them had ever used a touch screen. Thus, we used three elementary tasks as training for the TouchMouse.

Task	N	Mean (seconds)	Standard Deviation
1	21	57.71	22.21
2	21	58.19	24.41
3	21	88.57	42.15

Table 4. Mean and Standard Deviation for initial tasks.

Table 4 shows the mean and standard deviation for the initial tasks. You can find these tasks in the Appendix E. The mean describes the average of the data from the time in seconds spent by the users to accomplish a task, and the standard deviation describes the spread of the data. The confidence Intervals for these tasks are 95%.

4.5.2 Training tasks

We use three tasks as training for recording the packaging process. These training tasks help users to understand the processes and the interface of the forms

Training Task	N	Mean (minutes)	Standard Deviation
4	21	9.363	3.970
5	21	8.132	2.074
6	21	9.013	4.091

Table 5. Mean and Standard Deviation for the training tasks.

Table 5 shows the mean and standard deviation for these training tasks. The results present too much difference between the subjects; this can be attributed to the lack of knowledge of the interface, processes and the TouchMouse. You can find these tasks in the Appendix F.

The average time and standard deviation for each device with the training tasks are shown in Table 6. Also, it shows that the TouchMouse needs 52.46% more time than the Mouse to complete a task.

Device	N	Mean (minutes)	Standard Deviation	Used Time
Mouse	21	7.138	2.323	100.00%
Paper Form	21	8.486	1.861	118.87%
TouchMouse	21	10.883	4.606	152.46%

Table 6. Mean and Standard Deviation for each device in the training tasks.

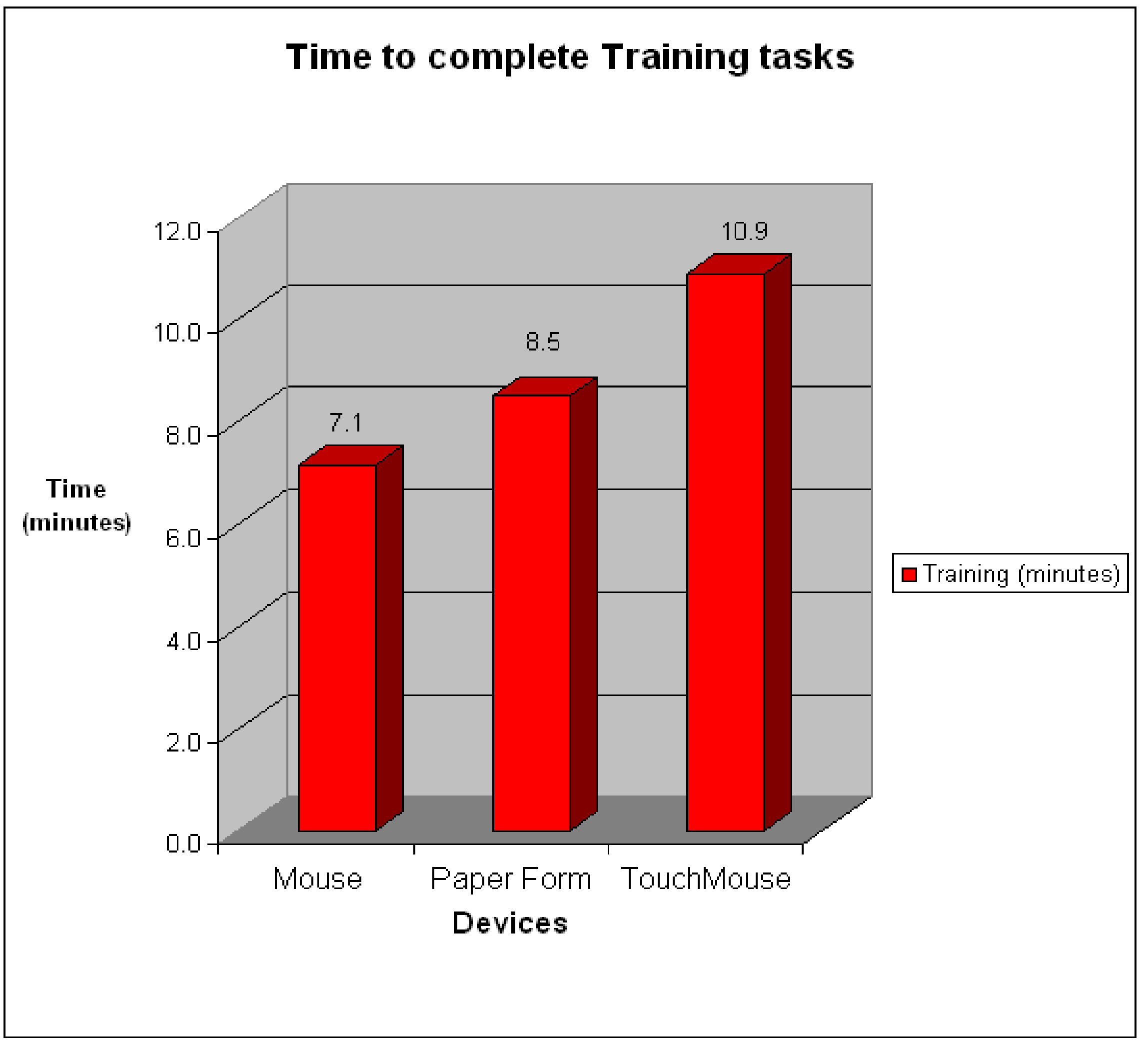

Figure 38. Training Task Comparison.

Figure 38 presents a comparison of the training tasks. The mouse presents the best results with 7.1 minutes in average to accomplish a task. The second is the paper form with 8.5 minutes in average to accomplish a task.

Although, the worst case is for the TouchMouse with 10.9 minutes, we expected problems to use a new technology. This was the reason to use three similar tasks for training and testing, because with this additional training we could separate the lack of knowledge of the interface from the lack of training with the TouchMouse.

Although, these tasks do not have any complex operation, the average time to complete a task with the Paper Form was 18.87% more than using the mouse. The average time to complete a task with the TouchMouse was 52.46% more than using the mouse. This 52.46% can be easily explained taking into account that none of the users had ever used touch screen technology and the lack of training with the TouchMouse.

Now, we attempted to use Analysis of Variance to validate the next hypotheses:

Null Hypothesis (H_0): There is no difference in the time to resolve the training tasks for the participants of the study.

Null Hypothesis (H_1): There is no difference in the device to resolve the training tasks for the participants of the study.

