Project Number: DCB-9906

# A Curious Browser: Implicit Ratings

A Major Qualifying Project

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor Science

By

Phong Le

Makoto Waseda

Date May 2, 2000

Approved:

Professor David C. Brown, Advisor

Professor Mark Claypool, Co-Advisor

# Abstract

This project studied the correlation between implicit ratings and the explicit rating for a single Web page. A browser was developed to record the user actions (implicit ratings) and the explicit rating of a page. Using the data collected by the browser, the individual implicit ratings and the combined implicit ratings were analyzed and compared with the explicit rating. We found that time spent on a page and scrolling time had a strong correlation with the explicit rating.

## Acknowledgement

We would like to thank our advisors, Professor David Brown and Professor Mark Claypool for their great advise and support throughout the project.  We would also like to thank to Professor Isabel Cruz for encouraging her students to participate in our user experiment.  Finally, we would like to thank all the students who have participated in the user experiment.

# Table of Contents

# 1. Introduction

## 1.1 Introduction

The Internet plays an essential role in media and information retrieval. The massive variety of information available in the Internet is cluttered and highly disorganized. Searching for a topic on the Web results in multiple pages being retrieved, but not all the pages are useful to the user. This disorder led to the development of Filtering/ Recommendation systems. This technology automatically monitors information sources to find documents for a particular information need (Lam and Yu 1999).

Currently there are many pieces of research utilizing explicit ratings to perform filtering and recommendation. With explicit rating systems, users are asked to respond to questionnaires or to evaluate particular objects based on a given scale. For instance, in the current Presidential Race, many media companies conduct polling to estimate the percentage of support for each candidate. However, there are some limitations to this approach, including users' participation level, the cost of structuring the evaluation process, and the cost of analysis.

Participation level plays an important role in explicit rating, especially with the Internet. Users must be willing to accurately provide their opinions for surveys in which those conducting the survey purposely categorize questions and pre-decide degrees of ratings. To evaluate, the attendees have to follow these pre-selected objects; therefore they may lose interest in giving accurate responses.

The cost for explicit rating is usually high. To gain valuable results, many steps are necessary. One needs to research and define subjects and objects of interest to the rating target. One must also gain the attention of the users since it will take their time and energy to complete the rating process. Finally, analytical steps are required. If a logical or technical error occurs, it is hard to ask for additional users' support later.

A different method for user analysis is to use implicit rating, where the user does not need to explicitly rate the pages being viewed. The central difference of implicit rating is that it obtains user actions, or user patterns of behavior, to predict their level of interest/approval. With implicit Web page rating, the actions or interest indicators may be such things as the time spent on the page, the movement of the mouse, or the bookmarking of the page's URL. These suggested indicators will be discussed in the solution section. Our project's methodology to investigate implicit rating is to develop an Internet browsing system that will capture user's interactions while he/she is navigating the Web.

## 1.2 Problem

Explicit ratings are common to daily activities. We can see it on many media such as surveys for movies, or products (e.g., voting for the ten best songs on MTV channel). People use this technique to obtain user opinions for marketing purposes, or general recommendations. The central feature of explicit ratings is that the evaluator has to examine the item and assign it a value on the rating scale (Nichols, 1997). It not only requires time for researching and preparing questions but also time for evaluating and selecting their favorable choices.

Explicit rating systems also pose other problems: use of appropriate scales, motivation and incentives for evaluators (Avery and Zeckhauser, 1997), avoiding the free-riding problem, achieving a critical mass of users (Oard and Marchionini, 1996), etc. Each survey probably has a particular rating scale, and it is hard to specify the depth of the scale. This can confuse users: for example, what's difference between "less" and "moderate"? How much weight is given to 8 or 9 in a 1 to 10 scale? In any survey, the more people participate, the more accuracy the ratings achieves. So, how to obtain an adequate quantity of attendees is a major issue. To avoid these aspects of explicit rating systems, the alternative, implicit rating, is used.

Implicit ratings help reduce the users' cost of time for examining and rating items. This method, instead of requesting user's analytical response, will focus on the user's actions, or behaviors to predict their level of acceptance. In implicit Web page rating, the actions or interest indicators may be time spent on the page, the mouse activities, or bookmark of the page's URL. These possible indicators will be analyzed in detailed in the section, "Approach", (Chapter 3).

Implicit rating has many advantages over explicit ratings. Time is the obvious and most beneficial advantage. Users do not have to spend time to examine a question and understand the standardized scales. They will not be interrupted to respond to the questions in a popup window. Users will not have to notice and follow any instructions.

In our project, users can actively and freely interact with the browser interface. They implicitly show their interests by their actions. However, implicit rating may be less accurate than explicit rating. Implicit rating relies solely on the actions of the user and not the intentions of the user. Thus, there will be some logical issues such as unintended actions, computer knowledge of users, and presentation of the page, that we will have to take into consideration during analysis.

Unintended actions can be a wrong mouse click or misstyping, or if the user accidentally hit a “bookmark” button instead of “stop” button. These will initialize an unwanted event. The user’s computer experience is a major factor for implicit rating. Novice users are usually confused by complex interfaces, links, animations, and plugged-in applets. If the page presents an interactive layout or an embedded applet, it will take time for novice user to adapt to these ‘fancy features’. Some commercial pages often contain marketing tools like popup advertisements, or query windows. It may affect the user’s time and attention.

This suggests that simple implicit ratings that rely on a single measure of user behavior may not be totally reliable. Therefore implicit rating should require a combination of many indicated actions to achieve greater accuracy. For instance, the time spent on the page alone is not enough to be an indicator or the user’s intentions because the user may open the browser, minimize/freeze the application to perform other tasks, or leave the desk for a moment. We need to detect not only the time, but also the user’s mouse movement to be more convinced of the user’s active and purposeful time spent with the page. Moreover, the combination of both rating systems can be best to verify the accuracy of the users’ opinions (explicit) with their actions (implicit).

In this project, we concentrate in developing a rating system for Web pages using the implicit rating approach. We develop a Web browsing system that supports some of the same basic functionality as current commercial browsers. Our Web browser system captures:

1. The time spent on a page.
2. The time spent reading a page.
3. The time spent moving the mouse.
4. The number of mouse clicks.
5. The time spent scrolling by the mouse.
6. The time spent scrolling by the keyboard.

These variables are stored for analysis. These variables are systematically combined into more meaningful values such as the combined time spent on a page and the mouse activities discussed above.

Moreover, the browser also provides tools to collect explicit rating of the page. We compare these explicit ratings with implicit interest indicators to evaluate the accuracy of the implicit rating method.

# 2. Related Work

## 2.1 Introduction

Since Filtering/Recommendation systems are a relatively new area of research, many problems in the area are not fully understood. However, there has been a lot of work related to the project. In this section, we present existing work on implicit ratings and explicit ratings.

## 2.2 Use of Time

Time is often used when gathering implicit ratings. GroupLens (1997) investigates the correlation between the time that a user spends reading Usenet article and the explicit rating that the user gives it. Their initial studies show that predictions based on time spent reading are nearly as accurate as predictions based on explicit numerical ratings. However, they do not provide any statistical correlation between them.

Morita and Shinoda (1994) found that the relationship between the time that people spend reading Usenet news and to explicit rating that they give holds true without regard for the length of the article. But the experiment used a very controlled situation. One variable control was that they requested the subject of this experiment not to do other things such as leaving the terminal to get a cup of coffee or read newly arrived e-mail messages during the experiment.

## 2.3 Use of Mouse Activity

Goecks and Shavlik (1999) researched the correlation between mouse activity on a page and the explicit rating of the page. They developed an agent that gathered information about the mouse activity of a user over a finite period of time, called the *active time.* They set the active time to be 20 minutes. On this experiment, they could not develop an agent to directly detect user's mouse activity. Instead, they considered mouse activity to be when the user moves the mouse over a hyperlink on a page or navigates through a menu. They conducted an experiment to find a correlation between mouse activity and a user's interest in a page by predicting his/her future mouse activity with the information about his/her actual mouse activity. Though they could not find a strong statistical correlation between the user's mouse activity and the user's future mouse activity on a page, they concluded that the data-collection error due to the difficulty of detecting mouse activity might be the error observed in this experiment.

## 2.4 Use of Scroll

Goecks and Shavlik (1999) also researched the correlation between scroll activity on a page and the explicit rating on the page. Similar to mouse activity, they could not develop an agent to directly detect the user's scroll activity. Instead, they considered it to be a scroll activity when *command-state change* occurred. This occurs when the user resizes the browser window, utilizes the Edit menu, or scrolls. However, they assumed that the vast majority of command-state changes are the result of user scrolling. As for the mouse activity, they conducted an experiment to find correlation between a user's scroll activity and his/her interest in a page by predicting his/her future scroll activity with the information of his/her actual scroll activity. Unfortunately, they could not find a strong statistical correlation between the user's scroll activity and his/her future scroll activity either.

## 2.5 Measurements of Explicit Rating

There is currently much discussion of measurements of Explicit Rating. International Telecommunication Union (ITU) standardized and recommended the methods for measuring the subjective quality of audiovisual communications. The 5-point quality scale is recommended for assessing the video quality, the audio quality and the overall audiovisual quality. However, Watson and Sasse (1998) criticized the recommended scales in terms of the vocabulary of the scale labels. It is impossible to find a vocabulary for the scale labels that is suitable for everyone. For example, one person feels that the distance between 'Good' and 'Fair' is equal to the distance between 'Poor' and 'Bad', but others may not feel the same way.

# 3. Solution

In this chapter, we discuss an approach to the problem discussed on chapter 2: the way to estimate a user's interest in a Web page.

## 3.1 Approach

We look at all possible implicit indicators: Mouse, Scroll, the Number of Visits, Bookmarks, Printing, Sending e-mail, and Copying, and decide which indicators are suitable to our project.

### 3.1.1 User actions

There are several user actions that might possibly express his/her interest while s/he is browsing the Internet. At the start of the project, each possible user action was discussed to consider its applicability for the project, but they were quickly reduced to just a few.

In the next five sections, the time that the user spent moving the mouse, the amount of page scrolling, the number of visits to a page, adding a bookmark for a page, printing, sending e-mail, and copying a page are discussed to see whether they are fit to include in the project.

### 3.1.2 Time

As we discussed in section 3.1.1, "Use of Time," the time that a user spends reading a Web page has been examined by some researchers. However, we hypothesize that the time that a user spends reading a page can be combined with the time of user mouse activity, or scrolling activity in order to predict his/her interest in a page more accurately. In addition, no research has tried to find the correlation between the time that user spends moving mouse and his/her interest in a page. Thus, the topic of the duration of mouse activity is interesting to us.

### 3.1.3 Scroll

The user scrolls the page down when s/he wants to read the whole page if it does not fit into one window. The user scrolls up a page if user wants to read it again, or wants to click a link at the top of the page in order to leave the page. Little investigation of user scrolling activity has been conducted. But it seems to us that there should be some statistical correlation between user scrolling activity and user explicit rating. Thus, the topic of scrolling is also interesting to us.

### 3.1.4 Number of Visits

Users visit a page more often if the page is more interesting. It can be combined with the time that a user spends reading on a page in order to obtain the average time that a user spends on a page. Thus it is also suitable to the project.

### 3.1.5 Bookmark

Users usually bookmark Web pages when they like the Web pages. Some users, however, bookmark Web pages when they don't have time to read the whole Web page so that they could read it later. In addition, users, especially students, add a course Web page to the bookmarks for easy access even if they were not interested in these topics.

However, in terms of applicability for the project, bookmarking is not a good way to estimate user's interest since the user doesn't bookmark Web pages often.

### 3.1.6 Printing, Sending e-mail, and Copying

When a user has interest in a page, he/she may print it, send it to his/her friend as an e-mail attachment, or copy the page and paste it to a text editor. However, because these activities occur rarely, they may not be suitable to study in this project.

## 3.2 System Design

Web browsers are used to browse the Internet. For this project, we will modify a Web browser to gather information about user actions in order to obtain implicit ratings. In this section, the advantages and disadvantages of each browser (Internet Explorer, Netscape, Mozilla, VB Browser, JoZilla, and building a Browser from scratch) for our project were discussed.

### 3.2.1 Browsers

Internet Explorer 5.0 (hereafter, referred to as IE) was developed by Microsoft Corporation. IE is one of the most popular browsers in the world so that most users are familiar with using IE. IE renders HTML perfectly. There are, however, some disadvantages of IE. Since Microsoft Corporation did not distribute the source code of IE, we cannot modify IE itself. Instead, we could develop a browser interface in Visual Basic or Visual C++ to interface with IE to gather information. On that way, the actual time spent moving the mouse the actual time spent scrolling could be estimated as

Goecks and Shavlik (1999) did, but they were not very accurate because on Visual Basic and Visual C++ do not have any direct method to gather the information of user actions on IE.

Netscape Communicator 4.7 (hereafter, referred to as Netscape) was developed by Netscape Communications Corporation. Netscape is also one of the most popular browsers in the world. Netscape renders HTML perfectly. However, Netscape also didn't make its source code public. Netscape's DDE (Dynamic Data Exchange) allows us to create a browser interface to interface with Netscape. But, as for IE, it could not gather all user actions because Netscape's DDE only has a method to gather information about the number of visits to a page.

Mozilla was the Web browser that derived from the source code of Netscape Navigator. Since the source code of Mozilla is free, we could inspect Mozilla and modify it. As Mozilla was based on Netscape Navigator it could render HTML perfectly. It was developed in Visual C++ on a Windows platform. However, Mozilla might have bugs and problems because it was a version of Netscape under development. It is also large and complex.

As for Mozilla, JoZilla is open source code, developed by the JoZilla developer community. Thus, we could modify it to gather information about user actions. JoZilla was written in 100% Pure Java so that it could run on any platform if Java 2 was installed. However, JoZilla cannot render HTML perfectly. It cannot render either tables or frames.

VB Browser, written in Visual Basic, is an application, in which the IE 5.0 HTML rendering engine was embedded. Thus, VB Browser can not only render HTML perfectly but can also catch information about more user actions, such as mouse movement, bookmark, print, copy & paste, and the number of visits. Though VB Browser was written in Visual Basic, Visual C++ could be used to develop additions because with Visual Basic, we can create a program that includes components of Visual C++. However, we could not gather information about scrolling since we could not modify the IE 5.0 HTML rendering itself.

The last possible selection of browser is a browser made from scratch by us. This would maximize the modifiability of the browser so that all user actions could be gathered. However, the HTML rendering by the browser would be more limited than any browser examined above because of the limit of our programming skills and time to spend on the project.

We decided to develop our browser based on VB Browser. We discuss the reason for this choice in the section 3.2.4, “Decision Making”.

### 3.2.2 Development Environments

The development environments chosen would have several effects on the project. In this section, we discuss the advantages and disadvantages of each development environment, Visual Basic 6.0, Visual C++ 6.0, and the Java 2™ Software Development Kit.

Visual Basic 6.0 (hereafter, referred to as VB) was developed by Microsoft Corporation. VB, one of the world’s most popular programming languages, would boost the speed of programmer to create high-performance applications and components in a RAD (Rapid Application Development) environment. VB would be suitable to create a program with a Graphical User Interface on a Windows platform. VB also provides some controls to interface with IE. However, we could not create a program for a non-Windows platform since VB is a development tool for the Windows platform only.

Visual C++ 6.0 (hereafter referred to as VC++) was also developed by Microsoft Corporation. VC++ is one of the most productive C++ tools to develop high-performance applications for the Windows platform. VC++ provides MFC (Microsoft Foundation Control), perhaps the most robust, productive, and widely used application framework available for Windows. VC++ also provides COM (Component Object Model), a software architecture that allows applications to be built from binary software components. Using COM technology, we could develop a program based on ActiveX technology that provided controls for Web browsers. However, as for VB, the platform for the software developed on VC++ was limited because VC++ can only create an application for the Windows platform, not Unix.

The last development environment option was the Java 2™ Software Development Kit (hereafter, referred to as Java SDK). Java SDK was developed by Sun Microsystems, Inc., and it is available free. Because of the feature of Java, "Write Once, Run Anywhere", the software developed on Java SDK is platform independent. Java SDK is officially available from Sun for the Windows platform and most Unix platforms. However, Java SDK might not be fit to create a Windows applications that needed deep control over another Windows application because Java SDK did not aim at only the Windows platform, unlike VB and VC++.

We chose Visual Basic to be our development environment. We discuss the reason for this choice in the section 3.2.4, "Decision Making".

### 3.2.3 Data Collection

The browsing system not only has to provide the friendly and functional GUI browser, but also has to allow us to efficiently and properly obtain all the indicated actions from the user. The program will temporally store implicit indicators in memory while the user is still at each page. All the data will be sent to a memory immediately after the user leaves the page. The data to be recorded and stored is:

- URL: the page's address.
- Time: when the page is visited and when the page is changed. We will measure time in milliseconds.
- Mouse activities:
  - Time that a user spends moving a mouse.
  - The number of clicks that a user clicks inside a browser window.
- Scroll activities:
  - Time that a user spends moving a scroll bar.
  - The number of times that a user moves a scroll bar.
- Keyboard activities:
  - Time that a user spends holding down a key.
  - The number of times that a user holds down a key.

To obtain an explicit rating, we will create a query window, which will receive rating input from the user for each of the Web pages. The query window has a 5-point quality scale with simple labels, "Most" and "Least", in order to avoid the problem of the suitability of the vocabulary of the scale labels. All of these ratings and the URL of the page will be saved for future comparison and evaluation.

We will use a plain-text file in order to store the data since this is the easiest way to store a file on a hard disk.

### 3.2.4 Decision Making

Given the strengths and weaknesses of the browsers discussed above, we decided to choose being between VB Browser and JoZilla. There were two main reasons for these choices. One was the availability of detecting mouse movement with the browsers. For this project, detecting the time that the user spent reading a Web page and moving the mouse had the highest priority among user actions. With IE we could not directly detect mouse movement. Instead, using the method that detected that the user moved the mouse over a hyperlink or navigated through a menu, we could indirectly detect mouse movement. But that was not adequate for our project. Even worse, for Netscape, there was no way to detect mouse movement. Therefore, we could not directly use either IE or Netscape for the project.

The other reason for eliminating some choices was the difficulty in programming. Neither Mozilla, which has a large source code and is difficult to modify, nor a browser from scratch, on which we had to spend most of our time in creating a HTML rendering engine, were good choices for the project. That was because the object of our project was to find whether there was correlation between user's interest and user actions, not to learn a programming language through creating or modifying a browser.

In order to decide the browser that we would use for our project, we made a table whose rows were the criteria, including user actions, browser performance, and implementation. As we discussed on the Section 3.1.1, "User Actions", three user actions were chosen as the indicators of user interests. They were: the time that the user spends moving the mouse (hereafter, referred to as Time of mouse activity), Scrolling, and the number of visits to the page (hereafter referred to as Visits).

Each user action was weighted with the importance and interest for the project. Each user action was scaled from 1 to 3 (3 was the most important and most interesting). Because we had more interest in Time of mouse activity than Scroll and Visits, and we also thought that Time of mouse activity had more importance for the project than the other two user actions, we weighted Time of mouse activity with a 3. And we weighted Scroll and Visits with a 2 because we thought that these two user actions have equal importance and interest for the project. HTML accuracy was scaled from 1 to 5. The reason why HTML accuracy had been more highly weighted than any user action was that HTML accuracy has a big effect on a user's ability to surf the Internet. If HTML accuracy is not good, because the HTML is rendered very badly, the Web page may become less interesting to the user than the actual Web page really was. Finally we scaled difficulty to implement the product from 1 to 3 (1 is difficult) because if we cannot finish developing the browser, we cannot collect any data. However, finishing the implementation also depends on how much we concentrate our effort on it. Thus, we weighted the same amount on the difficulty of implementation as on user actions. The table that we created is presented below.

| Criteria\Browser | Weight | VB Browser | | Jozilla | |
|---|---|---|---|---|---|
| | | Availability | Points | Availability | Points |
| **User's action:** | | | | | |
| Time of mouse activity | 3 | Yes (*1) | 3 | Yes (*1) | 3 |
| Scroll | 2 | No (*0) | 0 | Yes (*1) | 2 |
| Visit | 2 | Yes (*1) | 2 | Yes (*1) | 2 |
| **Browser Performance:** | | | | | |
| HTML accuracy | 5 | Perfect (*1) | 5 | Partly (*.4) | 2 |
| **Implementation:**<br>Difficulty to implement | 3 | Easy (*1) | 3 | Difficult (*.33) | 1 |
| **Total Point** | | | 13 | | 10 |

**Table 3.1: Comparison between two browsers, VB Browser and JoZilla**

Total Points could be calculated by the sum of the multiplication of each weight of a user action or Browser performance and points. JoZilla got 0.4 point on HTML accuracy since it could not render either tables or frames. As you would see on Table 3.1, VB Browser gained higher points than JoZilla.

The reason that VB got 3 points and JoZilla got 1 point on difficulty to implement was that it was easier for us to implement VB Browser than JoZilla. We were more familiar with VB than with Java SDK. And it would be easier to develop software with a Graphical User Interface on the Windows platform with VB rather than with Java SDK. Thus, we would be able to shorten the implementation phase and do more testing and analysis. Therefore, we decided to use VB Browser for our project.

## 3.3 Task

### 3.3.1 Implementation

During the first six weeks of C term, we designed the GUIs for the Browser and coded the functions for capturing the preferred indicators (the time of mouse activity, and the number of visits). Simultaneously, we designed the explicit rating query mechanism. We created a special button on the browser's menu bar. Using this button, the user performs the rating for the page with a scaling from one to five as least to most interesting respectively. To make sure that the user does the rating, the Browser pops up a message window to remind the user to perform the rating when s/he leaves the current page.

In the fifth and sixth week of C term, we tested the Browser's functionality and made small GUI modifications. We tested the accuracy of the times recorded and location of the mouse. We compared the recorded data with the explicit rating results. Finally, we set up installations of the Browser in designated labs for our experiments.

### 3.3.2 Design of Experiments

We set up the schedule, environment, and attendees. In the seventh week of C term, we ran the beta test and examined test data. We officially carried out the experiment in the beginning of D term for two weeks. We gained permission from the CS department to install the application in the PC lab. Our browser program was installed on thirty-eight PCs in the ADP lab and in the WINE Lab. Attendees were WPI students. We sent e-mail to introduce the project to the students and asked for their participation in our experiments.

We estimated that 60 users would participate in our experiments per day from these 38 computers and they would spend 30 minutes browsing the Web with our Curious Browser. We planed to conduct our experiments for one week. Thus, our estimation indicated that we expected approximately 420 experiments with our Curious Browser (60 users * 7 days).

### 3.3.3 Data Analysis

During the first two weeks of D term, while conducting the User Experiment, we studied and finalized the methodology for the data analysis, which began after collecting data in D term. We finished all analysis by the fourth week of D term. We used a statistic software tool, Minitab to carry out the task. We investigated the interrelationships among data and examine the accuracy of single and combined variables interest indicators.

We compared the implicit data with the explicit rating data to determine the accuracy of these values. We also investigated the relationship between certain variable values and the explicit results. By analyzing the values with the explicit results, we can approximate an implicit variable value (i.e., number of mouse clicks) for an explicit value (i.e., the usefulness of the Web page for the user). Beside time spent, we also observed the most used indicators to suggest future research.

# 4. Program Design

In this chapter, we discuss the design of our browser that records the implicit ratings and the explicit rating. We also discuss the algorithms for gathering each implicit rating and the explicit rating.

## 4.1 Overall Architecture

We design a Browser called the “Curious Browser” in order to accomplish our project. Figure 4.1 summarizes the main functions and interactions within the Curious Browser.

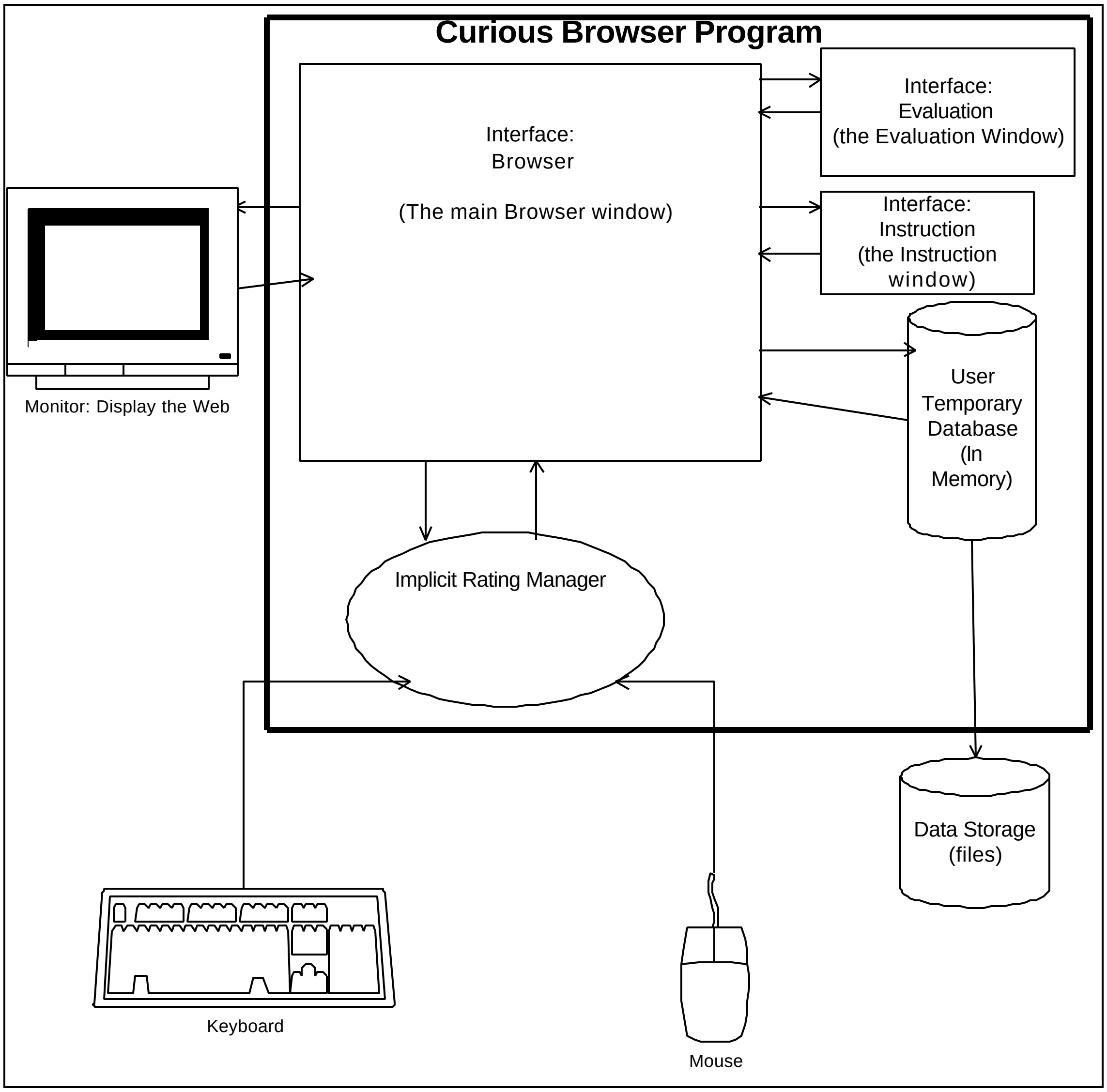



**Figure 4.1: The Overview of the Architecture of the Curious Browser. Each Interface represents the application window in the Curious Browser.**

Curious Browser is broken into three parts: its Graphical User Interface, the main Implicit Rating Manager, and the Database. The GUI of Curious Browser is what users see and interact with. The main Implicit Rating Manager captures all information from the mouse and the keyboard, decides whether they be used to form the user's implicit ratings or not, and accumulates the implicit ratings. The Database stores all user information including the user name, URL that s/he has visited, time and date when s/he visited, all the implicit ratings and the explicit ratings. These are stored in memory temporarily and in hard disk permanently. In the next section, we will discuss the HTML rendering engine and GUI of the Curious Browser. In Chapter 4.3, we will discuss the algorithms for capturing all user implicit ratings as well as user name and URL. We also discuss the Database in Chapter 4.4.

## 4.2 Graphical User Interface

### 4.2.1 Browser Interface

The main GUI of the Curious Browser is a browser itself. This GUI will greatly affect users' actions because they will browse the Web by using and seeing this browser. If this GUI is neither attractive nor functional, they will get frustrated with browsing the Web, it will make data biased, and we can not gather good data for implicit and explicit ratings. To solve these problems, we decided that the GUI of the browser should be very similar to IE 5.0. Figure 4.2-(a) below shows the GUI of the browser and Figure 4.2-(b) below shows the GUI of IE 5.0.


**Figure 4.2-(a): the GUI of the browser**


**Figure 4.2-(b): the GUI of Internet Explorer 5.0**

As you see from the figures above, the Back, Forward, Stop, Refresh, and Home buttons are identical to and in the same order as those of IE 5.0. The Address Bar, HTML rendering window, and the Status Bar are identical to those of IE 5.0 as well. The Address Bar shows the current URL address. The Status Bar at the bottom of the display shows the current status of HTML rendering engine such as a link that the mouse cursor is pointing to, or a URL of page that HTML rendering engine is downloading. The Title

Bar at the top of the display shows the current page name and it is also identical to that of IE 5.0. For example, in Figure 4.2-(a) above, the title bar shows “Worcester Polytechnic Institute (WPI) – Curious Browser”.

We added Evaluation, Instruction, and Exit buttons. We carefully chose the icons for these actions. The reason we chose the pencil icon for Evaluation button is that a pencil implies “write something down”, or “write something on the Evaluation Form”. The reason we chose a question mark for the Instruction button is that a user needs instruction when s/he has “questions” about it. The reason we chose an X mark for Exit button is that it is same as the close button on the upper-right side of an application window of Windows OS system, and because the name of the letter X sounds like the start of the word, “Exit”.

### 4.2.2 Evaluation Interface

The GUI of the Evaluation Interface is also important for our project. As discussed in Chapter 2.5, “Measurements of Explicit Rating”, a label on the scale makes the explicit rating data biased. As we discussed in Chapter 3.2.3, “Data Collection”, we chose a 5-scale explicit rating with minimum explanation, just the text, “Most” and “Least”. Figure 4.3 shows the GUI of the Evaluation Interface.

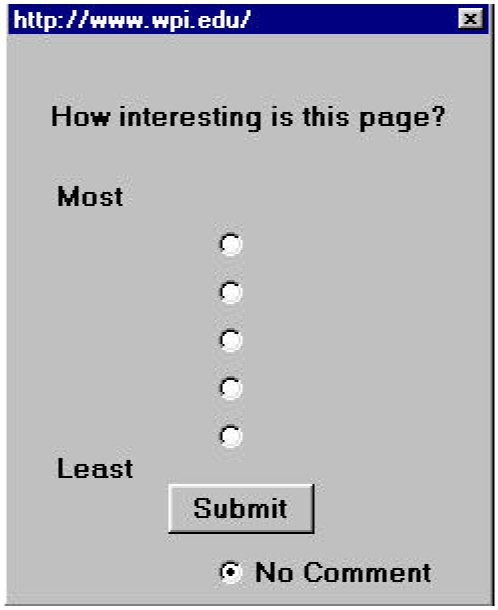



**Figure 4.3: Evaluation Interface**

As you see in the Figure above, there are 5 radio buttons for explicit rating above the "Submit" button. We added one "No Comment" button below the Submit button for a page that a user wants to evaluate later or does not want to evaluate at all. The default marked radio button is the "No Comment" button. We predict that many of the pages would be skipped. Therefore, it is convenient for users if the "No Comment" button is set as the default radio button so that they can quickly skip pages that they do not want to evaluate.

### 4.2.3 Instruction Interface

The Instruction Interface is very simple and gives basic instruction to browse the Web using Curious Browser. We also added our e-mail addresses to get feedback from users in the case of errors. Figure 4.4 shows the GUI of the Instruction Interface.


**Figure 4.4: Instruction Interface**

## 4.3 Algorithms

In this section, we discuss the algorithm that the Curious Browser uses to capture each implicit rating.

### 4.3.1 Mouse Activities

The Curious Browser captures two mouse activities. One is the number of mouse clicks. The other is the time spent moving the mouse. In order to capture correct mouse activities, the Curious Browser does not capture any mouse activities when the mouse is out of the browser window or when the browser window is not focused. "The mouse is out of the browser window" happens when the mouse cursor is out of the browser window where a user sees the HTML page, the vertical scroll bar, and the horizontal scroll bar. "The browser window is not focused" happens when a user opens another application and works on it. Figure 4.5 summarizes the state diagram of capturing mouse activities.

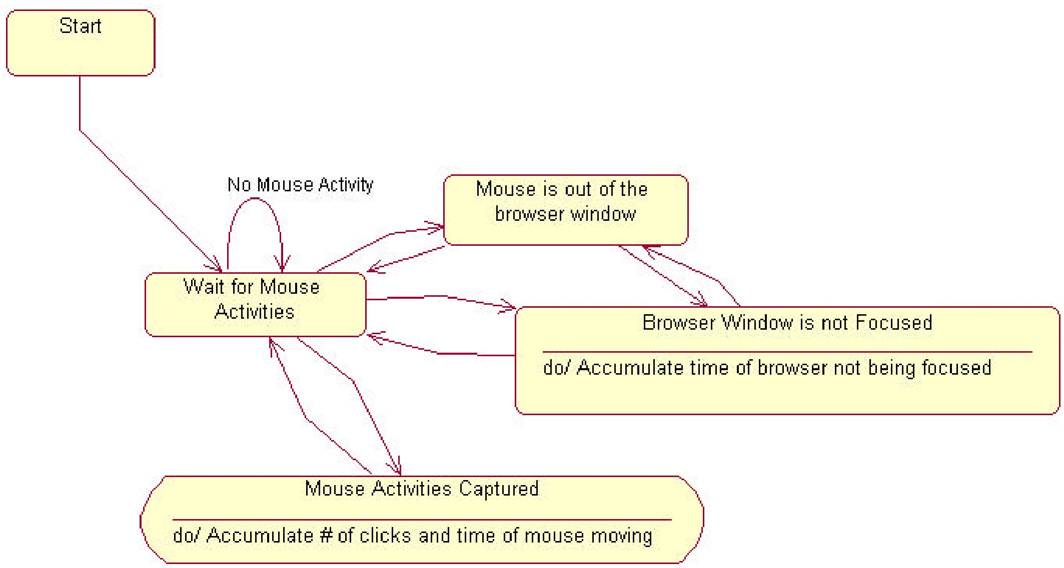



**Figure 4.5: State Diagram for capturing Mouse Activities**

The number of mouse clicks and the time spent moving the mouse are accumulated per Web page. The time spent moving the mouse is in milliseconds.

### 4.3.2 Scroll Activities

There are also two kinds of Scroll Activities that the Curious Browser will capture. One is the number of events (clicks) on the vertical and horizontal scroll bars. The other is the time spent on the vertical and horizontal scrolling. Similar to mouse activities, no scroll activities are captured by the Curious Browser when the mouse is out of the browser window or the browser window is not focused.  Figure 4.6 shows the state diagram for capturing scroll activities.

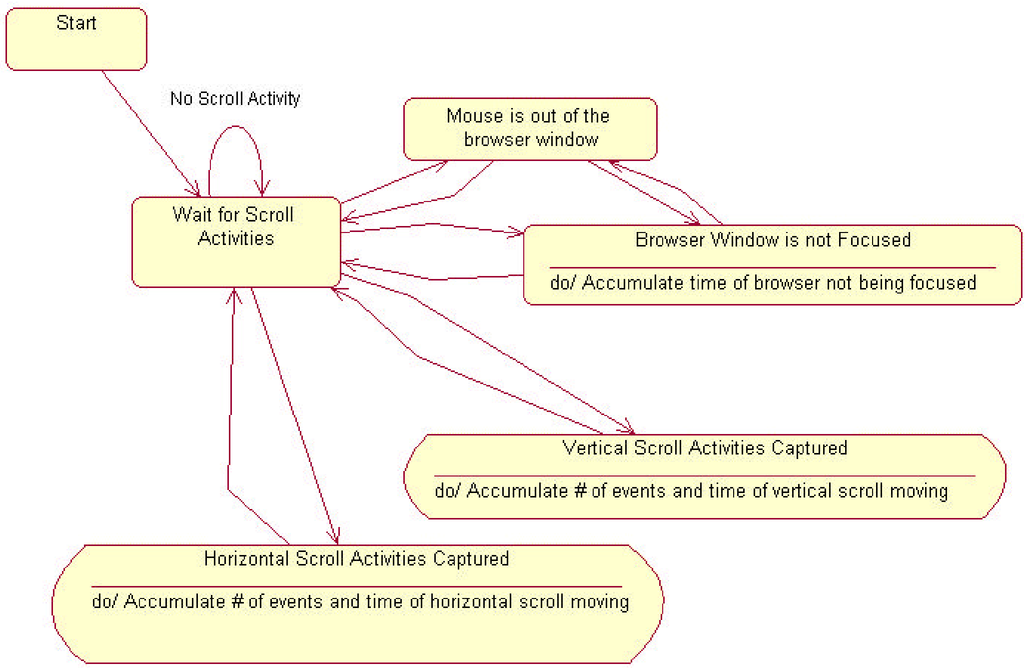



**Figure 4.6: State Diagram for capturing Scroll Activities**

The unit of time of scrolling is also in milliseconds.

### 4.3.3 Keyboard Activities

Keyboard use is the last implicit rating that the Curious Browser captures.  Some people prefer using a keyboard to using a mouse while browsing the Web.  Thus we decided to capture keyboard activities as well.  We chose 4 keys to detect and capture: Page Up, Page Down, Up Arrow and Down Arrow.  These four keys are used when a user wants to scroll a page without using a mouse.  There are two different keyboard activities. One is the number of times that a user holds down these keys. The other is the amount of time that these keys were held down. The unit of time is milliseconds. We store these data separately for each key. Figure 4.7 below shows how to capture each key activity.

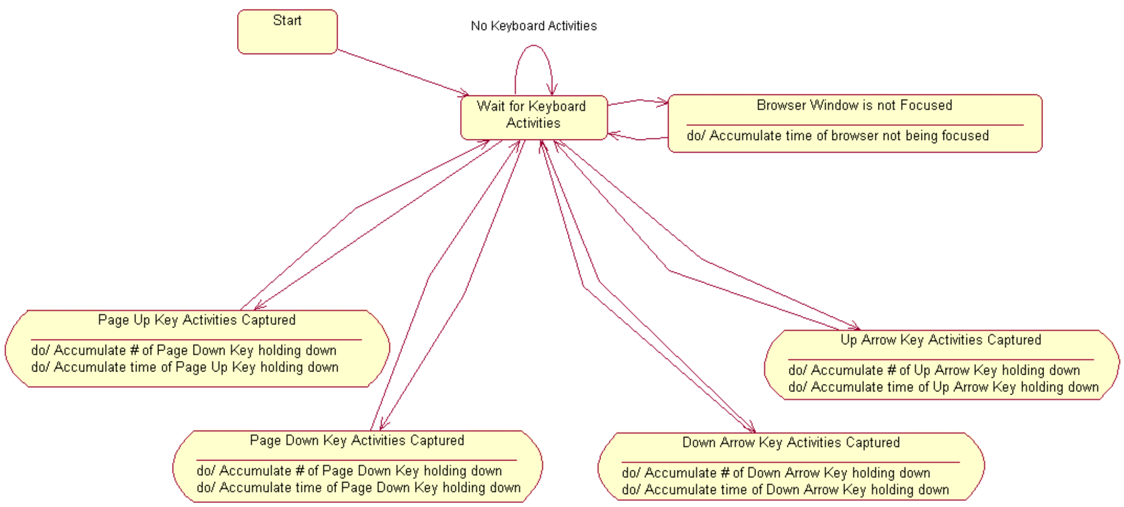



**Figure 4.7: Diagram for capturing Keyboard Activities**

The algorithm for capturing keyboard activities is similar to those of mouse and scroll activities. The difference is that there is no “The mouse is out of the browser window” state. This is because we can scroll using these keys even if mouse cursor is out of the browser window. However, we still need to have a “The browser is not focused” state.

### 4.3.4 Rating

There are two main situations in which the Curious Browser runs this rating algorithm. One is changing a page to another page. The other is on pushing the “Evaluation” button. There are also several ways to change a page to another: click a link, push the Back button, push the Forward button, or write a URL address directly into the Address Bar and hit the Enter key.

Before moving to the next page, all information including a user name, URL that s/he has visited, time and date when s/he has visited, and all implicit ratings are stored in the temporary user database (in Memory). After that, the Curious Browser checks if the user has evaluated this page before. If so, the Curious Browser does not pop up the Evaluation window and moves directly to the next page. If not, the Curious Browser pops up the Evaluation window and asks the user to evaluate the page. If the user submits her/his explicit rating for the page, the Curious Browser stores it in the temporary user database. Then, the Curious Browser moves to the next page. When the Curious Browser has finished loading the page, all variables of implicit ratings, as well as the time when s/he has visited the page, are reset to zero. Figure 4.8 shows the algorithm for getting an explicit rating.