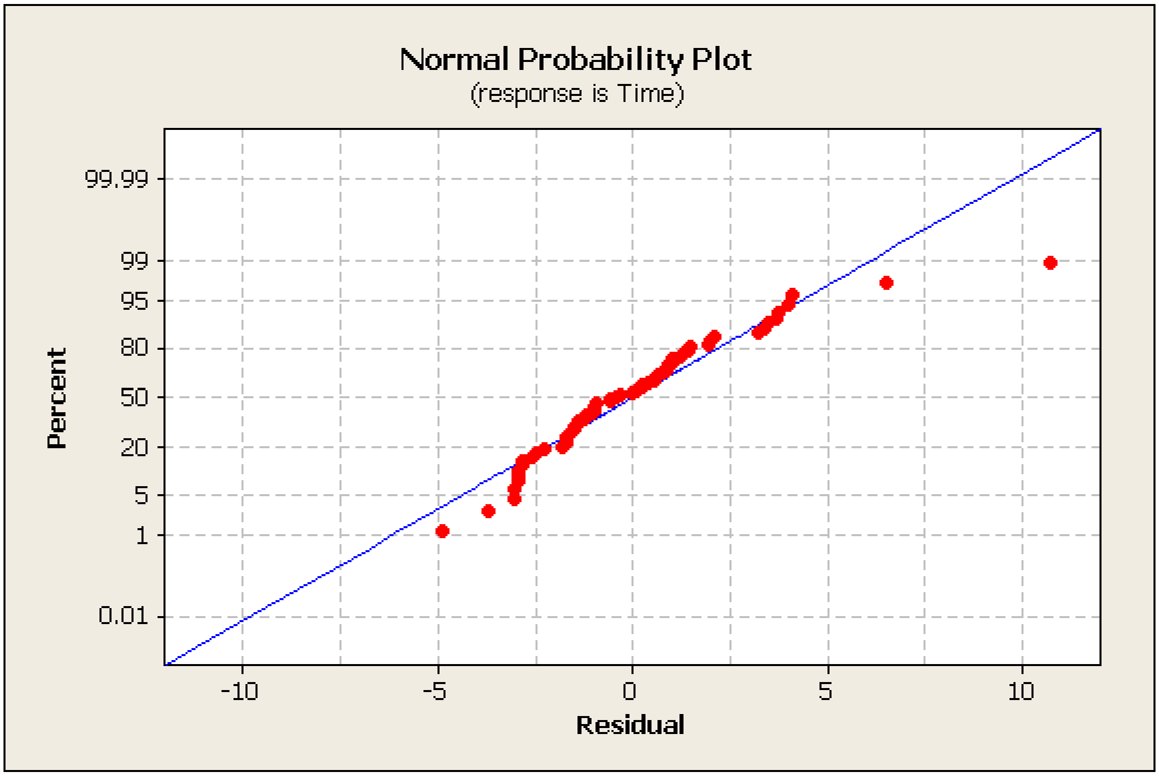

Figure 39. Normal Probability Plot of Training Tasks.

The general impression form examining the Figure 39 is that the error distribution is not normal. Also, the presence of many outliers can seriously distort the Analysis of Variance.

ANOVA assumes that although different samples may come from populations with different means, they have the same variance. We need to check the variances to use the Analysis of variance (ANOVA).

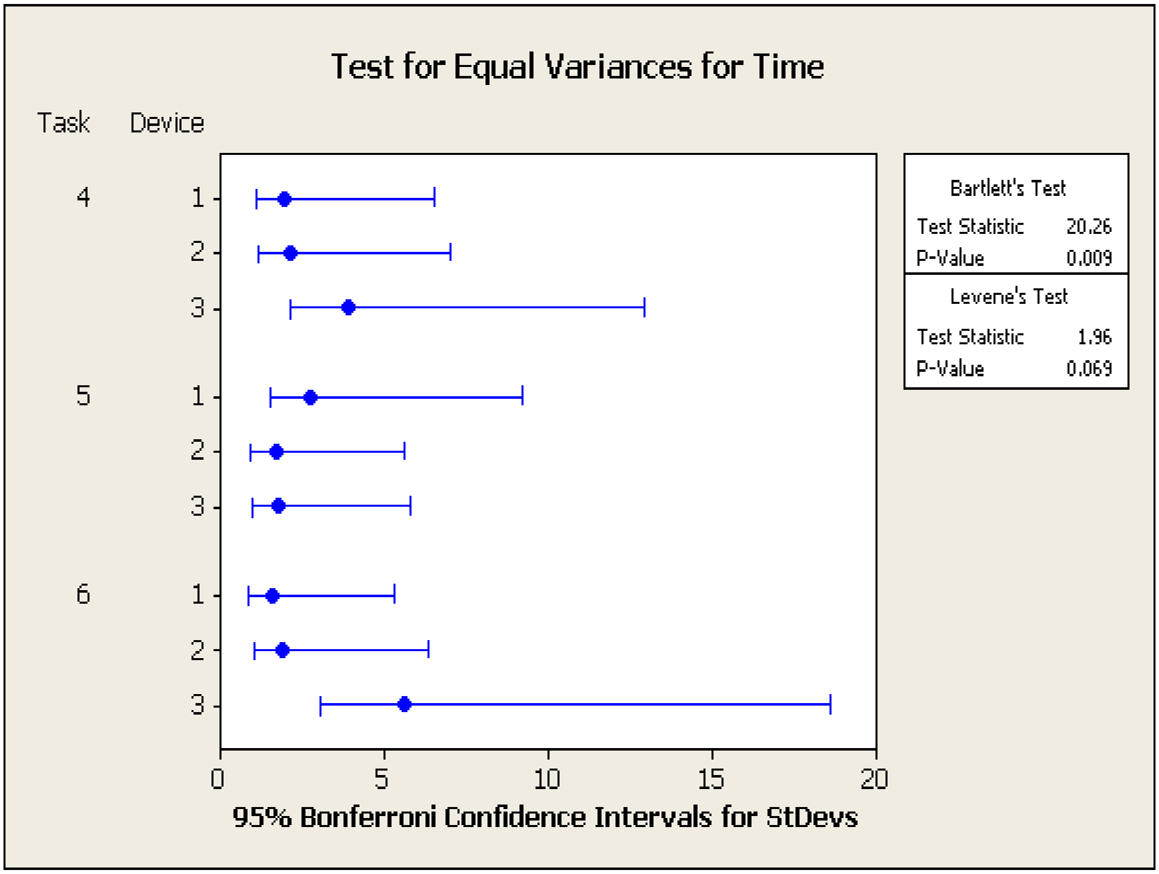

Figure 40. Test for Equal Variances for time with the Training Tasks.

Figure 40 presents the test for equal variances, Bartlett's test[14] gave a p-value of 0.009 and Levene's test[15] gave a p-value of 0.069, indicating that we can not use ANOVA because the training tasks have different variances. Data from training tasks is not appropriate for ANOVA therefore we can not accept nor reject the null hypotheses.

[14] Bartlett's test is used to test if k samples have equal variances. December 2007. http://en.wikipedia.org/wiki/Bartlett%27s_test

[15] In statistics, Levene's test is an inferential statistic used to assess the equality of variance in different samples. December 2007. http://en.wikipedia.org/wiki/Levene%27s_test

4.5.3 Testing tasks

We use three tasks for testing the recording of the packaging process. These are the real task that we used to calculate the efficacy of the TouchMouse versus Mouse versus the Paper Form. At this moment, subjects know the processes and the interface for the forms.

Testing Task	N	Mean (minutes)	Standard Deviation
7	21	6.831	1.359
8	21	6.665	1.457
9	21	6.815	1.910

Table 7. Mean and Standard Deviation for testing tasks.

Table 7 shows the mean and standard deviation for the three testing tasks. Although, the difficulty for training and testing tasks was similar, the mean for testing tasks are quite lower than the training tasks. Besides, the standard deviation for these tasks showed less variation between the users when they accomplished the tasks.

Device	N	Mean (minutes)	Standard Deviation	Used Time
Mouse	21	6.024	1.463	100.00%
Paper Form	21	7.673	1.372	127.38%
TouchMouse	21	6.614	1.465	109.80%

Table 8. Mean and Standard Deviation for each device in the testing tasks.

The average time for each device with the training tasks are shown in Table 8. Although the difficulty for training and testing tasks was similar, the means for testing tasks are quite lower than the training tasks. Besides, the standard deviation for these tasks showed less variation between the devices.