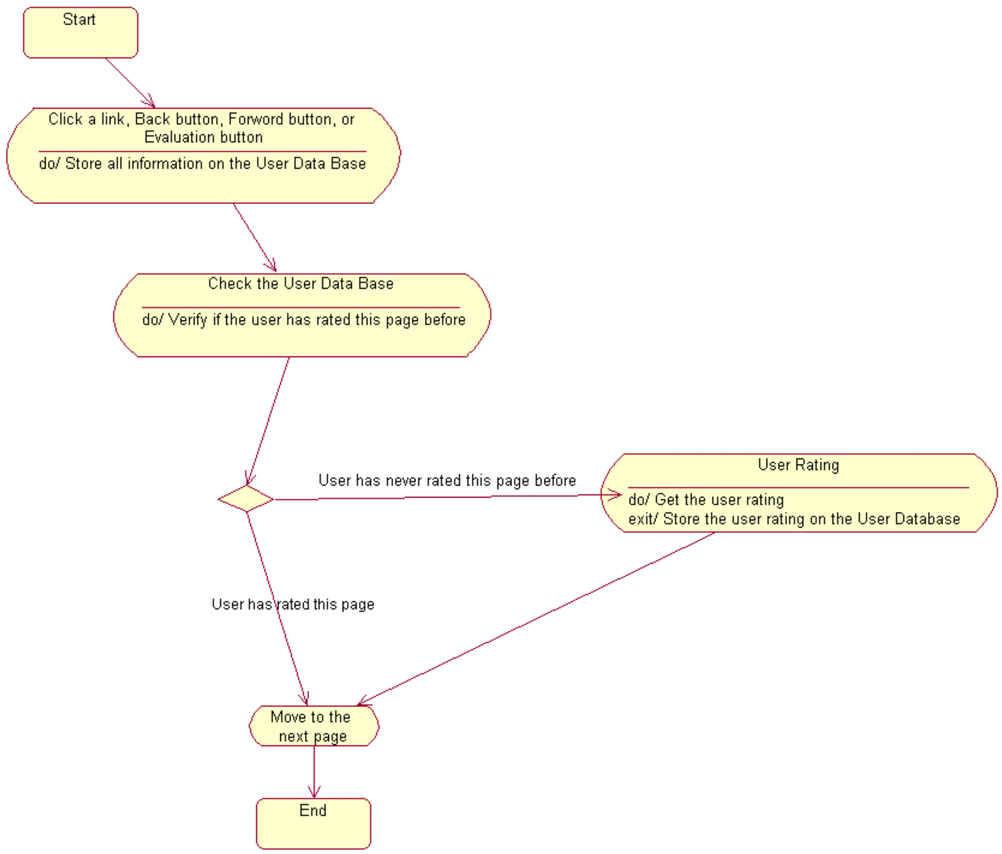



**Figure 4.8 State diagram for getting an explicit rating**

### 4.3.5 URL, User name, Time/Date Stamp

The algorithm for capturing the URL of a page and time when s/he has visited a page is not complicated. Each time a page is loaded, the current URL and the time (hours: minutes: seconds) are captured by the Curious Browser and stored in the temporary database. The Curious Browser runs these algorithms to capture the user name and the date of browser use when s/he starts to execute the Curious Browser. Figure 4.9 shows how to capture the user name.

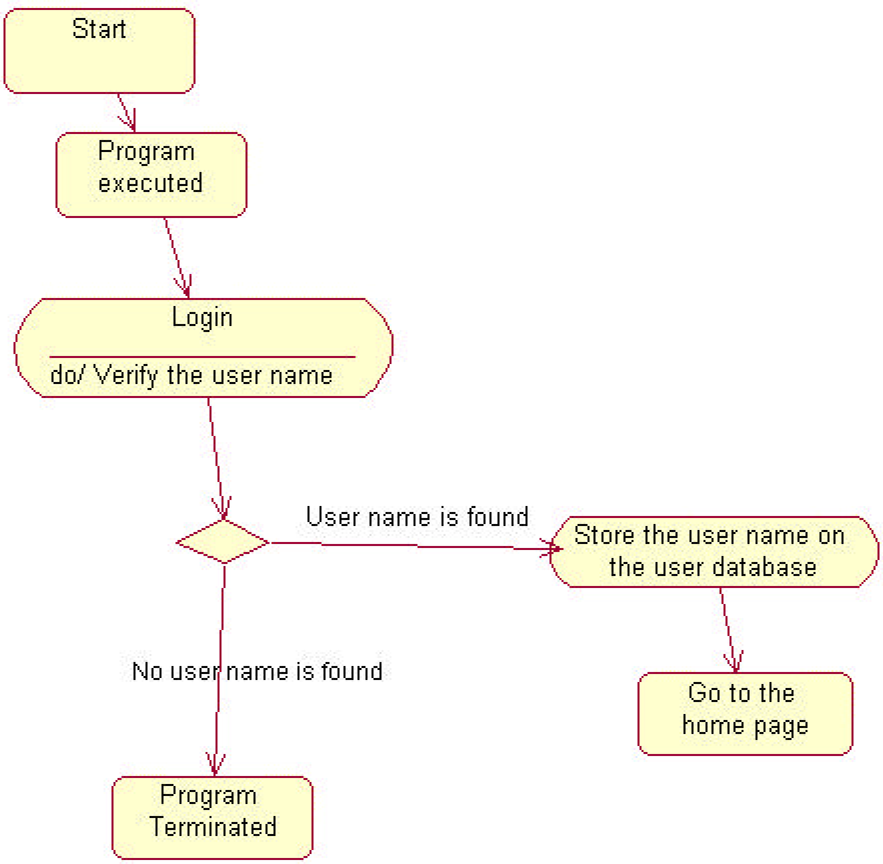



**Figure 4.9: Diagram for capturing the user name**

As you see on the figure above, the program will be terminated if no user name has been found. This happens when a user uses the Curious Browser without logging in to the Microsoft Network or to Novell Netware. The Curious Browser checks the user name in Microsoft Network or Novel Netware. In WINE lab where we install the Curious Browser, all computers have login system to the Microsoft Network, but users do not have to log in to the Microsoft Network in order to use the computer. They can use the computer by just hitting the “escape” button. Therefore, the Curious Browser needs to be closed if no user name is found in Microsoft Network or Novel Netware. As soon as the user name is captured, the Curious Browser also captures the date.

## 4.4 Database and Storage

### 4.4.1 Temporary Database in Memory

The temporary database will be used while the Curious Browser is executed. All information including the user name, URL that s/he has visited, time and the date when s/he has visited, all the implicit ratings, and the explicit rating, will be stored in the temporary database. Each data is distinguished by URL. Thus, if a user visits a page more than once, the Curious Browser will accumulate all implicit ratings and replace them over the data of the same URL. The temporary database is based on an array. Each variable (URL, time when s/he has visited a page, each implicit rating, and the explicit rating) is an array that holds 100 variables of each type. We conducted a small user experiment to see how many pages a user could visit for 30 minutes. From this experiment, we found that the biggest number of pages that the user could visit is approximately 50 pages. Therefore, we determined an array to hold 100 variables in case a user browses the Web longer than 30 minutes. Table 4.1 shows the type of variables that we use in the Curious Browser.

| Variable Name | Implicit/Explicit/Other | Type | Unit | Comment |
|---|---|---|---|---|
| URLArray | Other | String | NA | URL |
| TimeVisitArray | Other | String | NA | Clock Time that user visits |
| TimeOnPageArray | Implicit | Long | milliseconds | Time that a user spends on a page |
| TimeOnHScrollArray | Implicit | Long | milliseconds | Time that a user spends moving a horizontal scroll bar |
| TimeOnVScrollArray | Implicit | Long | milliseconds | Time that a user spends moving a vertical scroll |
| EventOnScrollArray | Implicit | Long | number | The number of events |
| TimeOnMouseArray | Implicit | Long | milliseconds | Time that a user spends moving a mouse |
| ClicksOnWindowArray | Implicit | Long | number | The number of clicks that a user clicks inside the browser window |
| NumOfUpArrowArray | Implicit | Long | number | The number of times that a user hits "Up Arrow" key |
| NumOfDownArrowArray | Implicit | Long | number | The number of times that a user hits "Down Arrow" key |
| MSecForUpArrowKey | Implicit | Long | milliseconds | Time that a user spends holding a "Up Arrow" key |
| MSecForDownArrowArray | Implicit | Long | milliseconds | Time that a user spends holding a "Down Arrow" key |
| MSecForPageUpArray | Implicit | Long | milliseconds | Time that a user spends holding a "Page Up" key |
| NumOfPageUpArray | Implicit | Long | number | The number of times that a user hits "Page Up" key |
| MSecForPageDownArray | Implicit | Long | milliseconds | Time that a user spends holding a "Page Down" key |
| NumOfPageDownArray | Implicit | Long | number | The number of times that a user hits "Page Down" key |
| RatingArray | Explicit | Integer | number | The user's explicit rating on a page |

**Table 4.1: Variables in the Curious Browser**

We have 2 String variables, 14 Long variables, and 1 Integer variable in the Curious Browser. Both String variables are declared as variable-length String. Since each character consumes 1 byte, each String variable needs storage of 10 bytes plus its string length. One Long variable needs 4 bytes. One Integer variable needs 2 bytes. In addition, 24 extra bytes are required in each variable since all variables are declared as a 1-dimension array of each type, which can contain 100 pieces of data. Therefore, in the worst case when a user visits 100 pages and the length of all URLs are 256 characters long, the variables of the Curious Browser require:

(100 variable-length String data that contain 256 characters) * (1 String Array) +

(100 variable-length String data that contains 11 characters) * (1 String Array) +

(100 Long data) * (14 Long Arrays) + (100 Integer data) * (1 Integer Array)

= 26624 (bytes) * 1 + 2124 (bytes) * 1 + 424 (bytes) * 14 + 224 (bytes) * 1

= 34,098(bytes)

~= 35 (KB)

Since more than 32 MB of memory is installed in all computers in WPI, they are able to handle the data from the Curious Browser.

### 4.4.2 Permanent Database in Hard Disk

When a user terminates the Curious Browser, all data that are stored in the temporary database, a user name, and date that s/he executes the Curious Browser will be written into a file as plain text. Figure 4.10 represents closing the Curious Browser.

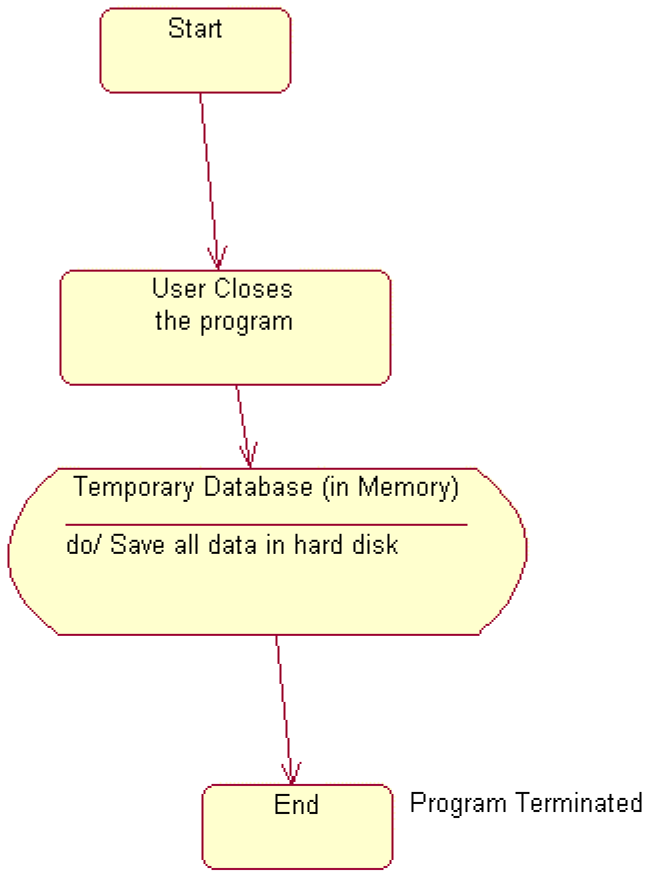



**Figure 4.10: Diagram for closing the Curious Browser**

The next table shows an example of the data format. We show the header as well though the actual data does not have the header.

| URL | User Name | Date when a user visits | Clock Time that a user visits | Time that a user spends on a page | Time that a user spends moving a horizontal scroll bar | Time that a user spends moving a vertical scroll | The number of events | Time that a user spends moving a mouse | The number of clicks that a user clicks inside the browser window |
|---|---|---|---|---|---|---|---|---|---|
| http://www.wpi.edu/ | mwaseda | 3-11-2000 | 12:38:44 PM | 20420 | 920 | 934 | 4 | 6259 | 6 |

| The number of times that a user hits "Up Arrow" key | The number of times that a user hits "Down Arrow" key | Time that a user spends holding a "Up Arrow" key | Time that a user spends holding a "Down Arrow" key | Time that a user spends holding a "Page Up" key | The number of times that a user hits "Page Up" key | Time that a user spends holding a "Page Down" key | The number of times that a user hits "Page Down" key | The user's explicit rating on a page |
|---|---|---|---|---|---|---|---|---|
| 8 | 6 | 494 | 392 | 254 | 4 | 197 | 3 | 5 |

**Table 4.2: Data Format**

Each line of a file needs 2 bytes plus the length of characters, and character needs one byte. In the worst case when the length of all URL is 256 character long and all Long variables are 10 characters long (the maximum of Long is 2,147,483,647), the file requires:

(100 data) * {(256 characters of URL) + (256 characters of user name) + (8 characters of date that user executes Curious Browser) + (11 characters of clock time that user visits a page) + (10 characters of implicit rating) * (14 implicit ratings) + (1 character of explicit rating) + (2 bytes of end of line)}

= 100 * {256 + 256 + 8 + 11 + 140 + 1 + 2}

= 67,400 (bytes)

~= 68 KB

Since all computers in WPI have approximately 40 MB of available space, they are able to handle to data requirements of the Curious Browser.

# 5. Implementation

We developed the Curious Browser in Visual Basic 5.0. Since Visual Basic is not an Object-Oriented Language but an Event-Driven Language, we could not clearly divide the whole program into objects. Therefore, in this chapter, we explain the implementation of the Curious Browser by its main activities. These are Mouse and Scroll activities, Keyboard activities, "Browser Window is not Focused" activities, "Mouse is out of the browser window" activities, and Evaluation activities.

## 5.1 Activities

### 5.1.1 Mouse and Scroll Activities

Mouse and Scroll activities are captured in the same event procedure and the same function. In order to capture Mouse Cursor movement activities and Scroll movement activities, the Timer object named "*timTimer2*" was used. The pseudocode for "*timTimer2*" is:

```
Private Sub timTimer2_Timer()
//This object is called every one millisecond.
//Do if "Browser Window is not Focused" activities are false and "The mouse is
out of the browser window" activities are false
// 1) Accumulate the time of Mouse movement if the position of a mouse cursor is
different from the last time when this object is called.  Mouse cursor should be
inside the browser window.
// 2) Accumulate the time of Horizontal Scroll movement if the position of a
mouse cursor is different from the last time when this object called.  Mouse
```

cursor should be inside either Horizontal Scroll bar and the left mouse button should be pressed down.
// 3) Accumulate the time of Vertical Scroll movement if the position of a mouse cursor is different from the last time when this object called. Mouse cursor should be inside either Vertical Scroll bar and the left mouse button should be pressed down.

In order to capture Mouse Clicks activities and Scroll event activities, the function named, "*MouseProc*" was used. *MouseProc* is a hook technique in the Window API for observing mouse portions of the Window message stream. The pseudocode of *MouseProc* is:

```
Public Function MouseProc()
//This function is called when the program captures mouse message.
//Do if "Browser Window is not Focused" activities are false
//1) Accumulate the number of Mouse Clicks if the left mouse button is pressed
    down inside the browser window without both Horizontal and Vertical Scroll
    bar.
//2) Accumulate the number of Horizontal Scroll events if the left mouse button is
pressed down inside the Horizontal Scroll bar.
//3) Accumulate the number of Vertical Scroll events if the left mouse button is
    pressed down inside the Vertical Scroll bar.
```

The figure below shows the boundary of the browser window, the horizontal scroll bar, and the vertical scroll bar.

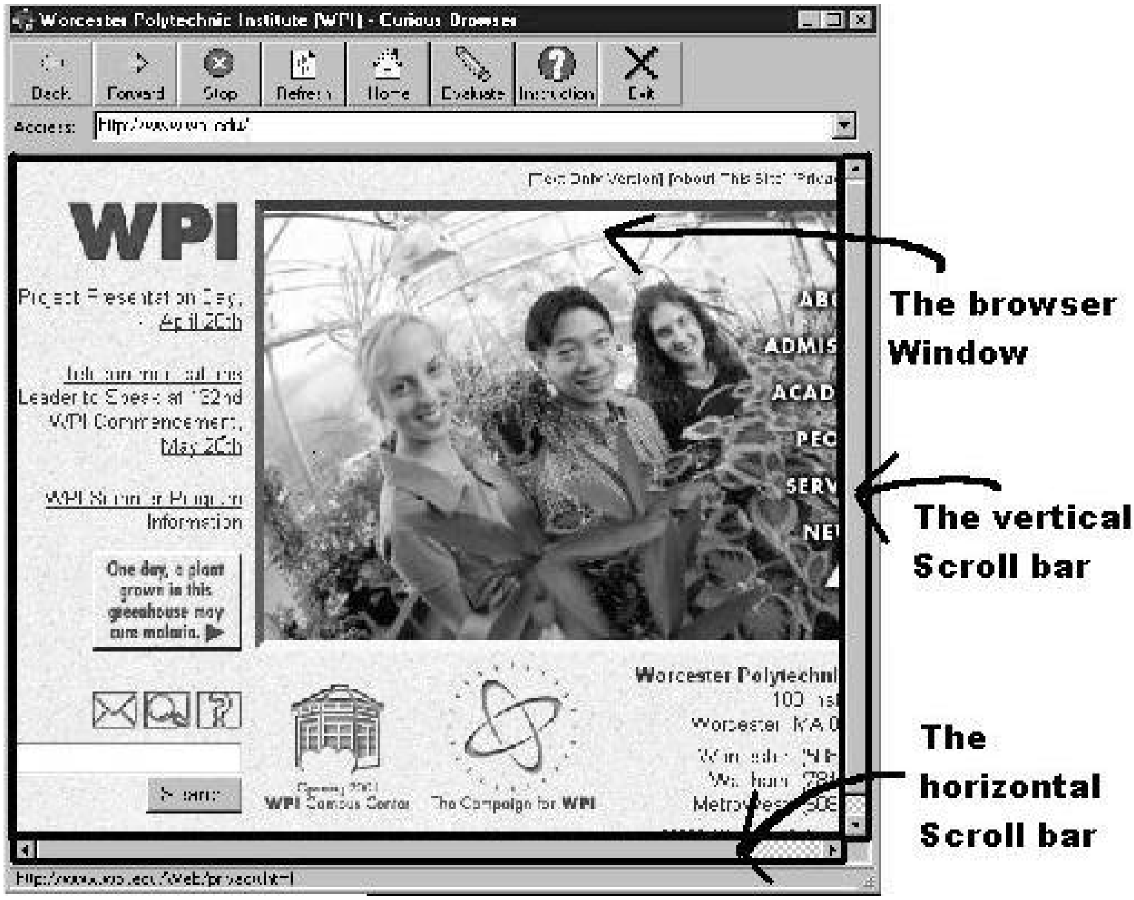



**Figure 5.1: The boundary of the browser window, the vertical scroll bar, and the horizontal scroll bar**

The browser window contains the HTML screen, the horizontal scroll bar, and the vertical scroll bar.

### 5.1.2 Keyboard Activities

Keyboard activities are captured in the function named, "*KeyboardProc*". *KeyboardProc* is a hook technique in the Window API for observing keyboard portions of the Window message stream. The pseudocode for *KeyboardProc* is:

```
Public Function KeyboardProc()
//This function is called when the program captures keyboard message
//Do if "Browser Window is not Focused" activities are false
// 1) Accumulate the time and the number of times that "Page Up" key is pressed
     down
// 2) Accumulate the time and the number of times that "Page Down" key is
pressed down
// 3) Accumulate the time and the number of times that "Up Arrow" key is pressed
     down
// 4) Accumulate the time and the number of times that "Down Arrow" key is
     pressed down
```

### 5.1.3 "Browser Window is not Focused" Activities

"Browser Window is not Focused" activities are captured using the function named "*GetActiveWindow*". *GetActiveWindow* is also a Windows API. The *GetActiveWindow* function retrieves the window handle of the active window associated with the calling thread's message queue, in our case, the Curious Browser.