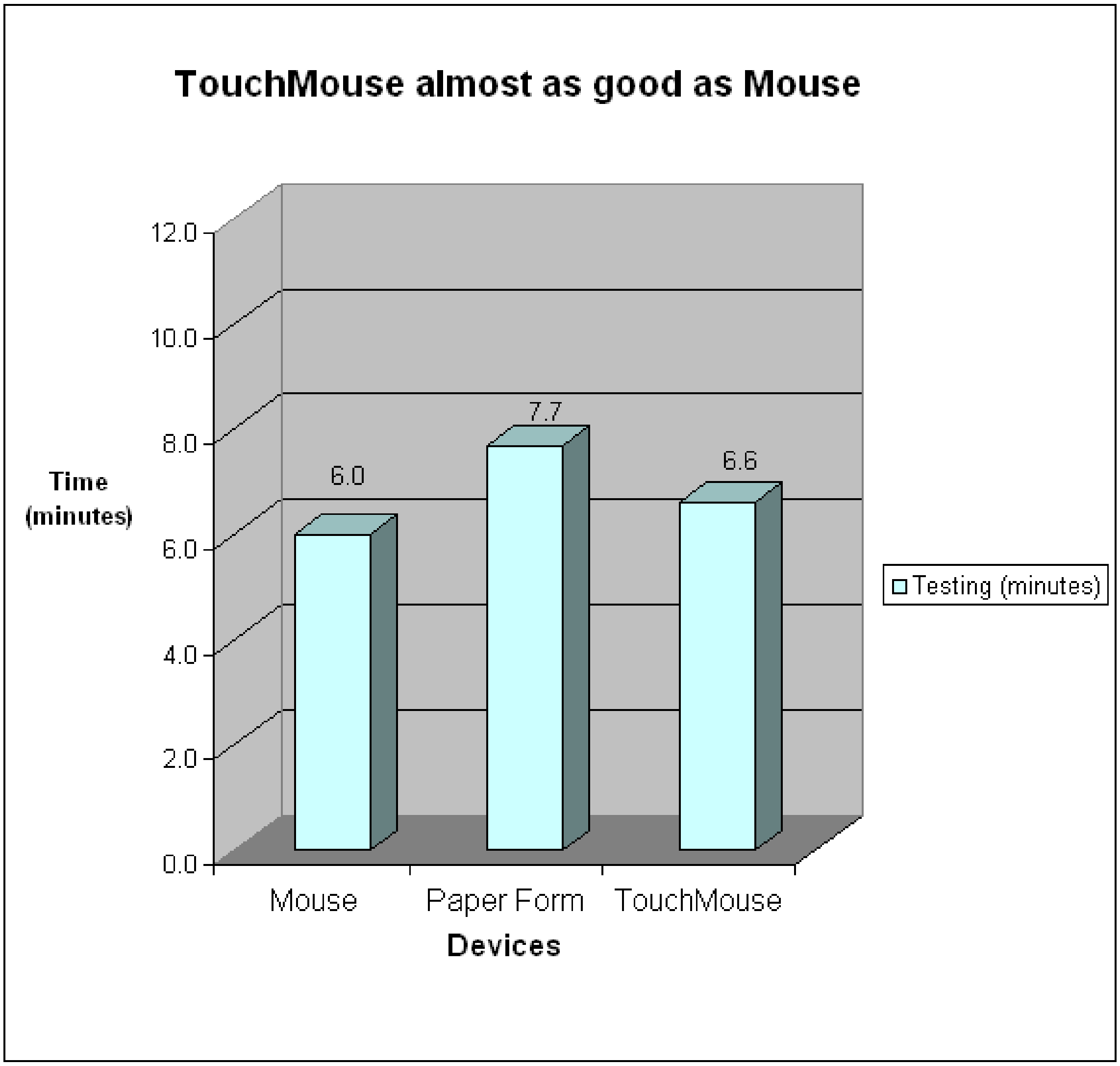

Figure 41. Testing Task Comparison.

The average time to complete a task with the Paper Form was **27.38%** more than using the mouse. The average time to complete a task with the TouchMouse was **9.8%** more than using the mouse. Given that the initial gap between these two devices was 57%, get a 9.8% can be considered promising results for the TouchMouse. Figure 41 presents a comparison of the testing tasks.

Now, we used Analysis of Variance (ANOVA) to validate the following hypotheses:

Null Hypothesis (H_0): There is no difference in the time to complete the testing tasks for the participants of the study.

Null Hypothesis (H_1): There is no difference in the device to complete the testing tasks for the participants of the study.

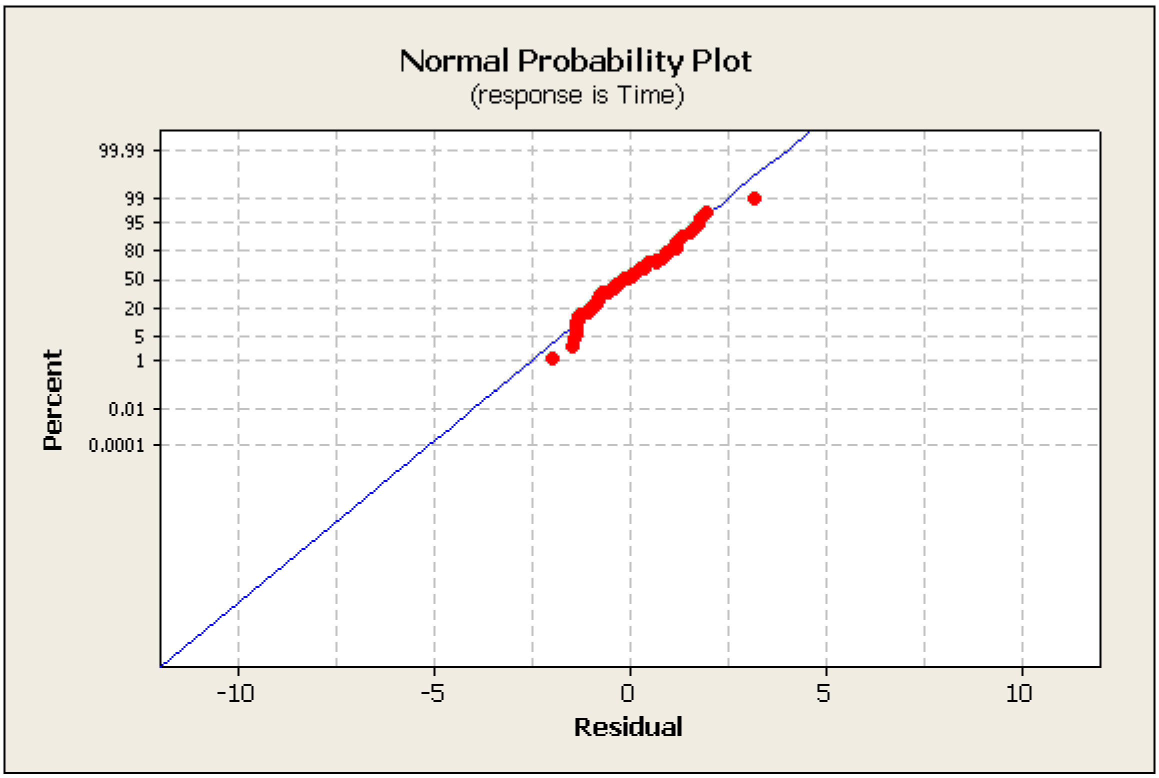

Figure 42. Normal Probability Plot of Testing Tasks.

The tendency of the Figure 42 to bend slightly on the left and right side is indicating that the data may be appropriate for ANOVA.

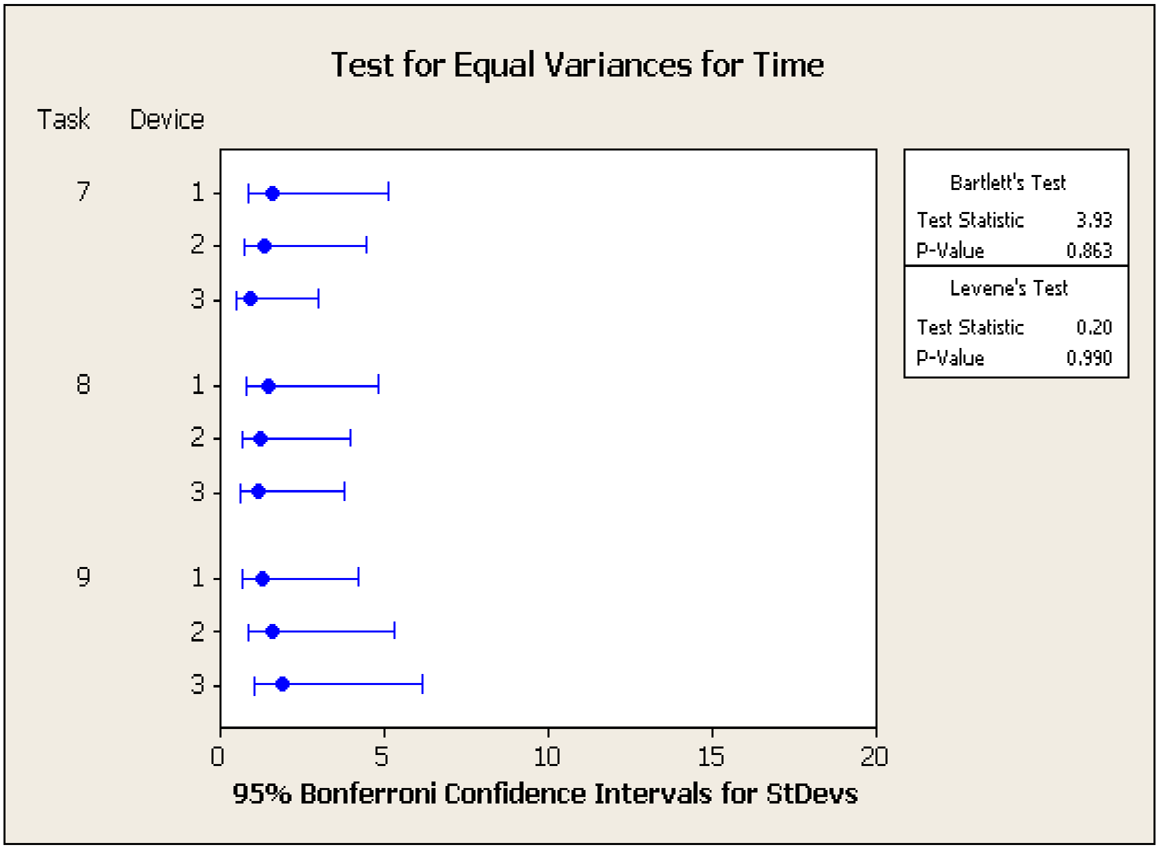

Figure 43. Test for Equal Variances for time with the Testing Tasks.

ANOVA assumes that although different samples may come from populations with different means, they have the same variance. Figure 43 shows the test for equal variances, Bartlett's test presents a p-value of 0.863 and Levene's test presents a p-value of 0.990. These tests indicate that we can use ANOVA because the training tasks have similar variances. So, we can accept or reject the null hypotheses.

Source	**Degrees of Freedom**	**Sum of Squares**	**Mean Square**	**F**	**P**
Task	2	0.35	0.18	0.07	0.933
Error	60	152.34	2.54		
Total	62	152.69			

Table 9. One-Way ANOVA for the testing tasks – Time versus Task.

Table 9 presents the Analysis of Variance (ANOVA) for the testing tasks with task as factor. The p-value of 0.933 for this test indicates that our hypothesis "H_0: There is no difference in the time to complete the testing tasks for the participants of the study" should be accepted. We must assume that tasks have a comparable difficulty.

Source	**Degrees of Freedom**	**Sum of Squares**	**Mean Square**	**F**	**P**
Device	2	29.33	14.66	7.13	0.002
Error	60	123.37	2.06		
Total	62	152.69			

Table 10. ANOVA for the training tasks – Time versus Device.

Table 10 presents the ANOVA for the testing tasks with device as factor, where the p-value of 0.002 for this test indicates that our hypothesis "H_1: There is no difference in the device to complete the testing tasks for the participants of the study" must be rejected.

The Analysis of Variance (ANOVA) shows that there is a statistical difference between TouchMouse, Mouse and Paper Form. Thus, we concluded that all the devices have influence in the results to complete a task in the testing tests.

Factor	**Type**	**Levels**	**Values**
Task	fixed	3	7,8,9
Device	fixed	3	1,2,3
Replica	fixed	7	1,2,3,4,5,6,7

Table 11. ANOVA for the training tasks – Factors and Levels.

Source	Degrees of Freedom	Sum of Squares	Adjust Sum of Squares	Adjust Mean Square	F	P
Task	2	0.352	0.352	0.176	0.13	0.881
Device	2	29.326	29.326	14.663	10.55	0.000
Replica	6	50.734	50.734	8.456	6.08	0.000
Error	52	72.282	72.282	1.390		
Total	62	152.694				

Table 12. ANOVA - General Linear Model: Time versus Task, Device, Replica.

We will use ANOVA - General Linear Model presented in Table 11 and Table 12 to get a better understanding of the influence of the task and devices. For the testing tasks we found that the p-value from the ANOVA with Task as factor is 0.881; this indicates that our hypothesis should be accepted. We concluded there is no difference in the time to resolve the testing tasks for the participants of the study. We found that the p-value for the factor Device is 0.000; this indicates that our hypothesis is rejected. We concluded there is difference in the device to resolve the testing tasks for the participants of the study.

Device	N	Mean (minutes)	Standard Deviation	Used Time
Mouse	21	6.024	1.463	78.5%
Paper Form	21	7.673	1.372	100.0%
TouchMouse	21	6.614	1.465	86.2%

Table 13. Percentage of average time to perform a task.

Table 13 indicates the Paper Form takes 21.5% more time in average than the Mouse and 13.85% more time in average than the TouchMouse. There is only **7.7%** of average difference between accomplishing a task by using a Mouse and the TouchMouse.

4.6 Comparative results

Device	Training (minutes)	Testing (minutes)	Improvement
Mouse	7.138	6.024	18.50%
Paper Form	8.486	7.673	10.58%
TouchMouse	10.883	6.614	64.54%

Table 14. Improvement from Training to testing.

Table 14 indicates that a user has the fastest improvement in his/her performance by using the TouchMouse. All users have had more than five years of experience with the mouse; in that case the improvement of 18.5% can be attributed to the speed of the user to learn how to work with the system.