### 5.1.4 "Mouse is out of the browser window" Activities

"Mouse is out of the browser window" activities are captured by setting the boundary of the browser window. Thus, if a mouse cursor moves out of the browser window, these activities are false.

### 5.1.5 Evaluation Activities

Evaluation activities are captured when a user leaves a page or pushes the Evaluation button on the Curious Browser. The function "*BeforeNavigate2*" in the object brwWebBrowser used Internet Explorer HTML rendering engine. The pseudocode for *BeforeNavigate2* is:

```
Private Sub brwWebBrowser_BeforeNavigate2()
// this function occurs when the WebBrowser control is about to navigate to a
    different URL, which may happen as a result of external automation, internal
    automation from a script, or the user clicking a link or typing in the address
    bar.  The container has an opportunity to cancel the pending navigation.
//Do if the page where Curious Browser is about be navigate has not been rated
// 1) Hide the browser window
// 2) Store all implicit ratings in the temporary database
// 3) Open the Evaluation Interface
```

## 5.2 Permanent Database

When a user terminates the Curious Browser, it will write all information into a plain-text file called "sample.dat" under the directory, "C:\Program Files\Microsoft Visual Studio". The reason why the file name is not unique is that it will be easier to find than if it contains a user name or computer name. Figure 5.2 represents the raw data generated by the Curious Browser after two users, "mwaseda" and "ple" executed and terminated it. The (End of Line) marker is added in the picture to easily see each line of the actual raw data file although there is no (End of Line) marker in the actual raw data file.

```
http://www.wpi.edu/|mwaseda|3/11/00|12:38:44 PM| 20420 | 920 | 934 | 4 | 6259 |
6 | 8 | 6 | 494 | 392 | 254 | 4 | 197 | 3 | 5 (End of Line)
http://www.cs.wpi.edu/|ple|3/19/00|1:10:26 PM| 5226 | 0 | 0 | 0 | 1124 | 2 | 0
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 (End of Line)
http://www.cs.wpi.edu/About/|ple|3/19/00|1:10:34 PM| 12269 | 0 | 1804 | 2 |
2251 | 4 | 0 | 0 | 0 | 0 | 201 | 1 | 173 | 1 | 5 (End of Line)
```

**Figure 5.2: The actual raw data format**

## 5.3 Testing

Since Curious Browser was to be installed in computers that were available for all WPI students, Curious Browser should be functional and stable, but should not crash a computer. Therefore, we performed product testing. Test cases were set up on an event-by-event or function-by-function basis, ensuring that each event or function individually satisfies its specifications. In addition, we performed stress testing making sure that Curious Browser behaves correctly when operating under a peak load such as user visits one page for the whole day.

Throughout these tests, Curious Browser has worked without incident on computers with Internet Explorer 5.0.

# 6. User Experiment

The first thing we had to do was to allocate a place and time for the user testing of our Curious Browser. We were able to get the support of the system staff for both the Wine lab and the ADP lab. We chose to install the program in the Wine lab because all of the HCI (CS3041) and Webware (CS4241) students had access to these computers. ADP also served as a good place to install our browser, because that lab is accessible by all WPI students and opened all day and night. To get the participation from more of the general population of students at WPI, we posted many flyers around key places on campus (Refer to Appendix A).

We installed our program on ten machines in the Wine lab and thirty machines in the ADP lab. Users were able to try out the program from March 20, 2000 to March 31, 2000. This is an excellent time frame because it was during the start of D-term and students had more free time and the labs were not congested. Our program was fairly easy to install and to collect data.

Once the browser is executed, the default page displays our Curious Browser instructions. This was designed so that users can easily understand that they will be prompted for a rating for every Web page. The prompt is the only time when the users have to do something extra from their usual interactions with the browser. All the interactions from the users are recorded automatically to a text file stored locally on the computers. We started collecting the data around 11:00PM every day during that week. We expected approximately 100 users and 2000 records to be collected, and thought that this was a reasonable volume of data, with which to test the indicators.

# 7. Results

We watched for any difficulty with using our Curious Browser and for bug reports from users. During the experiment we receive 5 bug reports. One was due to the browser's failure to perform "open link in new window", and the others were the errors cause by the conflict between the Browser and the Internet Explorer versions installed in the labs (the Curious Browser was implemented using the IE 5 engine). We also received a lot of feedback from users worried about how the browser recorded their involvement, we quickly responded by e-mail to those students.

Below are the general results from our experiment:

- Number of users: 84 participants.
- Number of records: 2618.
  - ♦ 2603 records without error.
  - ♦ 1822 records having the explicit rating.
  - ♦ 15 records that have errors. (This might occur because either the user or the computer abnormally terminates Curious Browser.)
  - ♦ 781 records having "no comment" rating.

Table 7.1 below shows the number of records for each implicit rating.

| | The number of records | Moving the mouse | Mouse Clicks | Scrolling by the mouse | Scrolling by the keyboard | Scrolling by the Mouse and by the keyboard |
|---|---|---|---|---|---|---|
| **All Records (without error)** | 2603 | 2024 | 1415 | 867 | 398 | 1131 |
| **Rated Records** | 1822 | 1679 | 1209 | 733 | 363 | 970 |

**Table 7.1: The number of records for each implicit rating**

Figure 7.1 below shows the number of records and the percentage of each explicit rating group. As you see, the distribution from the explicit rating, 1, to the explicit rating, 5 is bell-shaped.

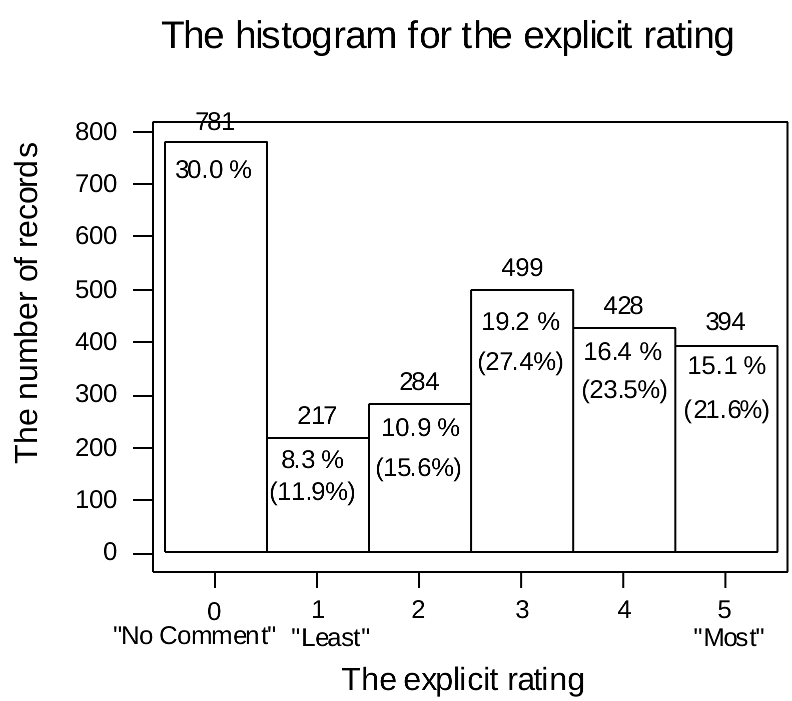



**Figure 7.1: The histogram for the explicit rating. The number above the each bar shows the number of records on each rating group. The upper percentage shows the percentage of the number of records of each rating group among the entire rating group (from 0 to 5). The lower percentage shows the percentage of the number of records of each rating group among the rating group from 1 to 5.**

# 8. Analysis

In this chapter, we discuss the methods for the statistical analysis on the data collected by the Curious Browser. We also discuss the result of our analysis. Finally, based on the analysis, we propose the combined implicit indicators.

## 8.1 Tools and Exceptional Situations

After collecting data, we began to do some statistical analysis. There were many software programs available that could perform the statistical analysis that we wanted. We chose Minitab due to its ease of use and compatibility with Excel. We examined the data with Minitab to focus on two areas: individual interest indicators vs. the explicit rating, and the combination of indicators vs. rating.

Prior to the tests on the data, we first had to collect and assemble the data into a format that Minitab could understand. Our raw data comprised of many text files. We had to copy the text data from each of files and paste them on one document. Next, we used Excel to convert from text format to a spreadsheet format appropriate for each of the indicators. Minitab takes Excel spreadsheets as the input for the analysis.

While analyzing the data with Minitab, we discovered that there were some exceptional situations. There are some outliers that have very large values comparing to the mean and median. The reasons for the outliers are that users make errors. Reasons include the unusual behaviors of the users due to loss of interest caused by the loading time of the page. This also includes situations when the user leaves the computer to do other things, or when the users' attention is diverted away from the browser. In addition, while most users tend to spend just a few minutes per page, others spend over an hour.

Users are not perfect and often hit the wrong key, or clicked on the wrong button. Events such as these qualify as outliers, and therefore were filtered out and not included with each test using Minitab. Thus, each time that we did the statistical analysis for an implicit rating or combined implicit ratings vs. the explicit rating, we filtered out the data that qualified as outliers and did not include them with the tests using Minitab. We discussed what data were filtered out on the section of each statistical analysis.

## 8.2 Methodology

We first focused on the median of the data because the median was not affected much by the outliers. We predicted that there should be differences in the values of indicators among the explicit rating groups (1 to 5). That means the greater the value of some indicators, the higher the rating. To verify the assumption, we ran the **Kruskal-Wallis** test (based on .05 level of significant). This test examines the degree of independence of the median among the explicit rating groups. We explain this test using an example:

**Example of Kruskal-Wallis test**

We have an implicit indicator called "x" and the explicit rating with 5-scale. We want to see if the median of "x" differs among the explicit rating groups at the 0.05 level of significant. We have the total of "n" records. To calculate Kruskal-Wallis test, we first assign ranks to all records. The smallest rank is 1 and the largest rank is "n". We take the average for the ties. Table 8.1 shows all records and ranks of "x" by each explicit rating.

| Raw Records | | | | | Ranks | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** | **1** | **2** | **3** | **4** | **5** |
| 17.54 | 19.75 | 21.6 | 23.74 | 23.48 | 1 | 3.5 | 10 | 17 | 16 |
| 18.74 | 20 | 21.8 | 24.1 | 25.28 | 2 | 5 | 11 | 18 | 20 |
| 19.75 | 20.4 | 22.75 | 25.1 | 26.45 | 3.5 | 7 | 13 | 19 | 23 |
| 20.23 | 20.6 | 23.4 | 25.4 | 27.73 | 6 | 8 | 14 | 21 | 24 |
| 21.54 | 22.2 | 23.5 | 26.31 | 29.48 | 9 | 12 | 15 | 22 | 25 |

**Table 8.1: The values and the ranks of the implicit rating, "x" by the explicit rating (from 1 to 5).**

With this data, we run the Kruskal-Wallis test. Table 8.2 shows the results:

## Kruskal-Wallis Test: the implicit rating, "x" vs. the explicit rating

```
Rating       N    Median    Ave Rank         Z
1            5     19.75         4.3     -2.96
2            5     20.40         7.1     -2.00
3            5     22.75        12.8     -0.07
4            5     25.10        19.4      2.17
5            5     26.45        21.4      2.85
Overall     25                  13.0

H = 20.50  DF = 4  P = 0.000
H = 20.51  DF = 4  P = 0.000 (adjusted for ties)
```

**Table 8.2: Kruskal-Wallis Test: The implicit rating, "x" vs. the explicit rating**

Null Hypothesis: The median for the implicit rating, “x” is the same for all explicit rating groups.

Hypothesis: The median for the implicit rating, “x” differs among the explicit rating groups.

The result of the Kruskal-Wallis Test includes:

- `Rating`: the explicit rating.
- `N`: the number of records for the records of each explicit rating.
- `Median`: the median for the records of the implicit rating in each explicit rating.
- `Ave Rank`: the average rank for the records of each explicit rating.
- `Z`: Z-value for the records of the implicit rating in each explicit rating. The value of Z indicates how the mean rank for the implicit rating in each explicit rating differs from the mean rank for the all records. It is calculated as:

  $$Z_j = \frac{\overline{R}_j - (N+1)/2}{\sqrt{(N+1)(N/n_j - 1)/12}}$$

  ($\overline{R}_j$ is the average of the ranks in the explicit ratig group $j$. $j$ is the explicit rating group. $N$ is the total number of records. $n_j$ is the number of records within the explicit rating group $j$.)

- `H`: the test statistic. Under the null hypothesis, the distribution of H can be approximated by a chi-square distribution with k - 1 degrees of freedom (in this case, k = 5 (5 scales of the explicit rating). The value of H. It can be referred to a table of the chi-square distribution with k-1 degrees of freedom, for a test of the hypothesis that all k population distributions are identical. Large values of H suggest that there are some differences in location among the k populations. H is calculated as:

$$H = \frac{12\sum n_i\left[\overline{R}_i - \overline{R}\right]^2}{N(N+1)}$$

($\overline{R}_i$ is the average of the ranks in the explicit ratig group $i$. $\overline{R}$ is the average of all ranks. $n_i$ is the number of records in the explicit rating group $i$. $N$ is the total number of records.

- **`H(adj)`**: the test statistics adjusted when there are ties in the data. When there are no ties, H(adj) = H. Under the null hypothesis, the distribution of H(adj) is also approximately a chi-square with k - 1 degrees of freedom. It is calculated as:

$$H(adj) = \frac{H}{1 - \left[\sum\left(d_j^{\,3} - d_j\right)/(N^3 - N)\right]}$$

($H$ is the test statistic calculated above. $N$ is the total number of records. Suppose there are $J$ distinct values among the $N$ observations and, for the $j$th distinct value, there are $d_j$ tied observations ($d_j$ = 1 if there are no ties))

- **`DF`**: the degree of freedom. In this example, the degree of freedom is 4 because it is calculated as: 5 (the number ofexplicit rating) – 1 = 4.
- **`P`**: the p-value acquired from the a table of the chi-square using the value of **`H`** or **`H(adj)`**.

Since the p-value of our example, 0.00, is smaller than the level of significant, 0.05, the null hypothesis is rejected. This test concludes that the median for the implicit rating by the explicit rating group differs among the explicit rating groups at the 95% confidence interval.

We also examined some basic descriptive statistics of the indicators by each explicit rating group. The descriptive statistics include:

1. **`N`**: the number of records for the implicit rating.
2. **`Mean`**: the average for the implicit rating.
3. **`Median`**: the median for the implicit rating.
4. **`TrMean`**: the trimmed mean for the implicit rating. The trimmed mean is the average value for the records that exclude the smallest 5% and the largest 5% values.
5. **`StDev`**: the sample standard deviation, which provides a measure of how spread out the data are.
6. **`SE Mean`**: Standard error of the mean. It is calculated as:

$$SEMean = \frac{StDev}{\sqrt{N}}$$

7. **`Minimum`**: the smallest number among the implicit rating.
8. **`Maximum`**: the largest number among the implicit rating.
9. **`Q1`**: the first quartile. It is calculated as the observation at the point, (N+1)/4.
10. **`Q2`**: the third quartile. It is calculated as the observation at the point, 3*(N+1)/4.

We analyzed the volume, minimum, maximum, and standard deviation for each indicator individually against the rating groups. By comparing the statistical figures of the Web pages with their rating groups, we could examine the trend of indicators with respect to each rating group.

Finally, we developed three charts to illustrate the trends of implicit ratings toward the explicit rating. The first chart illustrates the boxplot of the median for the implicit rating and outliers for the implicit rating against each explicit rating group. From the first chart, you will see the distribution of the records of the implicit rating as well as the median and the first and third quartile against each explicit rating group. The second chart zoomed in on the boxplots. From the second chart, you will clearly see the trend of the median, and the first and third quartile for the implicit rating against each explicit rating group. The third chart illustrates the means, the medians, outliers and 95 percent confidence intervals for the median for the implicit rating against each explicit rating group. The 95 percent confidence intervals for the median represents the probability that the data of the implicit rating will fall within that interval is 95%. These graphical presentations gave visual substantiation for our data.

## 8.3 Analysis for Each Indicator toward the Explicit Rating

### 8.3.1 Selection of the Indicators

We selected seven indicators that we collected. We did this because we chose to filter out the less effective indicators. Some of the indicators tested the same action. For example, there were indicators that recorded the number of times a key pressed and there were indicators that recorded the duration of the keys pressed. We kept the indicator that represented the duration of the key while it is pressed over the other indicator because it better represents the situation where a person is holding an arrow button to scroll down.

This would be one press, but the time would be much longer. The seven indicators we selected are:

1. The time spent on a page.
2. The time spent reading a page (we discuss how to calculate it later.).
3. The time spent moving the mouse.
4. The number of mouse clicks.
5. The time spent scrolling by the mouse.
6. The time spent scrolling by the keyboard.
7. The time spent scrolling by the mouse and the keyboard.

### 8.3.2 Statistical Analysis for the Time Spent on a Page vs. the Explicit Rating

Time spent on a page is captured right after loading the page and right before leaving the page. It includes all the actions and the reading time for the page. But it does not include the time that the Curious Browser is not on focus that means a user opens another application and works on it. Therefore, there are some factors that influence its accuracy such as loading time and unintended errors discussed above. Loading time depends on speed of connection, CPU speed and the amount of Internet traffic. Before running the test, we filtered out 91 outliers: 4 data points that have larger than 1,200,000 milliseconds (20 minutes) spent on a page, and 87 data points that have less than 1,000 milliseconds (1 second). Thus, the number of data we used to run the test was 1731.

First, we ran the Kruskal-Wallis test for the time spent on a page vs. the explicit rating. Table 8.3-(a) below shows the results:

**Kruskal-Wallis Test: The time spent on a page vs. The explicit rating**

| Rating | N | Median | Ave Rank | Z |
|---|---|---|---|---|
| 1 | 211 | 13414 | 684.4 | -5.63 |
| 2 | 272 | 18018 | 806.4 | -2.14 |
| 3 | 473 | 21217 | 880.6 | 0.75 |
| 4 | 393 | 24372 | 922.3 | 2.54 |
| 5 | 382 | 26798 | 932.8 | 2.96 |
| Overall | 1731 | | 866.0 | |

H = 43.93  DF = 4  P = 0.000

**H = 43.93  DF = 4  P = 0.000 (adjusted for ties)**

**Table 8.3-(a): Kruskal-Wallis Test: The time spent on a page vs. the explicit rating**

Null Hypothesis: The median of the time spent on a page is the same for all explicit rating groups.

Hypothesis: The median of the time spent on a page differs among the explicit rating groups.

Since the p-value is 0.00 < 0.05, the null hypothesis is rejected and the Kruskal-Wallis Test concludes that the median values for the five explicit rating groups differ.

Next, we ran the descriptive statistics. Table 8.3-(b) shows the result.

**Descriptive Statistics: The time spent on a page vs. The explicit rating**

| Rating | N | Mean | Median | TrMean | StDev |
|---|---|---|---|---|---|
| 1 | 211 | 28747 | 13414 | 20099 | 65837 |
| 2 | 272 | 37983 | 18018 | 26956 | 77988 |
| 3 | 473 | 47921 | 21217 | 34456 | 84126 |
| 4 | 393 | 50975 | 24372 | 36490 | 96646 |
| 5 | 382 | 59552 | 26798 | 41813 | 106383 |

| Rating | SE Mean | Minimum | Maximum | Q1 | Q3 |
|---|---|---|---|---|---|
| 1 | 4532 | 1026 | 835347 | 6986 | 27161 |
| 2 | 4729 | 1277 | 891319 | 9169 | 38650 |
| 3 | 3868 | 1015 | 824234 | 9738 | 50633 |
| 4 | 4875 | 1059 | 1142377 | 11310 | 49494 |
| 5 | 5443 | 1030 | 1158231 | 10620 | 59266 |

**Table 8.3-(b): Descriptive Statistics: The time spent on a page vs. the explicit rating**

We also analyzed the graphs of the time spent on a page (y-axis) vs. the explicit rating (x-axis)

Figure 8.1-(a) shows the boxplots of the median for the time spent on a page with the maximum time of 1,200,000 milliseconds (20 minutes). Figure 8.1-(b) zoomed in the boxplots so that you can see the median and the first and third quartile for the median for the time spent on a page. Figure 8.1-(c) shows the connected lines of the median and the mean for the time spent on a page as well as the 95 % confidence interval for the median in the maximum time of 60,000 milliseconds (1 minute) spent on a page.

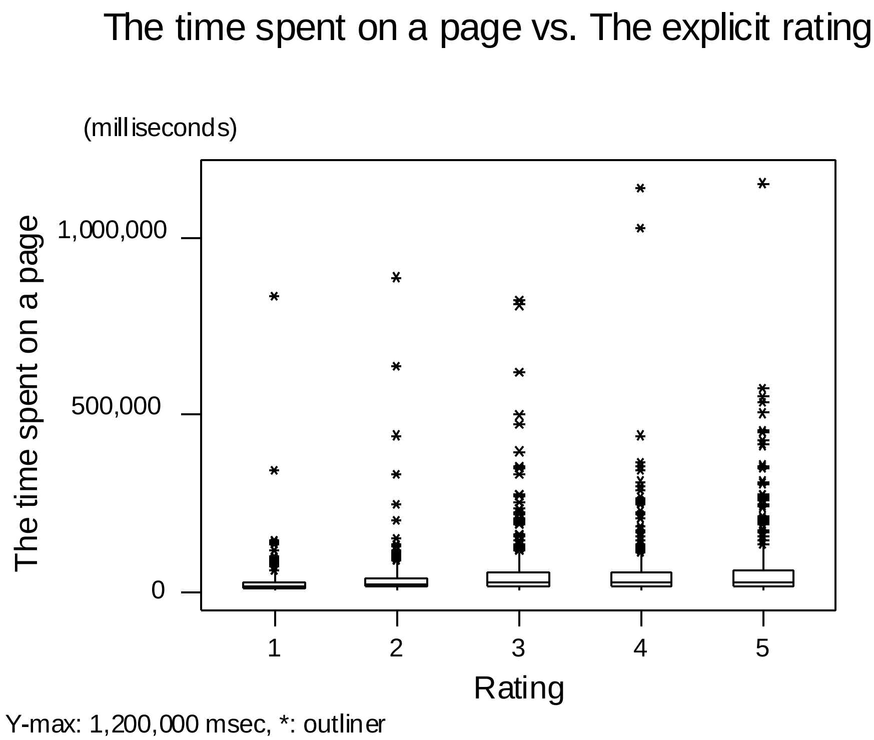



**Figure 8.1-(a): the boxplots of the time spent on a page vs. the explicit rating with the Y-maximum, 1,200,000 milliseconds (20 minutes)**

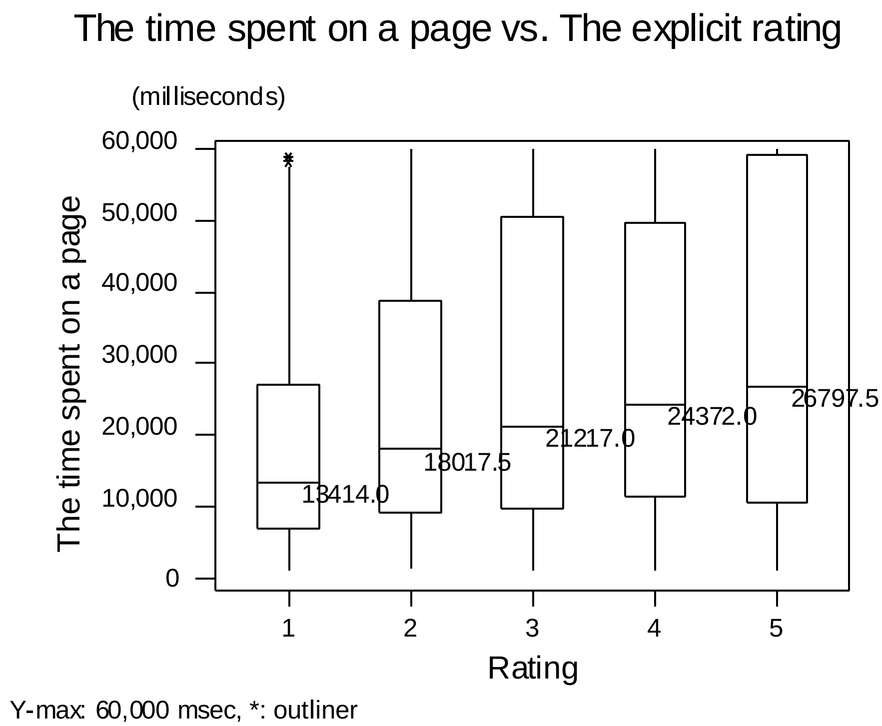



**Figure 8.1-(b): the boxplots of the time spent on a page vs. the explicit rating with the Y-maximum, 60,000 milliseconds (1 minute)**

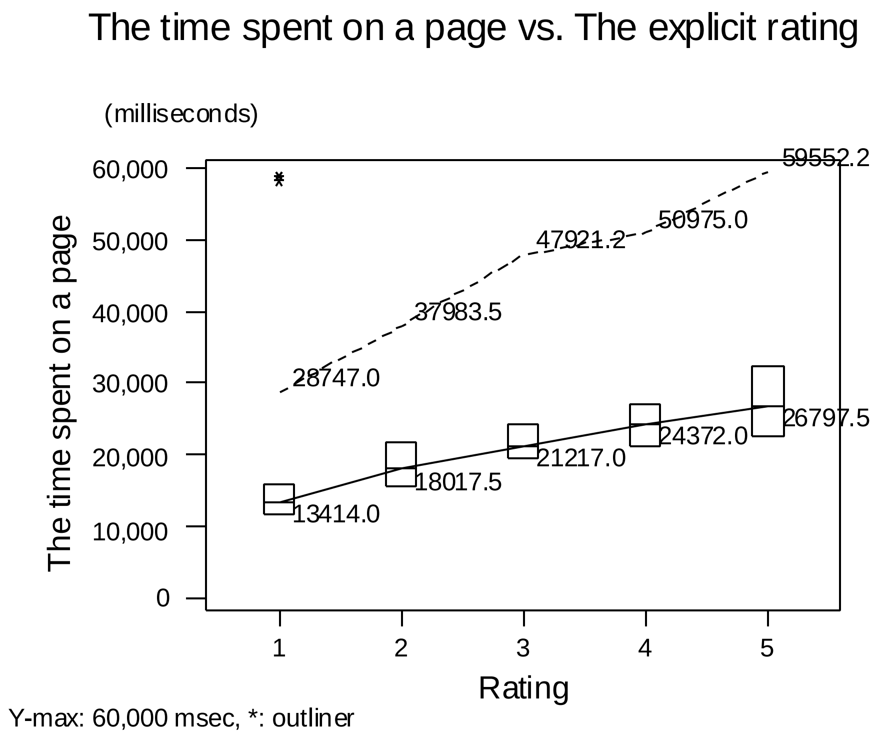



**Figure 8.1-(c): The upper dashed line shows the average time spent on a page. The lower line shows the median for the time spent on a page with the box of 95% confidence interval for the median. The maximum time is 60,000 milliseconds (1 minute).**

From the result from the descriptive statistics, we can see how the difference in medians is probably occurring. The median for those who give a rating of 1 is significantly less than the median for the other explicit rating groups (13414 vs. 18018 and higher respectively). From the descriptive statistic and Figure 8.1-(c), we can also see that there are substantial increases in volume, mean, median and the trimmed mean for the time spent on a page toward the explicit rating groups.

We conclude that *there is a positive strong correlation between the time spent on a page and the explicit rating.*

### 8.3.3 Statistical Analysis for the Time Spent Reading a Page vs. the Explicit Rating

Theoretically, the time spent reading a page is the time when the browser is active but the user is not interacting with the Web page, except reading the page. It will provide a more accurate and meaningful results than the time spent on page. Thus, we calculated the time spent reading a page by:

(The time spent reading a page) = (The time spent on a page) – {(The time spent scrolling by the mouse) + (The time spent scrolling by the keyboard)}

As explained earlier, we filtered out the total of 91 outliers: 4 outliers from the data that have the time spent reading larger than 1,200,000 milliseconds (20 minutes) and 87 outliers from the data that have the time spent reading less than 1000 milliseconds (1 second). Thus, the total number of the data that we ran the tests was 1731.

First we ran the Kruskal-Wallis test for the time spent reading a page vs. the explicit rating. Table below 8.4-(a) shows the results of the test.

**Kruskal-Wallis Test: The time spent reading a page vs. The explicit rating**

| Rating | N | Median | Ave Rank | Z |
|---|---|---|---|---|
| 1 | 211 | 11555 | 701.4 | -5.10 |
| 2 | 272 | 15080 | 805.3 | -2.18 |
| 3 | 473 | 17777 | 875.5 | 0.49 |
| 4 | 393 | 19111 | 912.8 | 2.11 |
| 5 | 382 | 20884 | 940.3 | 3.29 |
| Overall | 1731 | | 866.0 | |

H = 38.93  DF = 4  P = 0.000

**H = 38.93  DF = 4  P = 0.000 (adjusted for ties)**

**Table 8.4-(a): Kruskal-Wallis Test: The time spent reading a page vs. the explicit rating**

Null Hypothesis: The median time for the reading time is the same for all explicit rating groups.

Hypothesis: The median time for the reading time differs among the explicit rating groups.

Since the p-value is 0.00 < 0.05, the null hypothesis is rejected and the statistical conclusion is that the median values for the five explicit rating groups differ.

Next, we ran the descriptive statistics. Table 8.4-(b) shows the result.

**Descriptive Statistics: The time spent reading a page vs. The explicit rating**

| Rating | N | Mean | Median | TrMean | StDev |
|---|---|---|---|---|---|
| 1 | 211 | 24389 | 11555 | 16906 | 61045 |
| 2 | 272 | 31585 | 15080 | 21564 | 70374 |
| 3 | 473 | 41349 | 17777 | 28351 | 80195 |
| 4 | 393 | 42628 | 19111 | 29301 | 85825 |
| 5 | 382 | 51222 | 20884 | 34480 | 100030 |

| Rating | SE Mean | Minimum | Maximum | Q1 | Q3 |
|---|---|---|---|---|---|
| 1 | 4202 | 1026 | 823063 | 6706 | 22111 |
| 2 | 4267 | 1113 | 790087 | 7598 | 30984 |
| 3 | 3687 | 1015 | 804236 | 8222 | 36805 |
| 4 | 4329 | 1059 | 1014862 | 9276 | 39690 |
| 5 | 5118 | 1030 | 1158231 | 9067 | 46216 |

**Table 8.4-(b): Descriptive Statistics: The time spent reading a page vs. the explicit rating**

We also analyzed the graphs of the time spent reading a page (y-axis) vs. the explicit rating (x-axis)

Figure 8.2-(a) shows the boxplots of the median for the time spent reading a page with the maximum time of 1,200,000 milliseconds (20 minutes). Figure 8.2-(b) zoomed in the boxplots so that you can see the median and the first and third quartile for the median for the time spent reading a page. Figure 8.2-(c) shows the connected lines of the median and the mean for the time spent reading a page as well as the 95 % confidence interval for the median with the maximum time of 60,000 milliseconds (1 minute) spent reading a page. Figure 8.2-(d) shows the trend line for the median for the time spent reading a page vs. the explicit rating.

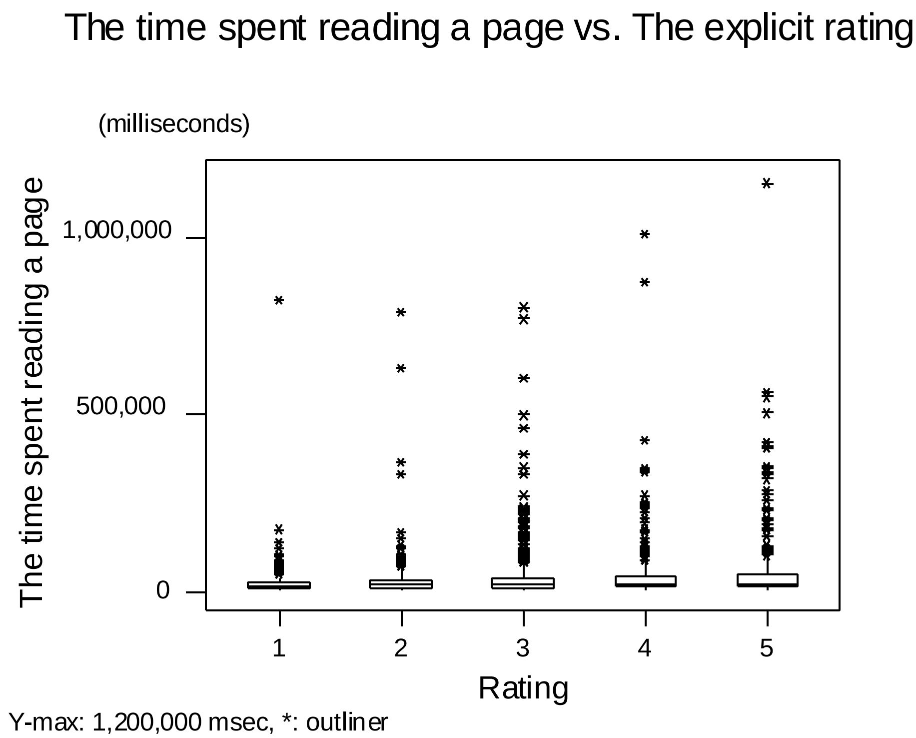



**Figure 8.2-(a): the boxplots of the time spent reading a page vs. the explicit rating with the Y-maximum, 1,200,000 milliseconds (20 minutes)**

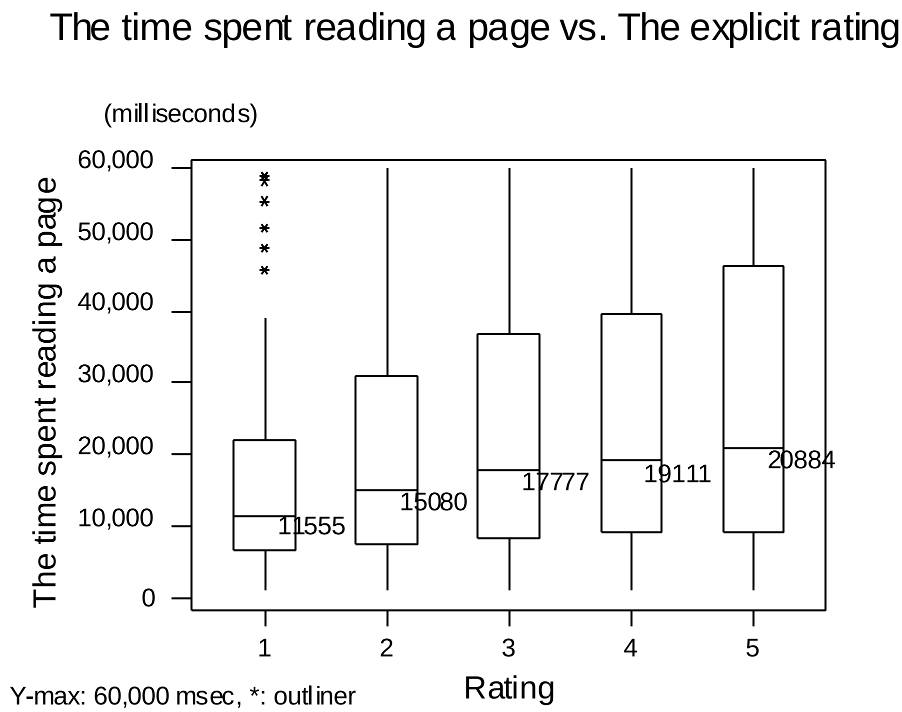



**Figure 8.2-(b): the boxplots of the time spent reading a page vs. the explicit rating with the Y-maximum, 60,000 milliseconds (1 minute)**

The time spent reading a page vs. The explicit rating

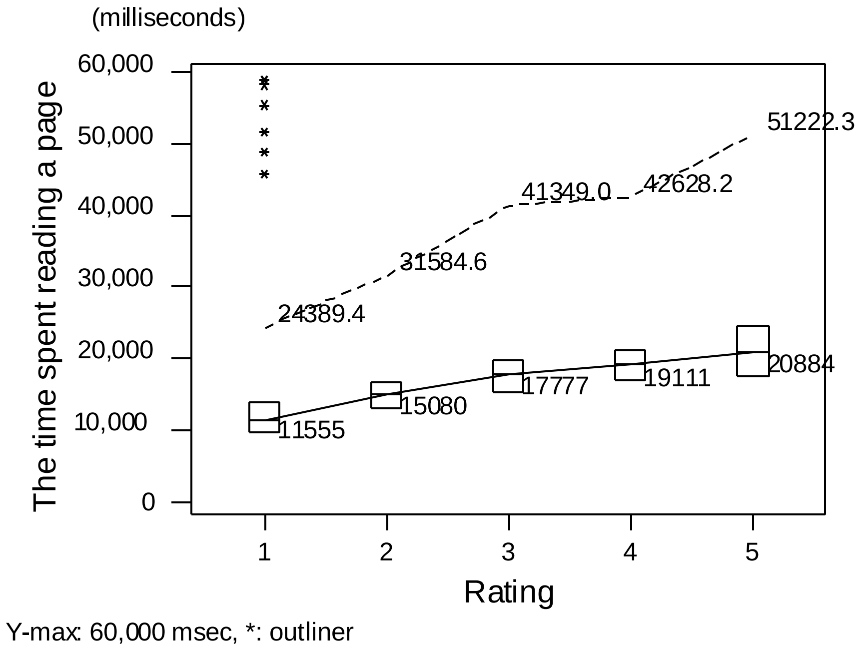



**Figure 8.2-(c): The upper dashed line shows the average time spent reading a page. The lower line shows the median for the time spent reading a page with the box of 95% confidence interval for the median. The maximum time is 60,000 milliseconds (1 minute)**

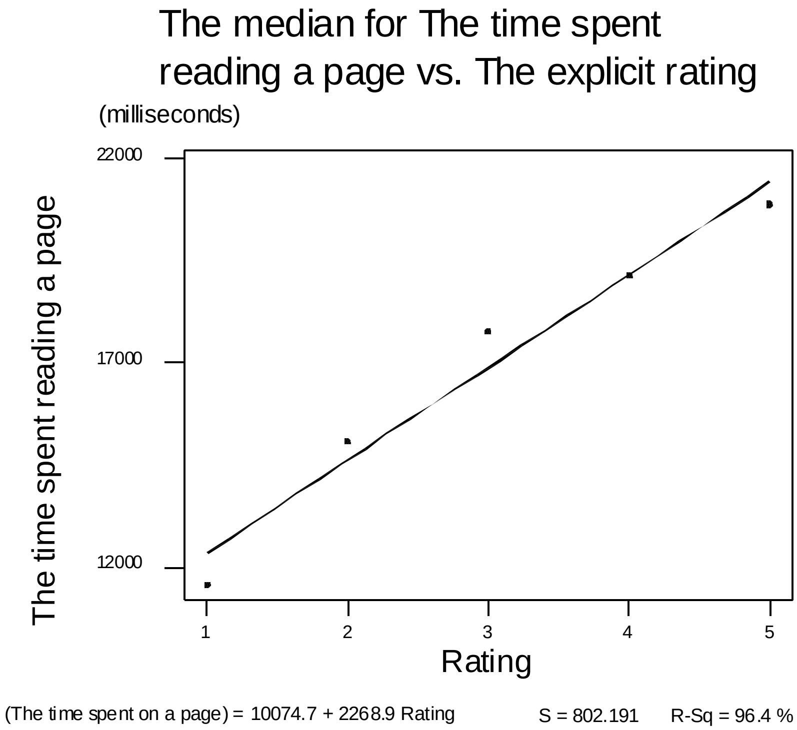



**Figure 8.2-(d): The trend line for the median for the time spent reading a page vs. the explicit rating**

**S** is the estimated standard deviation about the regression line. **R-sq** (R-Squared) is the coefficient of determination. It is calculated as R-Squared = Correlation(the explicit rating, the predicted by the median for the time spent reading a page).

Looking at the result from the descriptive statistics, we can see how the difference in medians is probably occurring. The median for those who give a rating of 1 is significantly less than the median for the other explicit rating groups (11550 vs. 15080 and higher respectively). Figure 8.2-(c) and Figure 8.2-(d) also shows that the time spent reading a page is positively proportional to the explicit rating groups. The statistical computations and the illustrated charts above strongly represent the correlation between the time spent reading a page and the explicit rating level. Especially, In Figure 8.2-(d), the trend line for the median vs. the explicit rating illustrates the strong relationship between them. We conclude that *the time spent reading a page has a strong correlation with the explicit rating*.

### 8.3.4 Statistical Analysis for the Time Spent Moving the Mouse vs. the Explicit Rating

The time spent moving the mouse will be detected if there is any mouse position change inside the active browser. Some users usually move the mouse along reading texts or looking at interest objects on the page, while others move the mouse more generally. We filtered out the total of 146 outliers: 143 outliers from the data that have 0 milliseconds spent moving the mouse, and 3 outliers from the data that have larger than 540,000 milliseconds (9 minutes) spent moving mouse. The reason that we chose the upper limit as 9 minutes is that there is the significant gap between the data larger than 6 minutes and less than 6 minutes. The maximum time was approximately 3 minutes after we filtered out the data larger than 6 minutes. The total number of data points for which we ran is 1676. First we ran the Kruskal-Wallis test for the time spent moving the mouse vs. the explicit rating. Table 8.5-(a) shows the result.