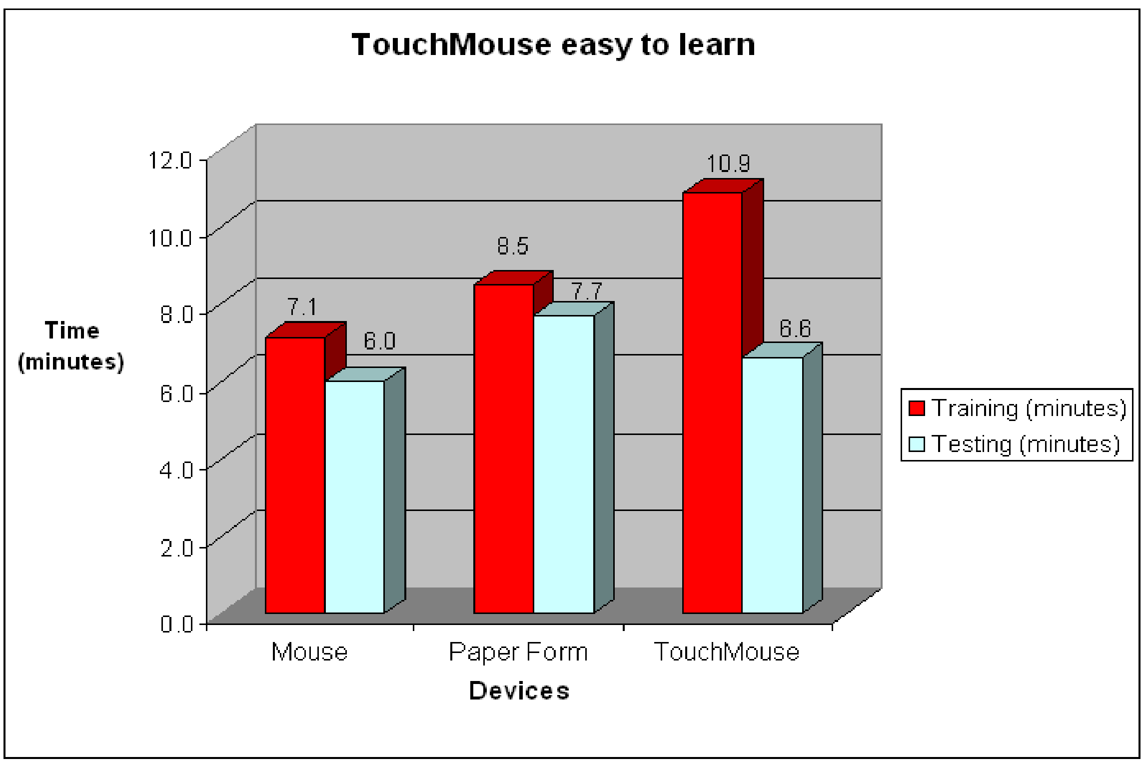

Figure 44. Training and Testing Task Comparison.

Figure 44 presents a comparison between the training and testing tasks. There are two important facts to analyze: first, none of the users had had experience with touch screens, and second, the improvements with the mouse and paper form were 18.5% and 10.58% respectively. Based on that, it can be concluded that the improvement of 64.54% is mostly due to the intrinsic ease of learning to use the TouchMouse.

4.7 Results of algorithms.

In this section, we present the results from the algorithms for the automatic arrow pointer.

4.7.1 Simple Algorithm.

Every finger movement makes changes in the arrow position. Thus, it is really difficult to click and to keep the arrow fixed at the same time. It is complicated to keep a selected position for the arrow because a click can be interpreted as a drag. Since there are 16 achievable positions for the arrow, the area of changes for each one is very small so the finger can jump from one area to another without noticing.

4.7.2 Extreme Points Algorithm.

Performance was not satisfactory. Sometimes, the algorithm for moving the arrow fails to achieve the correct position. Good results but there are a lot of room for improvement.

4.7.3 Moving Average Algorithm.

Although, we tested many n values to improve the accuracy of the automatic arrow pointer, we did not find a suitable n value with an outstanding performance. Besides, this algorithm has a problem due to the closed cyclic assignment of the positions. For instance, if the user is moving between the one and eight position alternately then the average is four or five; this makes an unstable behavior for the arrow pointer.

4.7.4 Score Algorithm.

Performance is better than the Moving Average Algorithm. Sometimes when user movements are very fast it is possible to see an unexpected change in the arrow position. This algorithm offers an improvement in the results, but due to the problems presented previously, there is a lot of room for improvement. An n value of 80 lets the user to move freely in short movements without changing the arrow and it is very good for long movements.

4.7.5 Linear Regression Algorithm.

Experiments showed that the direction of the arrow is reliable. An n value of 135 lets the user to move freely in short movements without changing the arrow; besides, this algorithm is the best for long movements. Thus, we can conclude that the Linear Regression Algorithm presents the best performance of all algorithms tested.

4.8 Results of Questionnaires

The range of values for the questionnaire is from 1 to 5. If the user could not answer a question then he used Not Available (NA). The questionnaire is in Appendix D.

Classification	Order	Description	Average
System Design	1	Have all the necessary controls.	4.27
	2	Difficulty to accomplish the tasks	**3.78**
	3	Motivation to use the system	4.04
Display Effectiveness	4	Reading characters on the screen	4.02
	5	Task selection is easy	**3.95**
	6	Organization of information	4.21
Learning	7	The grouping of the menu options is logical	4.37
	8	Learning to operate the system	4.53
	9	Performing tasks is straightforward	4.84
	10	Help messages on the screen	4.20
System Information	11	Menus have coherent names	4.67
	12	Prompts for input	4.75
	13	Error messages	4.07

Efficiency	14	System speed	4.70
	15	System reliability	4.95
	16	Correcting your mistakes	**3.68**
	17	Responsiveness	4.09
	18	Attempts to put correct data	**3.51**

Table 15. Results from Questionnaires.

It is possible to explain some of the difficulties because users had to work with an interface they had never seen and they did not have previous knowledge of our project.

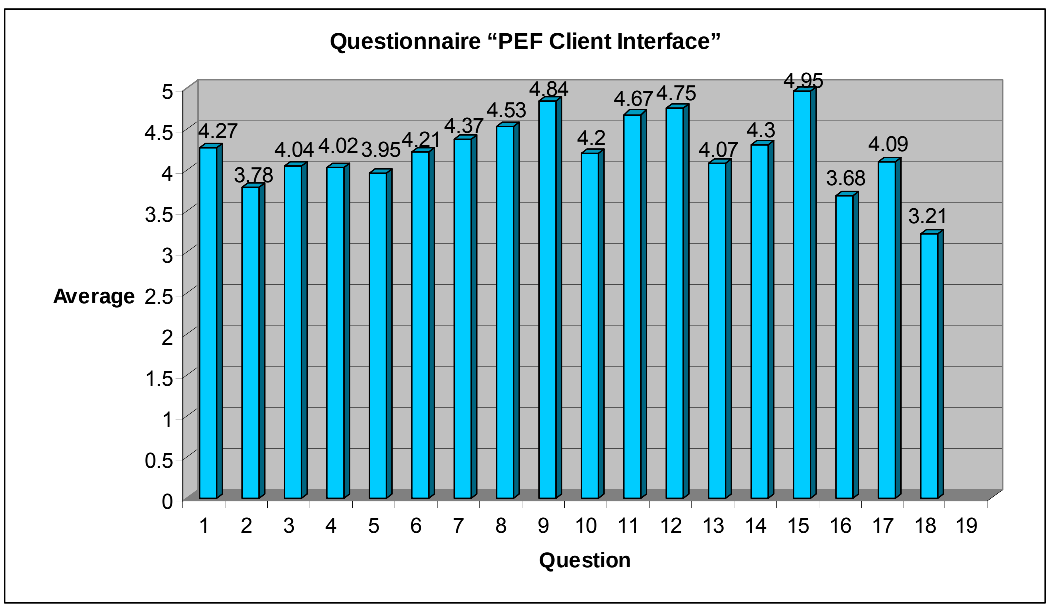

Figure 45. Results from Questionnaires.

Table 15 and Figure 45 show that the principal problems with the interface are in difficulty to accomplish the tasks, task selection, correcting mistakes and attempting to put correct data with values of 3.78, 3.95, 3.68 and 3.51

respectively. We used the values of the answers from the users to calculate the average. If a user's answer was NA then his/her answer was not used to calculate the average.

Moreover, Table 15 and Figure 45 show that the better advantages with the interface are in system reliability, performing tasks is straightforward, prompts for input and system speed with values of 4.95, 4.84, 4.75 and 4.70 respectively. This is understandable because a graphical user interface can provide with more information than a paper without help of any other person. Also, a computer-based solution can make processes automatically and can backup the information in real time.

5 CONCLUSIONS AND FUTURE WORK

5.1 Research Conclusion

The results of the study described in the previous chapter indicate that the TouchMouse is almost as good as the Mouse, this is a very amazing result because it contradicts our expectations that users would have a very huge difference in time with the TouchMouse because lack of training. Besides, both of them are better than the Paper Form.

Moreover, the TouchMouse is easy to learn. This can be explained by the improvement of 64.54% in the TouchMouse use from training tasks to testing tasks, while mouse and paper form were 18.5% and 10.58% respectively. The new TouchMouse leverages natural abilities of users, maintains the user gaze in the screen, it does not require additional desk space, and allows direct manipulation of small objects.

We did not have to make any changes in the existing PEF application. This implies that we can develop a single implementation for different pointing devices such as mouse, pen or the finger with the TouchMouse. A single implementation implies that the TouchMouse can be used where it is not possible to make changes in applications, either because of the lack of the source code or because it is not cost-effective.

Additionally, we defined and characterized the user problems with the PEF application. Although, the touch screen is more user-friendly than other input devices, overall training time for human users is required. In any case, touch screen technology can also help to make learning more intuitive and interactive, which can lead to a more beneficial training experience for both operators and managers.

We answered the next questions with this research:

- Can the hardware commercially available be used in our case study's conditions? In the last years the cost of touch screen monitors has dropped considerably; now, these cost reductions have made it possible for smaller businesses to take advantage of touch screen technology in conventional software applications.
- Are there any conditions that require special considerations? The TouchMouse has some difficulties because of the delay for the device to response to user tasks; a native implementation of the TouchMouse will outperform this issue.
- What can we do to compensate the required time to adopt this new technology? We already compensated the time to adopt this new technology, because the performance to achieve the tasks by the operators has improved by doing their tasks using the computer instead of the paper-based model. Besides the touch technology requires less training and is more intuitive to use.
- Which additional features can we implement to improve work conditions of the operators? To put the mouse in the new finger position instead of having to drag the TouchMouse to move it; this will be very useful and faster when the user needs to move the TouchMouse longer distances.

Design and develop configuration software will help users to adjust the TouchMouse to their needs.

5.2 Future Work

This section presents ideas and possibilities to improve the TouchMouse; they were generated by the work during this research, as well as suggestions received during the usability study.

Native code has several advantages over managed code such as faster execution and smaller code size. Thus, develop a native implementation for the TouchMouse will allow to use the full functionality of a touch screen; there will be faster response with lower delay time.

Many applications require a keyboard for alphanumeric data entry that it could be a problem because it has similar problems as a mouse; a keyboard requires a surface or can trap dust and dirt affecting clean rooms. Also, keyboards can be lost in public places. Our idea is to develop an on-screen keyboard for the alphanumeric data entry to continue exploring the advantages of touch screen technology.

Most the screens have a square design where the corners are the farthest from the center. A TouchMouse with only four location of the arrow pointer placed in the diagonals can reach every place in the screen. Thus, another idea is to change the number of positions of the arrow pointers from 8 to 4 and test the relative efficacy of the TouchMouse with 8 arrow pointer positions versus the TouchMouse with 4 arrow pointer positions.

Also, conduct on-site usability tests because these tests can expand the usability of the TouchMouse and to find out how the TouchMouse is actually used after installation. Moreover, it is feasible to find problems and inefficiencies and get some ideas about how to improve it.

REFERENCES

[1] Badía-Reyes, René D., A Framework for Auditable, Paper-Equivalent Electronic Forms using Data Logging, Secure Access Controls, and Electronic Certificates., Master of Science thesis in Computer Engineering at University of Puerto Rico Mayaguez Campus, June 2006.

[2] Fitzmaurice, George., Khan, Azam., Piek, Robert., Buxton, Bill., Kurtenbach, Gordon., Tracking menus., Proceedings of the 16th annual ACM symposium on User interface software and technology., pp. 71-80. November 2003.

[3] Food and Drug Administration, U.S. Department of Health And Human Services, 21 CFR, Part 11, Electronic Records; Electronic Signatures, Federal Register, vol. 62, no. 54, pp. 13430-13466, March 1997.

[4] Futo, Ivan., A General Purpose Front-Office System for eGovernment Communications, 28th International Conference on Information Technology Interfaces 2006, pp. 333-338. October 2006.

[5] Gross, Mark D., Yi-Luen Do, Ellen., Ambiguous intentions: a paper-like interface for creative design., Symposium on User Interface Software and Technology. Proceedings of the 9th annual ACM symposium on User interface software and technology., pp. 183-192. 1996.

[6] Grossman, Tovi., Hinckley, Ken., Baudisch, Patrick., Agrawala, Maneesh., Balakrishnan, Ravin., Pen: Hover widgets: using the tracking state to extend the capabilities of pen-operated devices., Proceedings

of the SIGCHI conference on Human Factors in computing systems CHI’06., pp. 861-870. April 2006.

[7] Hartswood, Mark., Procter, Rob., Rouncefield, Mark., Slack, Roger., Making a Case in Medical Work: Implications for the Electronic Medical Record., Computer Supported Cooperative Work 12., pp. 241266. 2003.

[8] Hinckley, Ken., Sinclair, Mike., Touch-sensing input devices., Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit., pp. 223-230. May 1999.

[9] Holzinger, Andreas., Usability engineering methods for software developers., Communications of the ACM, Volume 48 Issue 1., pp. 71-74. January 2005.

[10] Hornbaek, Kasper., Frokjaer, Erik., Methods & usability: Comparing usability problems and redesign proposals as input to practical systems development., Proceedings of the SIGCHI conference on Human Factors in computing systems CHI ’05., pp. 391-400. April 2005.

[11] Hsieh, Gary., Wood, Kenneth., Sellen, Abigail., Personal information management: Peripheral display of digital handwritten notes., Proceedings of the SIGCHI conference on Human Factors in computing systems CHI ’06., pp. 285-288. April 2006.

[12] Java Programming Language. Sun Microsystems, Inc., Java Technology, December 2007; java.sun.com.

[13] Kurtenbach, Gordon., Fitzmaurice, George., Baudel, Thomas., Buxton, Bill., The design of a GUI paradigm based on tablets, two-hands, and transparency., Proceedings of the SIGCHI conference on Human factors in computing systems., pp. 35-42. March 1997.

[14] Masui, Toshiyuki., An efficient text input method for pen-based computers., Proceedings of the SIGCHI conference on Human factors in computing systems., pp. 328-335. January 1998.

[15] McClard, Anne., Somers, Patricia., Unleashed: Web tablet integration into the home., Proceedings of the SIGCHI conference on Human factors in computing systems., pp. 1-8. April 2000.

[16] Montgomery, Douglas C., Design and Analysis of Experiments, Wiley, 6th Edition, ISBN: 978-0-471-48735-7, 660 pages, December 2004.

[17] Nielsen, Jackob., Usability Engineering, Academic Press/AP Professional, Cambridge, MA, ISBN 0-12-518406-9, 362 pages. 1993.

[18] Po, Barry A., Fisher, Brian D., Booth, Kellogg S., Smart interaction techniques 1: Comparing cursor orientations for mouse, pointer, and pen interaction., Proceedings of the SIGCHI conference on Human factors in computing systems., pp. 291-300. April 2005.

[19] Rubin, Jeffrey, Handbook of Usability Testing, John Wiley and Sons, New York, NY, ISBN 0-471-59403-2, 1994.

[20] Sutcliffe, Alistair., On the effective use and reuse of HCI knowledge., ACM Transactions on Computer-Human Interaction (TOCHI), Volume 7 Issue 2., pp. 197-221. June 2000.

[21] Wong, Cathy., Geiger, Glen., Derman, Yaron D., Busby, Carolyn R., Carter, Michael W., Redesigning the medication ordering, dispensing and administration process in an acute care Academic Health Sciences Center., Proceedings of the 2003 Winter Simulation Conference., pp. 1894-1902. 2003.