**Kruskal-Wallis Test: The time spent moving the mouse vs. The explicit rating**

| Rating | N | Median | Ave Rank | Z |
|---|---|---|---|---|
| 1 | 201 | 2750 | 680.0 | -4.95 |
| 2 | 267 | 4117 | 849.1 | 0.39 |
| 3 | 456 | 4350 | 869.8 | 1.62 |
| 4 | 383 | 4198 | 857.6 | 0.88 |
| 5 | 369 | 4286 | 858.7 | 0.91 |
| Overall | 1676 | | 838.5 | |

H = 24.85  DF = 4  P = 0.000

**H = 24.85  DF = 4  P = 0.000 (adjusted for ties)**

**Table 8.5-(a): Kruskal-Wallis Test: The time spent moving the mouse vs. the explicit rating**

Null Hypothesis: The median for the time spent moving the mouse is the same for all explicit rating groups.

Hypothesis: The median for the time spent moving the mouse differs among the explicit rating groups.

Since the p-value is 0.00 < 0.05, the null hypothesis is rejected and the statistical conclusion is that the median values for the five groups differ.

Next, we ran the descriptive statistics. Table 8.5-(b) shows the result.

**Descriptive Statistics: The time spent moving the mouse vs. The explicit rating**

| Rating | N | Mean | Median | TrMean | StDev |
|---|---|---|---|---|---|
| 1 | 201 | 5102 | 2750 | 3988 | 7010 |
| 2 | 267 | 8155 | 4117 | 5880 | 14469 |
| 3 | 456 | 7696 | 4350 | 6100 | 10296 |
| 4 | 383 | 7125 | 4198 | 5887 | 8729 |
| 5 | 369 | 8113 | 4286 | 6027 | 15239 |

| Rating | SE Mean | Minimum | Maximum | Q1 | Q3 |
|---|---|---|---|---|---|
| 1 | 494 | 109 | 51965 | 1414 | 6227 |
| 2 | 885 | 55 | 153381 | 2148 | 7923 |
| 3 | 482 | 5 | 80871 | 2328 | 9249 |
| 4 | 446 | 12 | 79283 | 2234 | 9013 |
| 5 | 793 | 18 | 184302 | 1964 | 9282 |

**Table 8.5-(b): Descriptive Statistics: The time spent moving the mouse vs. the explicit rating**

We also analyzed the graphs of the time spent moving the mouse (y-axis) vs. the explicit rating (x-axis)

Figure 8.3-(a) shows the boxplots of the median for the time spent moving the mouse with the maximum time of 200,000 milliseconds (approximately 3.3 minutes). Figure 8.3-(b) zoomed in the boxplots so that you can see the median and the first and third quartile for the median for the time spent moving the mouse. Figure 8.3-(c) shows the connected lines of the median and the mean for the time spent moving the mouse as well as the 95 % confidence interval for the median with the maximum time of 10,000 milliseconds (10 seconds) spent moving the mouse.

The time spent moving the  mouse vs. The explicit rating

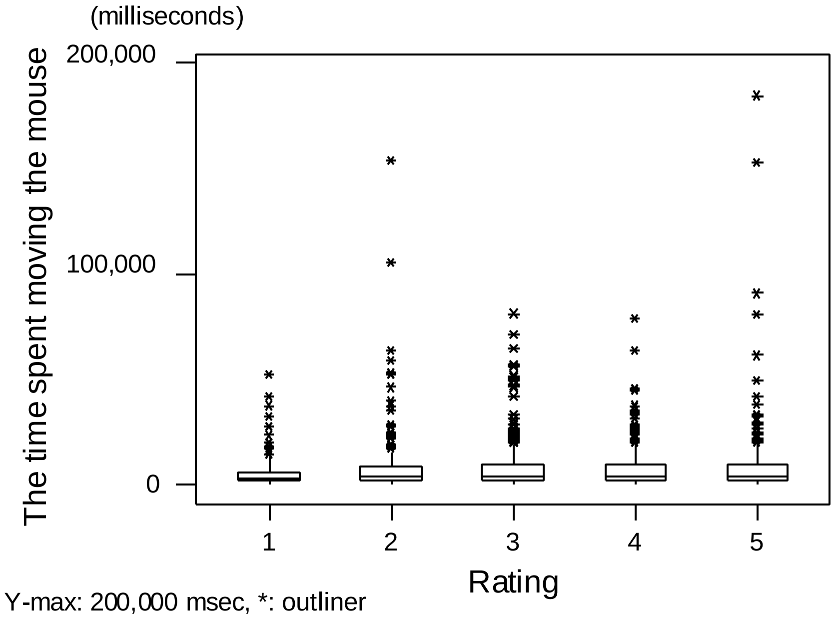



**Figure 8.3-(a): the boxplots of the time spent moving the mouse vs. the explicit rating with the Y-maximum, 200,000 milliseconds (approximately 3.33 minutes)**

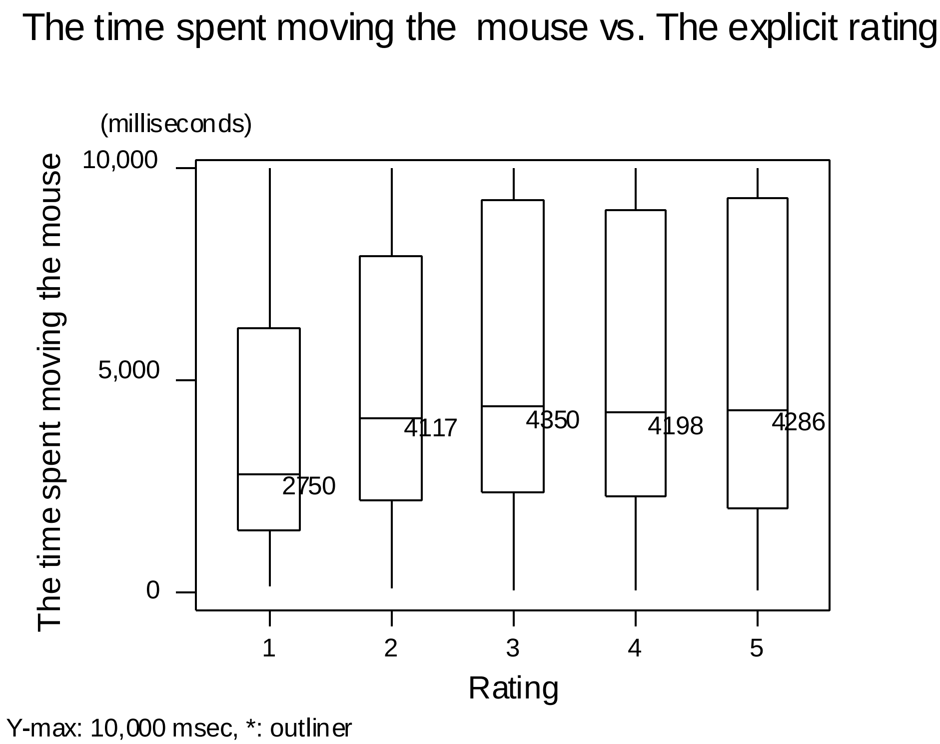



**Figure 8.3-(b): the boxplots of the time spent moving the mouse vs. the explicit rating with the Y-maximum, 10,000 milliseconds (10 seconds)**

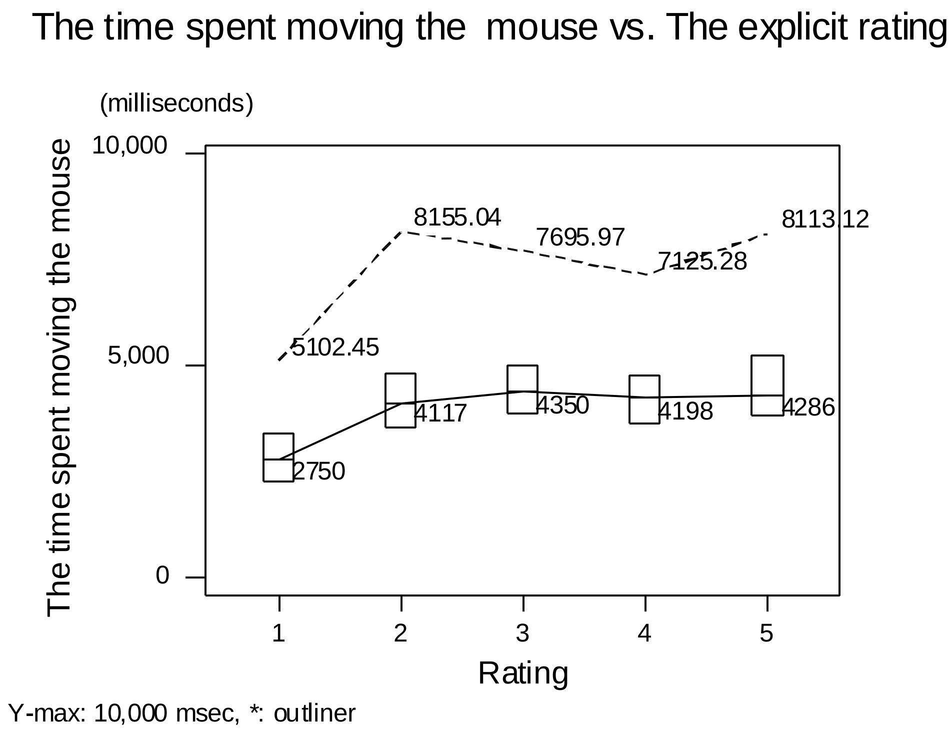



**Figure 8.3-(c): The upper dashed line shows the average time spent moving the mouse. The lower line shows the median for the time spent moving the mouse with the box of 95% confidence interval for the median. The maximum time is 10,000 milliseconds (10 seconds)**

From Figure 8.3-(c), the median for those who give a rating of 1 is significantly less than the median for the other explicit rating groups (2750 vs. 4117 and higher respectively). Compared with the explicit rating group 1, we can see there are small difference among the explicit rating group 2, 3, 4, and 5. From the descriptive statistics, we can observe that the time spent moving the mouse is positively proportional to the explicit rating. However, they are not linearly proportional as you see in the connected lines of the mean and the median in Figure 8.3-(c). The explicit rating 3 has the largest time spent moving the mouse. The time spent moving the mouse includes any changes of mouse's position inside running application. Therefore it may contain others actions such as moving the mouse to the scroll bar for navigation or moving along the text for reading, etc. Our conclusion is that *there is a positive relationship between the time spent moving the mouse and the explicit rating, but this implicit indicator is not as good as the time spent reading a page*.

### 8.3.5 Statistical Analysis for the Number of Mouse Clicks vs. the Explicit Rating

Clicking may be considered as a moderate interest indicator. Our application detects user’s action on a single page. However, there is a high chance that users (experts and novices) click to a link that will leave that page. We should assume that users sometimes clicked on wrong button or link. We filtered out the total of 617 outliers: 613 outliers from the data that have 0 mouse clicks, and 4 outliers from the data that have larger than 100 mouse clicks. Thus, the total number of data for which we ran the test was 1205. First we ran the Kruskal-Wallis test for the number of mouse clicks vs. the explicit rating. Table 8.6-(a) shows the result.

**Kruskal-Wallis Test: The number of mouse clicks vs. The explicit rating**

| Rating | N | Median | Ave Rank | Z |
|---|---|---|---|---|
| 1 | 118 | 1.000 | 546.4 | -1.86 |
| 2 | 181 | 1.000 | 582.2 | -0.87 |
| 3 | 342 | 2.000 | 615.7 | 0.80 |
| 4 | 288 | 2.000 | 614.6 | 0.65 |
| 5 | 276 | 2.000 | 613.0 | 0.54 |
| Overall | 1205 | | 603.0 | |

H = 4.78  DF = 4  P = 0.311

**H = 5.42  DF = 4  P = 0.247 (adjusted for ties)**

**Table 8.6-(a): Kruskal-Wallis Test: The number of mouse clicks vs. the explicit rating**

Null Hypothesis: The median for the number of mouse clicks is the same for all explicit rating groups.

Hypothesis: The median for the number of mouse clicks differs among the explicit rating groups.

Since the p-value is 0.247 > 0.05, the null hypothesis accepted and the Kruskal-Wallis test concludes that the median values for all explicit groups are the same.

Next, we ran the descriptive statistics. Table 8.6-(b) shows the result.

**Descriptive Statistics: The number of mouse clicks vs. The explicit rating**

| Rating | N | Mean | Median | TrMean | StDev |
|---|---|---|---|---|---|
| 1 | 118 | 2.907 | 1.000 | 1.896 | 6.291 |
| 2 | 181 | 3.448 | 1.000 | 2.313 | 6.853 |
| 3 | 342 | 3.868 | 2.000 | 2.461 | 7.794 |
| 4 | 288 | 3.358 | 2.000 | 2.358 | 6.490 |
| 5 | 276 | 3.145 | 2.000 | 2.294 | 5.858 |

| Rating | SE Mean | Minimum | Maximum | Q1 | Q3 |
|---|---|---|---|---|---|
| 1 | 0.579 | 1.000 | 59.000 | 1.000 | 3.000 |
| 2 | 0.509 | 1.000 | 57.000 | 1.000 | 3.000 |
| 3 | 0.421 | 1.000 | 84.000 | 1.000 | 3.000 |
| 4 | 0.382 | 1.000 | 73.000 | 1.000 | 3.000 |
| 5 | 0.353 | 1.000 | 80.000 | 1.000 | 3.000 |

**Table 8.6-(b): Descriptive Statistics: The number of mouse clicks vs. the explicit rating**

We also analyzed the graphs of the number of mouse clicks (y-axis) vs. the explicit rating (x-axis)

Figure 8.4-(a) shows the boxplots of the median for the number of mouse clicks with the maximum 100 mouse clicks. Figure 8.4-(b) zoomed in the boxplots so that you can see the median and the first and third quartile for the median for the number of mouse clicks. Figure 8.4-(c) shows the connected lines of the median and the mean for the number of mouse clicks as well as the 95 % confidence interval for the median with the maximum 5 mouse clicks.

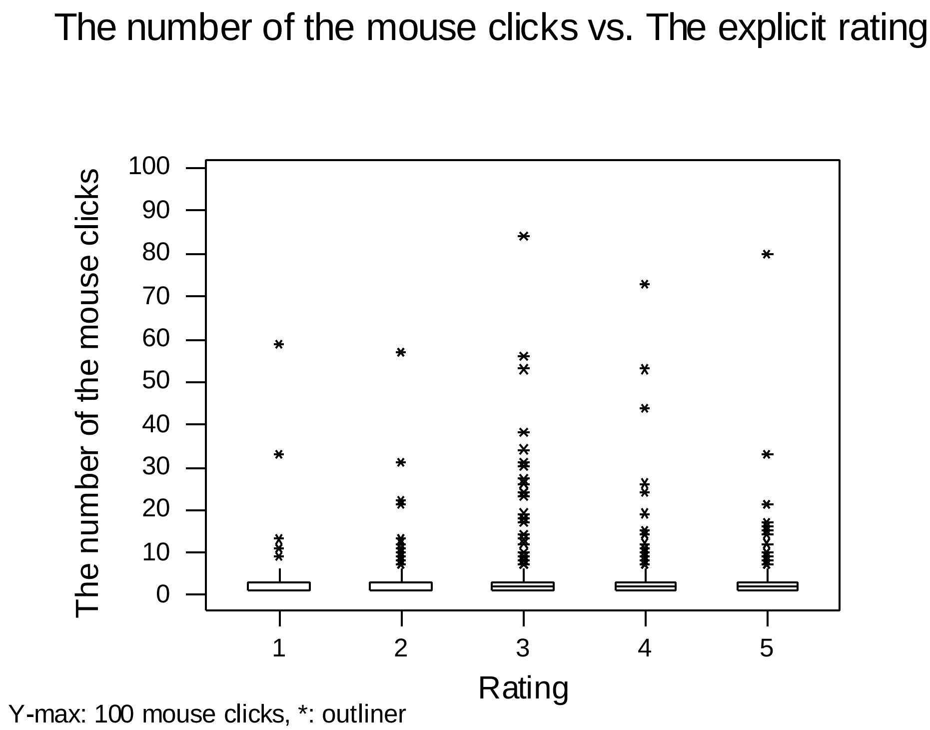



**Figure 8.4-(a): the boxplots of the number of mouse clicks vs. the explicit rating with the Y-maximum, 100 mouse clicks**

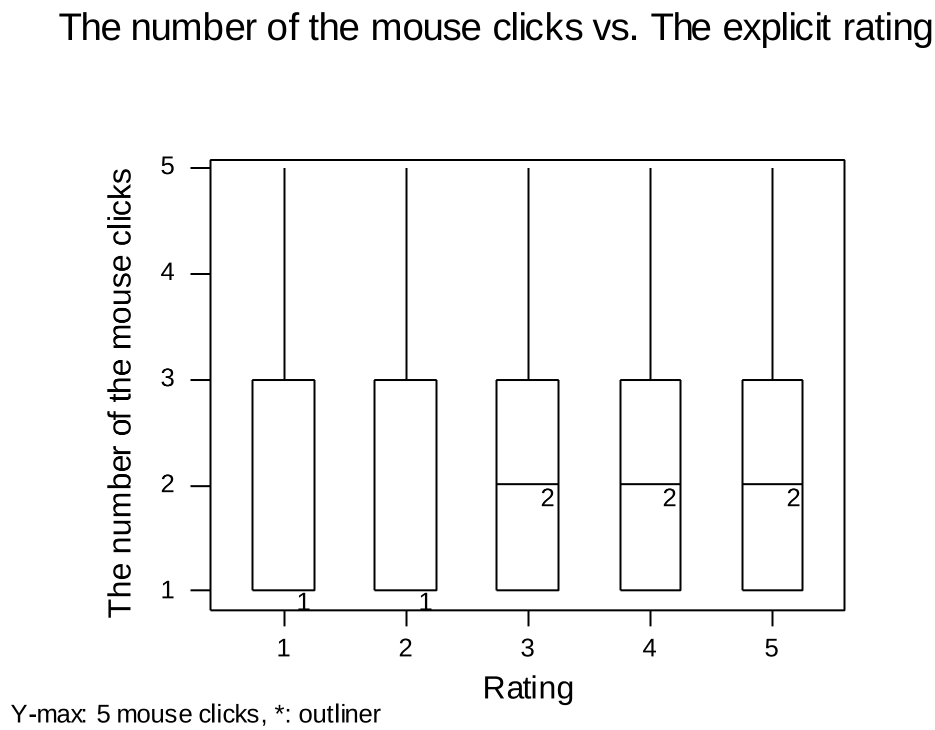



**Figure 8.4-(b): the boxplots of the number of mouse clicks vs. the explicit rating with the Y-maximum, 5 mouse clicks**

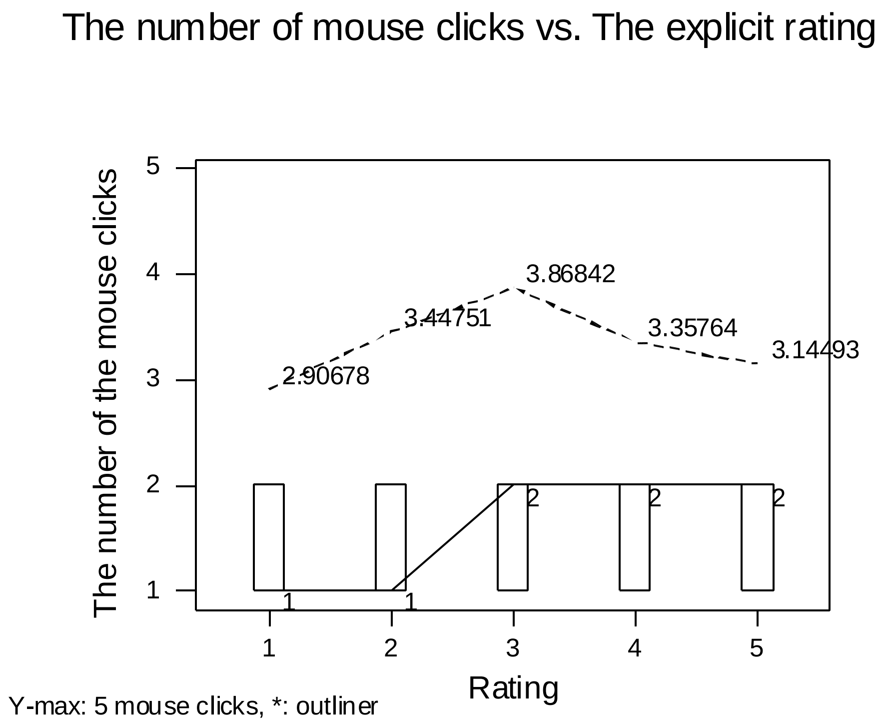



**Figure 8.4-(c): The upper dashed line shows the average number of mouse clicks. The lower line shows the median for the number of mouse clicks with the box of 95% confidence interval for the median. The maximum is 5 mouse clicks**

After running the descriptive statistic and the Kruskal-Wallis test, we can observe that there is no significant difference in the median for the number of mouse clicks by each explicit rating group. We can see that the averages of the number of mouse clicks are around 3 to 4 among the explicit rating groups. We can conclude that *clicking action is not an effective implicit rating, as there is less correlation between the number of mouse clicks and the explicit rating*.