[22] Yee, Ka-Ping., Late breaking result papers: Two-handed interaction on a tablet display, CHI'04 extended abstracts on Human factors in computing systems, pp. 1493-1496. April 2004.

APPENDIX A: Questionnaire “General Information”

Please answer the following questions completely.

Participant ID: _________

About you:

1. Age Range: 18-24___ 25-34___ 35-44___ 45-54___ 55+___
2. Gender: Male _____ female _____
3. Occupation: ____
4. Do you need glasses or contact lenses for:
   - Reading? Yes_____ No_____
   - Writing? Yes_____ No_____
   - Using computers? Yes_____ No_____

About your background and experience:

5. Which way do you learn best?
   - Trial and Error: _____
   - By Example: _____
   - Reading Documentation: _____
   - Other: ________________________________
6. How long have you been using personal computers?
   - Less than 1 year: _____
   - 2 - 3 years: _____
   - 3 - 5 years: _____
   - 5+ years: _____

7. In a typical day, how often do you use a computer?

    Hours:_____

8. Do you have any experience with a touch-screen monitor?

    Yes_____ No_____

If the answer (8) was “No”, please continue with question 11.

9. How long have you been using a touch-screen monitor?

    Years_____ Months_____

10. In a typical day, how often do you use a touch-screen monitor?

    Hours:_____

11. Do you have any other comments or suggestions? If you have any other ideas for this study that you would like to suggest, please let me know.

____

____

____

____

Thank you for filling out this questionnaire. Your input will help us to ensure the success of this work!

APPENDIX B: “Consent Form and Information Letter”

This study is being undertaken by Luis Leon from the Advanced Data Management Group, University of Puerto Rico at Mayaguez Campus. The purpose of the project is to assess the usability of the new interface for the Paper Equivalent Form project (PEF) and its target users; in other words, assessing how easy or difficult it is for people to use the new version of PEF. Data from this research will be used to make improvements. A final report on the research will be presented to the University community, and the results will be written up for publication or presentation.

As a participant in this study, you will be asked to complete a talk aloud usability test. This is the most common form of usability evaluation. You will carry out a specified set of tasks while thinking out loud about what you are doing. You will be seated at a computer with a test facilitator. The session will be audio-recorded. You will also be asked to complete a questionnaire before and one after the tasks are given. It is expected that the study will take 1 hour of your time.

All information that you are providing will be held in confidence and you will not be identified in any way in the final report. There will be no way to link your name on this consent form with any of the other information gathered during your participation in this study.

An observer will be present in the room for the period of testing and will take notes. The observer/ note taker will not reveal any information about individual participants. The researcher will be the only person with access to the digital audio. Certain comments may be quoted or paraphrased in the final report, but no identifying information will be given beyond selected demographic characteristics. You can withdraw from this study at any time for any reason without explanation and any collected data will be withdrawn and not included in the study.

If you have any questions or concerns about this study please contact:

Luis Leon, M.S. Student, Computer Engineering

Advanced Data Management Group

Electrical and Computer Engineering

University of Puerto Rico at Mayaguez Campus

Email: luis.leon@ece.uprm.edu

I have read and understand the above information, and agree to participate in this study.

Participant's Name: ____

Participant's Signature: ______________________________________

Date (yyyy-mm-dd): __________________________________

APPENDIX C: “Orientation Script”

Hello, my name is Luis Leon. I will work with you today I asked you to be here, because I need your help to do a usability test of the Paper Equivalent Form project (PEF). You will do typical tasks with PEF and fill some questionnaires. This means it is necessary to achieve the same speed and concentration as you always do. Do the best you can and do not worry about the results, because we are testing PEF not you.

You can ask questions any time, but it is possible that I do not answer some of them, because for testing purposes it is quite relevant to watch the performance of the system when it is been used for one person.

During this session, I will ask you to fill some questionnaires and it is very important that you will do as accurate as possible. My role here is to find advantages and disadvantages of PEF from your point of view. While you are working I will be taking notes and times. Also, I will use an ID code instead of your name for the questionnaires. Do you have any questions? Then, let’s begin!

APPENDIX D: Questionnaire “PEF Client Interface”

Please read the following sentences and rate the usability of the system.

Try to respond to all the items.

For items that are not applicable, mark the NA column.

System Design	**Description of the lowest value**	**NA**	**1**	**2**	**3**	**4**	**5**	**Description of the highest value**
Have all the necessary controls.	Deficient							Adequate
Difficulty to accomplish the tasks	Difficult							Easy
Motivation to use the system	Boring							Interesting

Display Effectiveness		**NA**	**1**	**2**	**3**	**4**	**5**	
Reading characters on the screen	Hard							Easy
Task selection is easy	Not At All							Very Much
Organization of information	Confusing							Very Clear

Learning	**Description of the lowest value**	**NA**	**1**	**2**	**3**	**4**	**5**	**Description of the highest value**
The grouping of the menu options is logical	Bad							Good
Learning to operate the system	Difficult							Easy
Performing tasks is straightforward	Never							Always
Help messages on the screen	Unhelpful							Helpful

System Information	**Description of the lowest value**	**NA**	**1**	**2**	**3**	**4**	**5**	**Description of the highest value**
Menus have coherent names	Never							Always
Prompts for input	Confusing							Clear
Error messages	Unhelpful							Helpful

Efficiency	**Description of the lowest value**	**NA**	**1**	**2**	**3**	**4**	**5**	**Description of the highest value**
System speed	Too Slow							Fast Enough
System reliability	Unreliable							Reliable
Correcting your mistakes	Difficult							Easy
Responsiveness	Bad							Good
Attempts to put correct data	Many							Few

List the most negative aspect(s):

1. ____
   ____
2. ____
   ____
3. ____
   ____

List the most positive aspect(s):

1. ____
   ____
2. ____
   ____
3. ____
   ____

APPENDIX E: Initial Tasks

Please read the following steps completely.

Please, use the **TouchMouse** to complete the next tasks; click **Start** in the Stopwatch at the beginning of each task and click **Stop** after finish the task.

Task 1

Please, click **Start** in the Stopwatch; now open the file “**Tasks1.txt**”.

Write the word “**Training**”; close the file; when the window “Do you want to save changes?” appears then click **NO**.

Click **Stop** in the Stopwatch.

Task 2

Please, click **Start** in the Stopwatch; now open the file “**Tasks2.txt**”, maximize the file.

Write the sentence “**Task is done**” and close the file; when the window “Do you want to save changes?” appears then click **YES**.

Click **Stop** in the Stopwatch.

Task 3

Please, click **Start** in the Stopwatch; now open a Windows Explorer; In **My Computer**, go to the local drive **C**, open the folder **enough-Training** and then open the folder **Test**; select and open the file “**task3.txt**”. Read it and close it.

Click **Stop** in the Stopwatch.

APPENDIX F: Training Tasks

Task 4 ➔ Mouse

Please read the following steps completely. Use the **Mouse** to complete the next tasks.

This is your first day working for the company and the manager told you to put the packaging information about today's shift in the System. You need to store the data received from the operator in charge.

Please, click **Start** in the Stopwatch; Date is: **October 22, 2007**.

The operator said "this is the line **1** and shift **2**, we are working in the packaging process of this item, Reference: **374385-14**, product: **Pain Relief 24's**, lot number: **213674C**, machine name: **Blisters**, number of units: **24**, RPM of the machine: **32**."

The starting time for the shift 2 is 2:00 pm.

2:00 pm ➔ number of boxes: **295**.

The manager called and said "there was a mistake and the lot number is **216374F**".

3:00 pm ➔ number of boxes: **600**.