### 8.3.6 Statistical Analysis for the Time Spent Scrolling by the Mouse vs. the Explicit Rating

Scrolling by the mouse includes the time spent scrolling the horizontal and the vertical scroll bar by the mouse. Some factors influence its accuracy. For example, some users prefer scrolling by the keyboard rather than by the mouse. In addition, no scroll bar exists when the browser window is large enough to show all HTML page. Moreover, some users move scroll bar by the mouse quicker than the others. We filtered out 1089 outliers that have 0 milliseconds spent scrolling by the mouse. The total number of the data points on which we ran the tests was 733. First we ran the Kruskal-Wallis test for the time spent scrolling by the mouse vs. the explicit rating. Table below 8.7-(a) shows the results of the test.

**Kruskal-Wallis Test: The time spent scrolling by the mouse vs. The explicit rating**

| Rating | N | Median | Ave Rank | Z |
|---|---|---|---|---|
| 1 | 71 | 5665 | 349.0 | -0.75 |
| 2 | 116 | 4813 | 334.4 | -1.81 |
| 3 | 200 | 6038 | 358.4 | -0.67 |
| 4 | 194 | 7518 | 383.1 | 1.24 |
| 5 | 152 | 8860 | 391.0 | 1.57 |
| Overall | 733 | | 367.0 | |

H = 6.67  DF = 4  P = 0.154

**H = 6.67  DF = 4  P = 0.154 (adjusted for ties)**

**Table 8.7-(a): Kruskal-Wallis Test: The time spent scrolling by the mouse vs. the explicit rating**

Null Hypothesis: The median of time spent scrolling by the mouse is the same for all explicit rating groups.

Hypothesis: The median of time spent scrolling by the mouse differs among the explicit rating groups.

Since the p-value is 0.154 > 0.05, the null hypothesis is accepted. The statistical conclusion is that the median values for the explicit rating groups contain at least two of them having the same value.

Next, we ran the descriptive statistics. Table 8.7-(b) shows the results.

**Descriptive Statistics: The time spent scrolling by the mouse vs. The explicit rating**

| Rating | N | Mean | Median | TrMean | StDev |
|---|---|---|---|---|---|
| 1 | 71 | 11981 | 5665 | 9154 | 18786 |
| 2 | 116 | 14475 | 4813 | 9532 | 29591 |
| 3 | 200 | 13246 | 6038 | 10562 | 20819 |
| 4 | 194 | 14825 | 7518 | 11093 | 24427 |
| 5 | 152 | 19902 | 8860 | 12806 | 40538 |

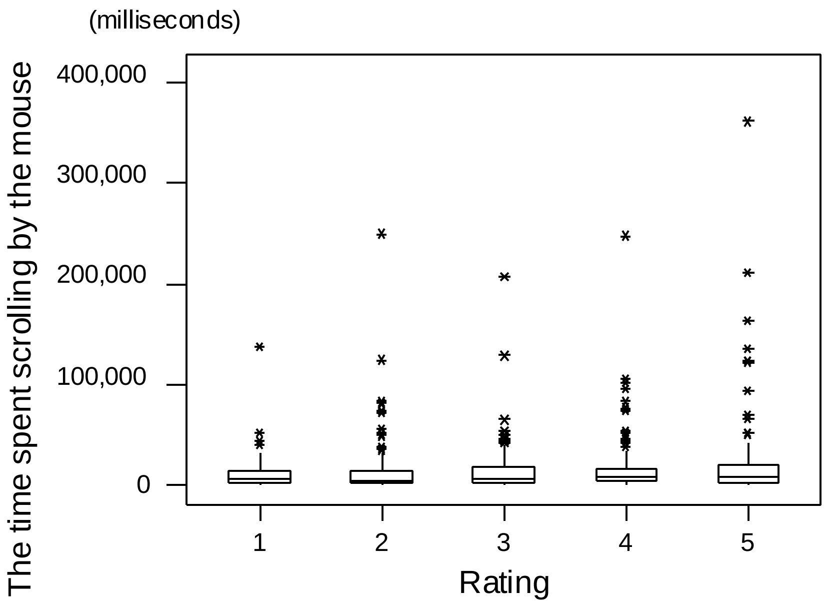
| Rating | SE Mean | Minimum | Maximum | Q1 | Q3 |
|---|---|---|---|---|---|
| 1 | 2230 | 17 | 137789 | 2146 | 14516 |
| 2 | 2747 | 373 | 249831 | 1882 | 14050 |
| 3 | 1472 | 57 | 207912 | 2121 | 17121 |
| 4 | 1754 | 19 | 247833 | 3209 | 16623 |
| 5 | 3288 | 15 | 361552 | 2698 | 20830 |

**Table 8.7-(b): Descriptive Statistics: The time spent scrolling by the mouse vs. the explicit rating**

We also analyzed the graphs of the time spent moving the mouse (y-axis) vs. the explicit rating (x-axis)

Figure 8.5-(a) shows the boxplots of the median for the time spent scrolling by the mouse with the maximum time of 420,000 milliseconds (7 minutes). Figure 8.5-(b) zoomed in the boxplots so that you can see the median and the first and third quartile for the median for the time spent moving the mouse. Figure 8.5-(c) shows the connected line of the median and the mean for the time spent moving the mouse as well as the 95 % confidence interval for the median with the maximum time of 30,000 milliseconds (30 seconds) spent moving the mouse.

The time spent scrolling by the mouse vs. The explicit rating




Y-max: 420,000 msec, *: outliner

**Figure 8.5-(a): the boxplots of the time spent scrolling by the mouse vs. the explicit rating with the Y-maximum, 420,000 milliseconds (7 minutes)**

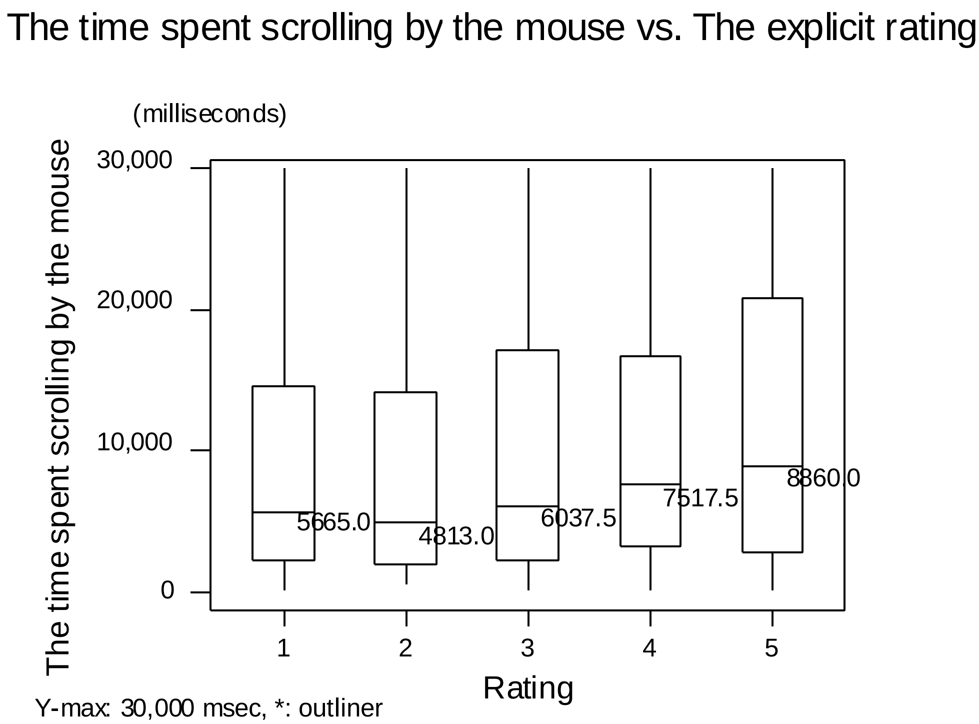



**Figure 8.5-(b): the boxplots of the time spent scrolling by the mouse vs. the explicit rating with the Y-maximum, 30,000 milliseconds (30 seconds)**

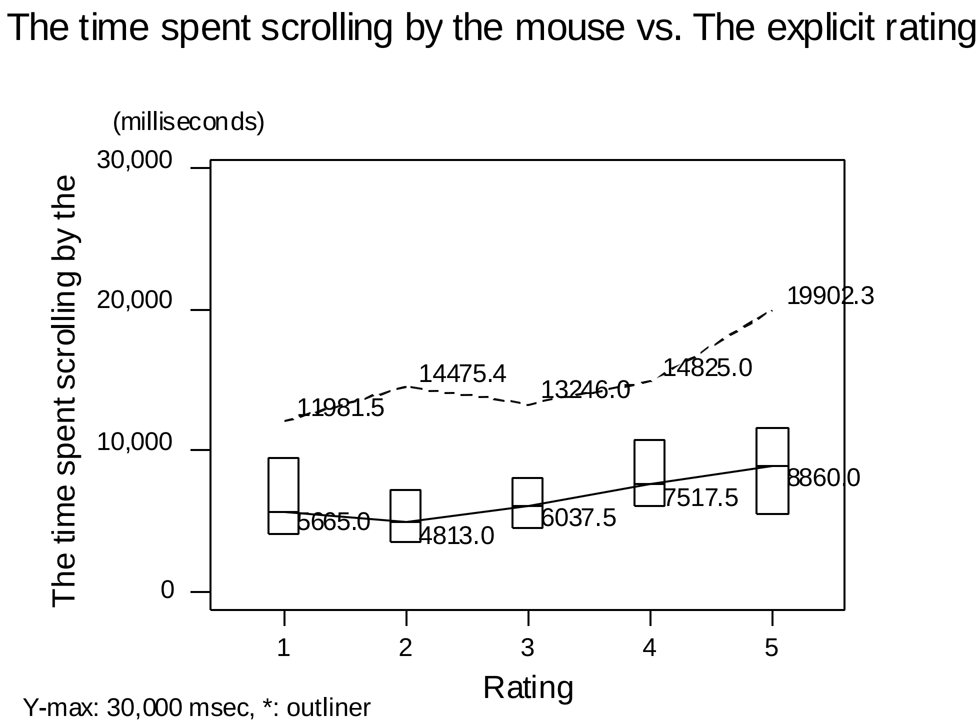



**Figure 8.5-(c): The upper dashed line shows the average time spent scrolling by the mouse. The lower line shows the median for the time spent scrolling by the mouse with the box of 95% confidence interval for the median. The maximum time is 30,000 milliseconds (30 seconds)**

From descriptive statistic and the Kruskal-Wallis test, we can observe that there is no significant difference in scrolling by mouse between two explicit rating groups. However, we can see that excluding the data in the explicit rating 1 or 2, there is positive relationship between the median for the time spent scrolling by the time and the explicit rating. In Figure 8.5-(c) above, you can also see that there is an increasing trend in the mean between the explicit rating 3 and the explicit rating 5. Therefore, our conclusion is that *the time spent scrolling by the mouse itself is not a good implicit indicator, and we need to add the time scrolling by using the keyboard to it*.

### 8.3.7 Statistical Analysis for the Time Spent Scrolling by the Keyboard vs. the Explicit Rating

Scrolling by the keyboard includes the time spent scrolling by the Up Arrow, Down Arrow, Page Up, and Page Down keys. The same factors as those of the time spent scrolling by the mouse influence its accuracy. For example, some users prefer scrolling by the mouse rather than by the keyboard. In addition, no scroll bar exists when the browser window is large enough to show the whole HTML page. We filtered out 1459 outliers that have 0 milliseconds spent scrolling by the keyboard. Thus, the number of data points for which we ran the tests was 363. First we ran the Kruskal-Wallis test for the time spent scrolling by the keyboard vs. the explicit rating. Table below 8.8-(a) shows the results of the test.

## Kruskal-Wallis Test: The time spent scrolling by the keyboard vs. The explicit rating

| Rating | N | Median | Ave Rank | Z |
|---|---|---|---|---|
| 1 | 39 | 543.0 | 104.3 | -4.90 |
| 2 | 47 | 1637.0 | 185.9 | 0.27 |
| 3 | 105 | 2719.0 | 199.0 | 1.97 |
| 4 | 87 | 2394.0 | 187.6 | 0.57 |
| 5 | 85 | 2381.0 | 188.8 | 0.68 |
| Overall | 363 | | 182.0 | |

H = 24.83  DF = 4  P = 0.000

**H = 24.84  DF = 4  P = 0.000 (adjusted for ties)**

**Table 8.8-(a): Kruskal-Wallis Test: The time spent scrolling by the keyboard vs. the explicit rating**

Null Hypothesis: The median time for the time spent scrolling by the keyboards is the same for all explicit rating groups.

Hypothesis: The median time for the time spent scrolling by the keyboards differs among the explicit rating groups.

Since the p-value is 0.000< .05, the null hypothesis is rejected and the statistical conclusion is that the median values for the explicit rating groups are varied.

Next, we ran the descriptive statistics. Table 8.8-(b) shows the results.

**Descriptive Statistics: The time spent scrolling by the keyboard vs. The explicit rating**

| Rating | N | Mean | Median | TrMean | StDev |
|---|---|---|---|---|---|
| 1 | 39 | 1763 | 543 | 939 | 4903 |
| 2 | 47 | 6621 | 1637 | 3993 | 15000 |
| 3 | 105 | 6356 | 2719 | 3927 | 14516 |
| 4 | 87 | 4735 | 2394 | 3583 | 8015 |
| 5 | 85 | 6109 | 2381 | 4327 | 12127 |

| Rating | SE Mean | Minimum | Maximum | Q1 | Q3 |
|---|---|---|---|---|---|
| 1 | 785 | 198 | 30704 | 308 | 1086 |
| 2 | 2188 | 138 | 87010 | 612 | 5271 |
| 3 | 1417 | 110 | 117340 | 780 | 5433 |
| 4 | 859 | 91 | 59032 | 735 | 4726 |
| 5 | 1315 | 99 | 97331 | 544 | 7135 |

**Table 8.8-(b): Descriptive Statistics: The time spent scrolling by the keyboard vs. the explicit rating**

We also analyzed the graphs of the time spent scrolling by the keyboard (y-axis) vs. the explicit rating (x-axis)

Figure 8.6-(a) shows the boxplots of the median for the time spent scrolling by the keyboard with the maximum time of 120,000 milliseconds (2 minutes). Figure 8.6-(b) zoomed in the boxplots so that you can see the median and the first and third quartile for the median for the time spent scrolling by the keyboard. Figure 8.6-(c) shows the connected lines of the median and the mean for the time spent scrolling by the keyboard

as well as the 95 % confidence interval for the median with the maximum time of 10,000 milliseconds (10 seconds) spent moving the mouse.

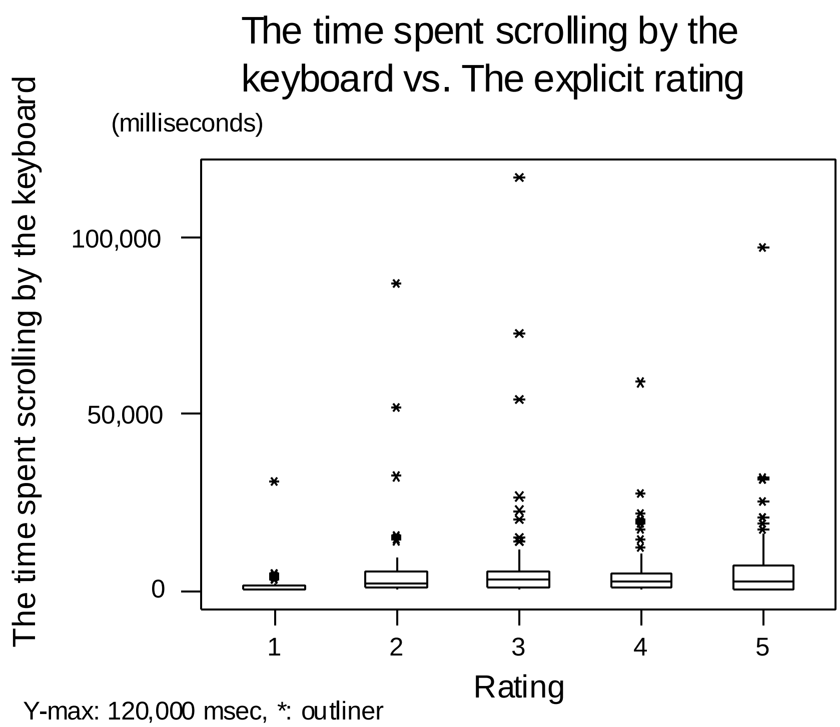



**Figure 8.6-(a): the boxplots of the time spent scrolling by the keyboard vs. the explicit rating with the Y-maximum, 120,000 milliseconds (2 minutes)**

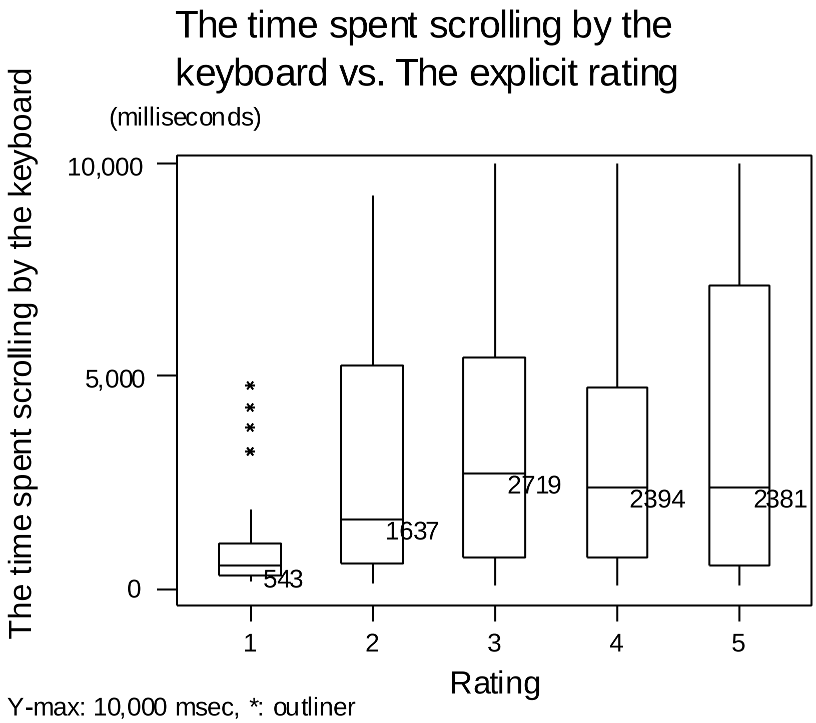



**Figure 8.6-(b): the boxplots of the time spent scrolling by the keyboard vs. the explicit rating with the Y-maximum, 10,000 milliseconds (10 seconds)**

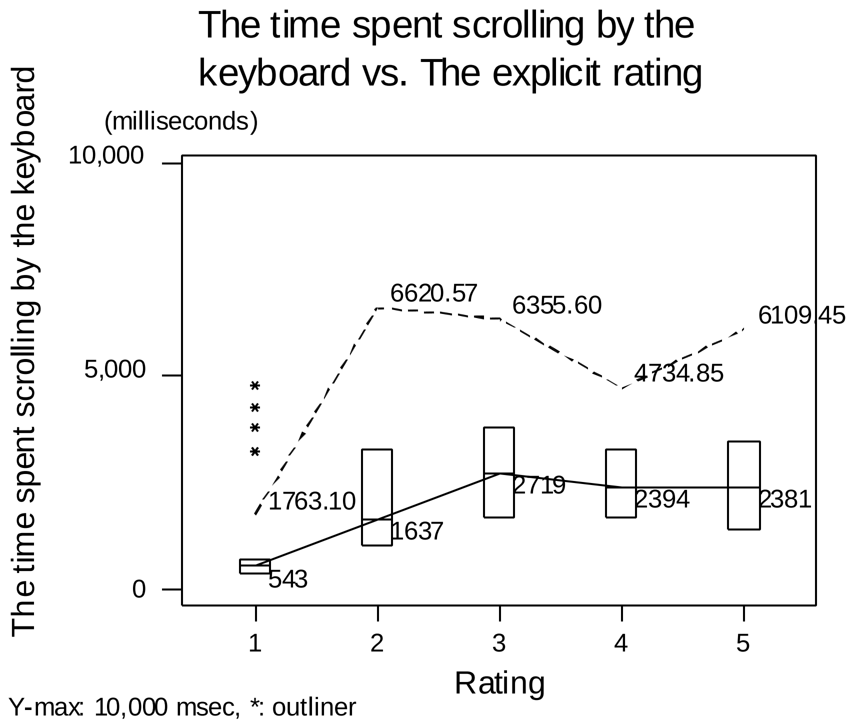



**Figure 8.6-(c): The upper dashed line shows the average time spent scrolling by the keyboard. The lower line shows the median for the time spent scrolling by the keyboard with the box of 95% confidence interval for the median. The maximum time is 10,000 milliseconds (10 seconds)**

From the descriptive statistics and the figures above, the median for the time spent scrolling by the scroll is not linearly proportional to the explicit rating. As we discussed above, some prefer scrolling by the mouse rather than by the keyboard. Some might use both methods to scroll. Therefore, our statistical conclusion is that *the time spent scrolling by the keyboard itself is not a good implicit indicator, and we need to add the time scrolling by using the mouse to it*.