It is 4:00 pm and you have to leave because there is a **Meeting** schedule for one hour and ten minutes.

Since there was no work during the third hour, you must put the same number of boxes of the 3:00 pm into the 4:00 pm hour. So, 4:00 pm ➔ number of boxes: **600**.

5:00 pm ➔ number of boxes: **750**.

It is 6:00 pm and you have a **Break** of 30 minutes to eat something.

6:00 pm ➔ number of boxes: **920**.

7:00 pm ➔ number of boxes: **1250**.

8:00 pm ➔ number of boxes: **1560**.

Last hour boxes: **1715**. The shift has finished. Click **Stop** in the Stopwatch.

Task 5 ➔ Paper Form

Please read the following steps completely. Use the **Paper Form** to complete the next tasks.

This is your second day working for the company and you are in charge of entering the information in the System.

Please, click **Start** in the Stopwatch; Date is: **October 23, 2007**.

You must ask the operator about today's packaging. The operator said "this is the line **1** and shift **2**, we are working in the packaging process of this item, Reference: **37485-14**, Product: **Pain Relief 24's**, lot number: **216374F**, machine name: **Blisters**, number of units: **24**, RPM of the machine: **32**."

Remember to calculate Produced Quantity (PQ)

$$PQ = NumberOfBoxes * units$$

Remember to calculate Useful Time (UT) for every hour. $UT = \frac{PQ}{RPQ}$

The Reference Produced Quantity (RPQ) for 24 units is **8910**.

The starting time for the shift 2 is 2:00 pm.

2:00 pm ➔ number of boxes: **310**.

3:00 pm ➔ number of boxes: **470**.

4:00 pm ➔ number of boxes: **790**.

It is 5:00 pm and you have a **Yoga Class** schedule for one hour. This is a Meeting and training event.

Since there was no work during the fourth hour, you must put the same number of boxes of the 4:00 pm into the 5:00 pm hour. So, 5:00 pm ➔ number of boxes: **790**.

It is 6:00 pm and you have a **Break** of 30 minutes to eat something.

6:00 pm ➔ number of boxes: **1030**.

7:00 pm ➔ number of boxes: **1250**.

8:00 pm ➔ number of boxes: **1560**.

Last hour boxes: **1880**. The shift has finished. Click **Stop** in the Stopwatch.

Task 6 ➔ TouchMouse

Please read the following steps completely. Use the **TouchMouse** to complete the next tasks.

This is your third day working for the company and you are in charge of entering the information in the System.

Please, click **Start** in the Stopwatch; Date is: **October 24, 2007**.

You must ask the operator about today's packaging. The operator said "this is the line **1** and shift **2**, we are working in the packaging process of the item, Reference: **374295-23**, Product: **Pain Relief 24's**, lot number: **246173C**, machine name: **Blisters**, number of units: **24**, RPM of the machine: **32**."

The manager called and said "change the number of units to **24-KAP**".

The starting time for the shift 2 is 2:00 pm.

2:00 pm ➔ number of boxes: **380**.

It is 3:30 pm and you take a **Break** of thirty minutes for coffee.

3:00 pm ➔ number of boxes: **560**.

4:00 pm ➔ number of boxes: **910**.

5:00 pm ➔ number of boxes: **1250**.

It is 6:00 pm and you have a **Break** of 30 minutes to eat something.

6:00 pm ➔ number of boxes: **1430**.

7:00 pm ➔ number of boxes: **1790**.

8:00 pm ➔ number of boxes: **2160**.

Last hour boxes: **2530**. The shift has finished. Click **Stop** in the Stopwatch.

APPENDIX G: Testing Tasks

Task 7 ➔ Mouse

Please read the following steps completely. Use the **Mouse** to complete the next tasks.

This is another day working for the company and you are in charge of entering the information in the System.

Please, click **Start** in the Stopwatch; Date is: **October 25, 2007**.

The operator said "this is the line **1** and shift **2**, we were working in the packaging process of the item: Reference: **456789-27**, Product: **Pain Relief 24's**, lot number: **456296B**, machine name: **Blisters**, number of units: **24**, RPM of the machine: **32**."

The starting time for the shift 2 is 2:00 pm.

The manager called and said "the reference number is **456781-25**".

2:00 pm ➔ number of boxes: **240**.

It is 3:00 pm and you have to leave because there is a scheduled **Meeting** for one hour and 20 minutes.

Since there was no work during the second hour, you must put the same number of boxes of the 2:00 pm into the 3:00 pm hour.

4:00 pm ➔ number of boxes: **390**.

5:00 pm ➔ number of boxes: **750**.

It is 6:00 pm and you have a **Break** of 30 minutes to eat something.

6:00 pm ➔ number of boxes: **810**.

7:00 pm ➔ number of boxes: **1130**.

8:00 pm ➔ number of boxes: **1460**.

Last hour boxes: **1720**. The shift has finished. Click **Stop** in the Stopwatch.

Task 8 ➔ Paper Form

Please read the following steps completely. Please, use the **Paper Form** to complete the next tasks.

This is another day working for the company and you are in charge of entering the information in the System.

Please, click **Start** in the Stopwatch; Date is: **October 26, 2007**.

You must ask the operator about today's packaging. The operator said "this is the line **1** and shift **2**, we were working in the packaging process of the item, Reference: **37865-2**, Product: **Pain Relief 24's**, lot number: **216374F**, machine name: **Blisters**, number of units: **24**, RPM of the machine: **32**."

Remember to calculate Produced Quantity (PQ)

$$PQ = NumberOfBoxes * units$$

Remember to calculate Useful Time (UT) for every hour. $UT = \frac{PQ}{RPQ}$

The Reference Produced Quantity (RPQ) for 24 units is **8910**.

The starting time for the shift 2 is 2:00 pm.

2:00 pm ➔ number of boxes: **290**.

It is 3:30 pm and you take a **Break** of thirty minutes for coffee.

3:00 pm ➔ number of boxes: **470**.

4:00 pm ➔ number of boxes: **780**.

It is 5:00 pm and you have a workshop called **Business and Finance Strategy** schedule for two hours and 30 minutes. This is a Meeting and training event.

Since there was no work during the forth and fifth hour, you must put the same number of boxes of the 4:00 pm into the 5:00 pm and 6:00 pm hour.

7:00 pm ➔ number of boxes: **980**.

8:00 pm ➔ number of boxes: **1320**.

Last hour boxes: **1640**. The shift has finished. Click **Stop** in the Stopwatch.

Task 9 ➔ TouchMouse

Please read the following steps completely. Please, use the **TouchMouse** to complete the next tasks.

This is another day working for the company and you are in charge of entering the information in the System.

Please, click **Start** in the Stopwatch; Date is: **October 27, 2007**.

You must ask the operator about today's packaging. The operator said "this is the line **1** and shift **2**, we were working in the packaging process of the item, Reference: **45748-9**, product: **Pain Relief 24's**, lot number: **213647A**, machine name: **Blisters**, number of units: **24-KAP**, RPM of the machine: **35**."

The starting time for the shift 2 is 2:00 pm.

2:00 pm ➔ number of boxes: **370**.

It is 3:30 pm and you have a **Meeting** of thirty minutes.

3:00 pm ➔ number of boxes: **580**.

4:00 pm ➔ number of boxes: **930**.

5:00 pm ➔ number of boxes: **1290**.

It is 6:00 pm and you have a one hour **Meeting** with the manager.

You noticed that the RPM of the machine is not set to 35; it has been 32 during the entire shift. You must change the RPM to **32**.

6:00 pm ➔ number of boxes: **1290**.

7:00 pm ➔ number of boxes: **1530**.

8:00 pm ➔ number of boxes: **1890**.

Last hour boxes: **2140**. The shift has finished. Click **Stop** in the Stopwatch.