### 8.3.8 Statistical Analysis for the Time Spent Scrolling by the Mouse and the Keyboard vs. the Explicit Rating

As we discussed in the last two sections, we need to add the time spent scrolling by the mouse to the time spent scrolling by the keyboard. However, some factors still influence its accuracy. For example, no scrolling is necessary if the browser window is large enough to show the entire HTML page. We filtered out 852 outliers that have 0 milliseconds spent scrolling by the mouse and by the keyboard. The total number of data points for which we ran the tests was 970. First we ran the Kruskal-Wallis test for the time spent scrolling by the mouse vs. the explicit rating. Table below 8.9-(a) shows the results of the test.

**Kruskal-Wallis Test: The time spent scrolling by the mouse and the keyboard vs. The explicit rating**

| Rating | N | Median | Ave Rank | Z |
|---|---|---|---|---|
| 1 | 101 | 3485 | 390.2 | -3.61 |
| 2 | 151 | 4079 | 447.4 | -1.82 |
| 3 | 270 | 5268 | 493.3 | 0.54 |
| 4 | 243 | 6444 | 512.8 | 1.75 |
| 5 | 205 | 7424 | 518.0 | 1.87 |
| Overall | 970 | | 485.5 | |

H = 19.76  DF = 4  P = 0.001

**H = 19.76  DF = 4  P = 0.001 (adjusted for ties)**

**Table 8.9-(a): Kruskal-Wallis Test: The time spent scrolling by the mouse and the keyboard vs. the explicit rating**

Null Hypothesis: The median time for the time spent scrolling by the mouse and the keyboard is the same for all explicit rating groups.

Hypothesis: The median time for the time spent scrolling by the mouse and the keyboard differs among the explicit rating groups.

Since the p-value is 0.001 < .05, the null hypothesis is rejected and the statistical conclusion is that the median values for the explicit rating groups are varied.

Next, we ran the descriptive statistics. Table 8.9-(b) shows the results.

**Descriptive Statistics: The time spent scrolling by the mouse and the keyboard vs. The explicit rating**

| Rating | N | Mean | Median | TrMean | StDev |
|---|---|---|---|---|---|
| 1 | 101 | 9103 | 3485 | 6401 | 19006 |
| 2 | 151 | 13181 | 4079 | 8396 | 27766 |
| 3 | 270 | 12283 | 5268 | 9403 | 20065 |
| 4 | 243 | 13531 | 6444 | 10097 | 23471 |
| 5 | 205 | 17290 | 7424 | 11055 | 37628 |

| Rating | SE Mean | Minimum | Maximum | Q1 | Q3 |
|---|---|---|---|---|---|
| 1 | 1891 | 17 | 168493 | 756 | 10589 |
| 2 | 2260 | 138 | 249831 | 1480 | 11877 |
| 3 | 1221 | 57 | 207912 | 2400 | 15064 |
| 4 | 1506 | 19 | 265233 | 2451 | 15750 |
| 5 | 2628 | 99 | 362426 | 2049 | 19002 |

**Table 8.9-(b): Descriptive Statistics: The time spent scrolling by the mouse and the keyboard vs. the explicit rating**

We also analyzed the graphs of the time spent scrolling by the mouse and by the keyboard (y-axis) vs. the explicit rating (x-axis). Figure 8.7-(a) shows the boxplots of the median for the time spent scrolling by the mouse and by the keyboard with the maximum time of 420,000 milliseconds (7 minutes). Figure 8.7-(b) zoomed in the boxplots so that you can see the median and the first and third quartile for the median for the time spent scrolling by the mouse and by the keyboard. Figure 8.7-(c) shows the connected lines of the median and the mean for the time spent scrolling by the mouse and by the keyboard as well as the 95 % confidence interval for the median with the maximum time of 20,000 milliseconds (20 seconds) spent scrolling by the mouse and by the keyboard.

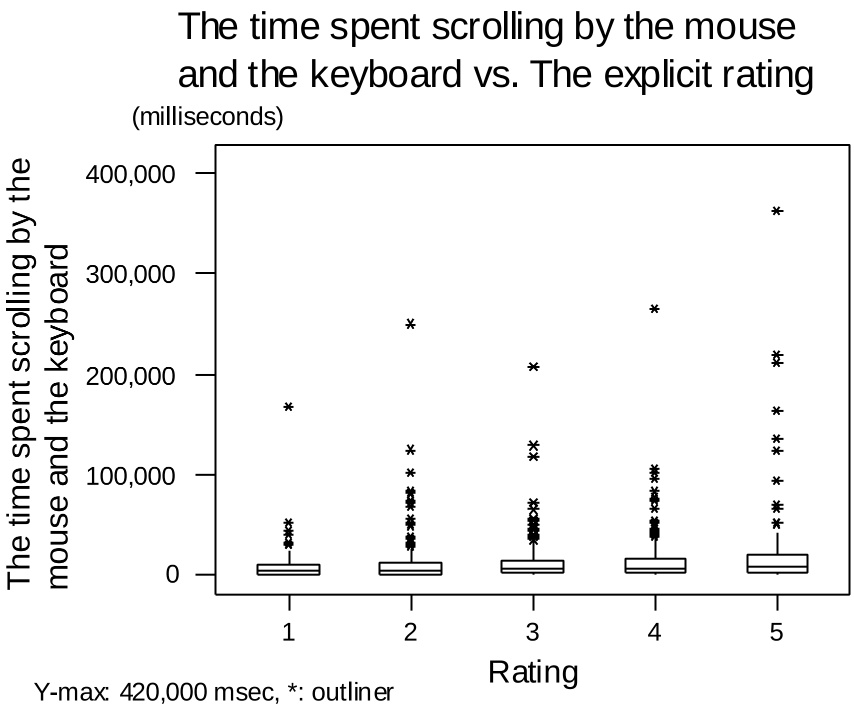



**Figure 8.7-(a): the boxplots of the time spent scrolling by the mouse and the keyboard vs. the explicit rating with the Y-maximum, 420,000 milliseconds (7 minute)**

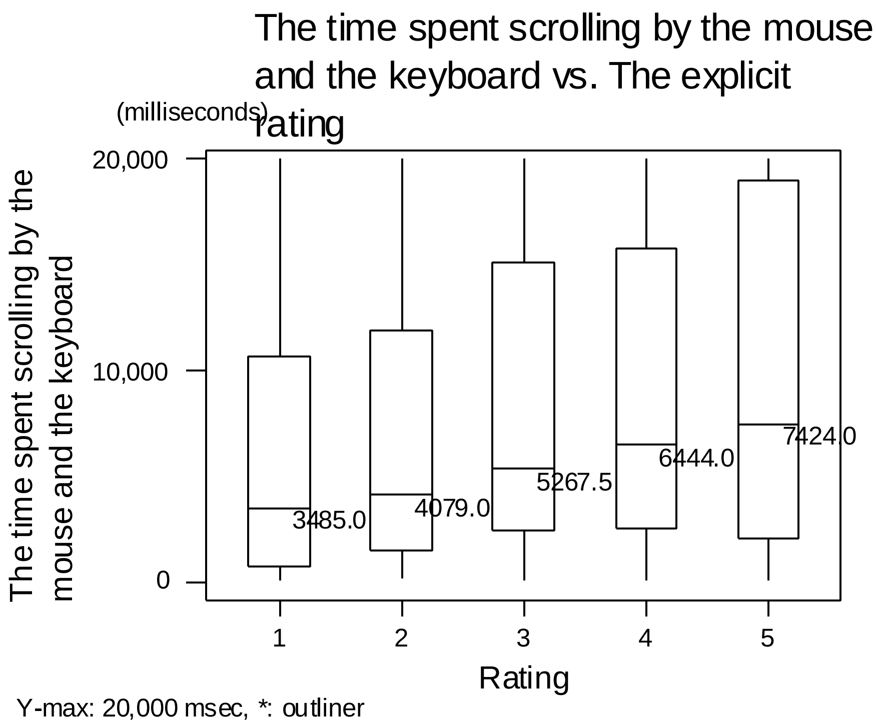



**Figure 8.7-(b): the boxplots of the time spent scrolling by the mouse and the keyboard vs. the explicit rating with the Y-maximum, 20,000 milliseconds (20 seconds)**

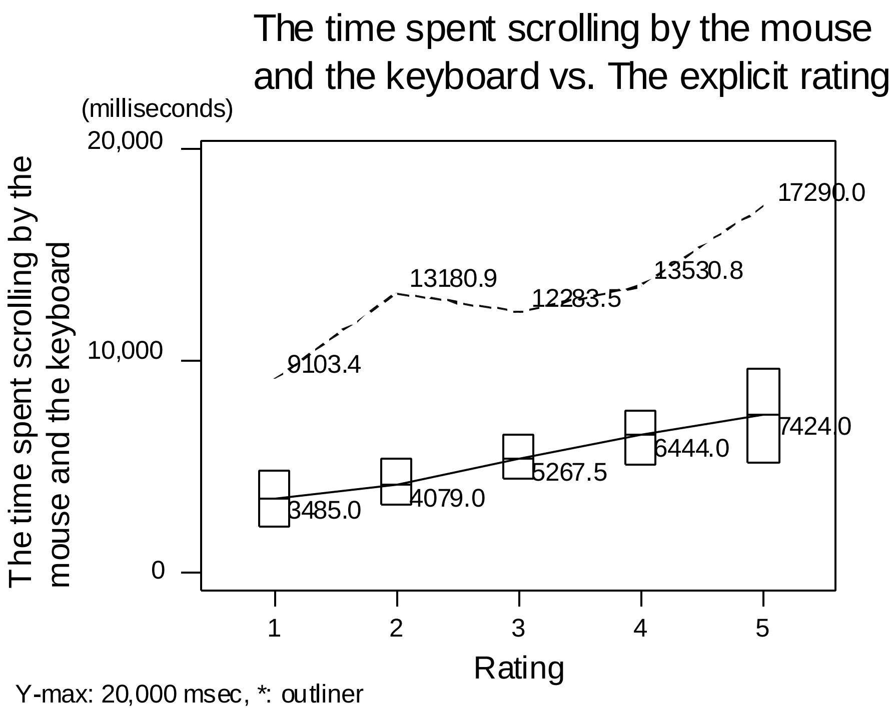



**Figure 8.7-(c): The upper dashed line shows the average time spent scrolling by the mouse and the keyboard. The lower line shows the median for the time spent scrolling by the mouse and the keyboard with the box of 95% confidence interval for the median. The maximum time is 20,000 milliseconds (20 seconds)**

From the descriptive statistics and the figures above, the median for the time spent scrolling by the mouse and by the keyboard is linearly proportional to the explicit rating. Compared with the time spent scrolling by the mouse itself and by the keyboard itself, this implicit indicator is much accurate. Therefore, our statistical conclusion is that *the time spent scrolling by the mouse and by the keyboard is a good implicit indicator and the longer the time spent scrolling, the higher the explicit rating.*

## 8.4 Combined Indicators vs. the Explicit Rating

### 8.4.1 Meaning of the Combined Indicators

While running statistical analysis for each indicator, we realize that the use of combined indicators will allow us to see a greater correlation with the rating of the page. The combined indicators will allow us to observe more clearly the connection between the actions with the rating of the page. For example, some users scroll down a Web page while reading it. Some users read a Web page while moving the mouse.

### 8.4.2 Selection of the Combination of the Indicators

First, we did not choose the time spent on a page even though it shows the strong correlation with the explicit rating. This is because this implicit rating includes all the other implicit ratings except the number of mouse clicks. Thus, we eliminated it from the selection. From the results of the section 8.3, “Analysis for each indicator toward the explicit rating”, we saw that the time spent reading a page and the time spent scrolling by the mouse and by the keyboard show the stronger correlation with the explicit rating than the other implicit ratings. Therefore, we chose these two implicit ratings for our proposed combined indicator

### 8.4.3 Combined Indicator (the time spent reading a page and the time spent scrolling by the mouse and by the keyboard) vs. the Explicit Rating

In order to get the Combined Indicator, we did the multivariate regression analysis for these two implicit ratings. We used all records (1822 records ) that we received in the user experiment to calculate it. We divide the records into two parts: those who have scroll activities (967 records) and those who do not (855 records). We did the multiple regression analysis for these two implicit ratings with the records that have scroll activities and we did the regression analysis for the time spent reading a page with the records that do not have any scroll activities. By doing this, the regression analysis can generate the more accurate formula. The result of the multiple regression analysis is:

**Multiple Regression Analysis: The explicit rating vs. the time spent reading a page and the time spent scrolling by the mouse and by the keyboard**

The regression equation is:

$$(PredictedExplicitRating) = 3.29 + 1.597 \cdot 10^{-7} \cdot (TimeSpentReading) + 2.1548 \cdot 10^{-6} \cdot (TimeSpentScrolling)$$

S = 1.243 R-Sq = 0.6%

**S** is the estimated standard deviation about the regression line. **R-sq** (R-Squared) is the coefficient of determination. It is calculated as R-Squared = Correlation(the explicit rating, the predicted by the time spent reading a page and by the time spent scrolling by the mouse and by the keyboard).

**Regression Analysis: the explicit rating vs. the time spent reading a page**

The regression equation is

$$\left(PredictedExplicitRating\right) = 3.17 + 19843 \cdot 10^{-6} \cdot (TimeSpentReading)$$

S = 1.328      R-Sq = 0.7%

Therefore, the formula for the Combined Indicator is:

If scroll activities exist,

$$\left(PredictedExplicitRating\right) = 3.29 + 1.597 \cdot 10^{-7} \cdot (TimeSpentReading) + 2.1548 \cdot 10^{-6} \cdot (TimeSpentScrolling)$$

If no scroll activities exist,

$$\left(PredictedExplicitRating\right) = 3.17 + 19843 \cdot 10^{-6} \cdot (TimeSpentReading)$$

### 8.4.4 Comparison between the Combined Indicator vs. the Time Spent on a Page

In order to know our proposed combined indicator, we compared it with the time spent on a page. We did the regression analysis for the time spent on a page with the same two records as we used for the combined indicator. The result is:

**Regression Analysis: the explicit rating vs. The time spent on a page with the records that have scroll activities.**

The regression equation is

$$\left(PredictedExplicitRating\right) = 3.31 + 2.537 \cdot 10^{-7} \cdot (TimeSpentOnAPage)$$

S = 1.243      R-Sq = 0.5%

**Regression Analysis: Rating versus The time spent on a page with the records that do no have any scroll activities.**

The regression equation is

$$\left(PredictedExplicitRat\,ing\right) = 3.17 + 19839 \cdot 10^{-6} \cdot (TimeSpentOnAPage)$$

S = 1.328 R-Sq = 0.7%

As you see the results, on the tests with the records that have scroll activities, there is 0.1 % difference on the R-squared between the combined indicator and the time spent on a page. Thus, we can say that our combined indicator is more accurate than the time spent on a page itself if scroll activities exist. However, 0.1% is too small to determine that there is significant difference between them. Therefore, we conclude that *there is no difference between the combined indicator (the time spent reading a page and the time spent scrolling by the mouse and by the keyboard) and the time spent on a page, and that the combined indicator is as accurate as the time spent on a page*.

## 8.5 Conclusion

Through the analysis of each indicator and the combined indicator, we conclude that the time spent on a page and the combined indicator (the time spent reading a page and the time spent scrolling by the mouse and by the keyboard) show better correlations with the explicit rating than the other indicators do. However we can not determine which indicator is better than the other since we found that both indicators show almost same correlation with the explicit rating.  The table below shows the result of each indicator that we analyzed.

| Indicator | Result |
|---|---|
| The time spent on a page | There is a strongly positive correlation between the time spent on a page and the explicit rating **(Good Implicit Indicator)** |
| The time spent reading a page | There is a strong positive correlation between the time spent reading a page and the explicit rating **(Good Implicit Indicator)** |
| The time spent moving the mouse | There is the positive relationship between the time spent moving the mouse and the explicit rating, but this implicit indicator is not as good as the time spent reading a page. **(Not Good  Implicit Indicator)** |
| The number of mouse clicks | Clicking is not an effective implicit rating, as there is less correlation between the number of mouse clicks and the explicit rating.   **(Bad Implicit Indicator)** |
| The time spent scrolling by the mouse | The time spent scrolling by the mouse itself is not good implicit indicator, and we need to add the time scrolling by the keyboard to it.   **(Not Good Implicit Indicator)** |
| The time spent scrolling by the keyboard | The time spent scrolling by the keyboard itself is not good implicit indicator, and we need to add the time scrolling by the mouse to it.   **(Not Good Implicit Indicator)** |
| The time spent scrolling by the mouse and by the keyboard | The time spent scrolling by the mouse and by the keyboard is good implicit indicator and the longer time spent scrolling, the higher explicit rating.  **(Good Implicit Indicator)** |
| The combined indicator (the time spent reading a page and the time spent scrolling by the mouse and by the keyboard) | The combined indicator is as accurate as the time spent on a page.  **(Good Implicit Indicator)** |

**Table 8.10: The result of the analysis for each implicit indicator.**

# 9. Conclusion and Future Work

## 9.1 Conclusion

The goal of our MQP was to find the correlation between the implicit ratings and the explicit rating. In order to accomplish this goal, we built a browser named "Curious Browser" that would record both implicit ratings and explicit rating of a Web page. Next, we conducted a user experiment in order to gather data. Finally, we analyzed the collected data. As indicated in the chapter 8, "Analysis", we have successfully found correlations between some implicit ratings such as the time spent on a page, the time spent reading a page, and the time spent scrolling by the mouse and by the keyboard and the explicit ratings. We combined the reading time and the scrolling time to create a new combined implicit indicator. In order to find the accuracy of the new combined indicator, we compared it with the time spent on a page. We have found that there is no significant difference between the combined indicator and the time spent on a page in terms of its accuracy.

Throughout out MQP, we have learned many things throughout the project. These are the list of things that we learned.

- How implicit interest indicators work.
- How explicit interest indicators work.
- What filtering/recommendation system is.
- How to apply many methods of statistical analysis to our data.
- How to develop the program (Software Engineering).
- How to set up the user experiment.

## 9.2 Future Work

We have found one bug from the users' feedback on the experiment. On Internet Explorer 4.0, when a user clicks a link that will open another browser window, the computer significantly slows down. It takes approximately 3 minutes for the computer to recover. However, this bug does not occur on Internet Explorer 5.0.

The Curious Browser can be improved by some approaches. One approach is that when a Web page uses frames, implicit ratings should be collected and distinguished by each frame. Another approach is to find a way to collect scroll events and scrolling time directly from the change of scroll objects, not from the boundary of the browser window. This will make the implicit rating of scroll events and scrolling time more accurate. The last approach is to add more implicit ratings to the Curious Browser, such as detecting highlighting texts of a Web page, bookmarking, or printing. We could also improve the database of the Curious Browser. We can use the Microsoft Access or the Oracle 8 to store the data. This will make the analysis easier.

Our user experiment can be improved as well. If a larger group of users participate in the experiment, we will get more accurate data. With the larger data, we may be able to find another interesting trend of the implicit rating vs. the explicit rating. We did the experiment at WPI in our project. If users from the different location or of different ages participate in the experiment, we might get the different result.

Finally our analysis can be improved. There are many statistical analysis methods that we do not know. Some of these statistical analyses could be used in our analysis. Especially, the analysis for the combined implicit rating can be improved by them. Since some people's behaviors while surfing the Web are different from others, we can compare the implicit ratings for each user vs. the explicit ratings. Through this analysis, we will find the correlation between them by each user. We could also compare the implicit ratings and the explicit ratings by each Web site. This analysis will give us interesting results. We can analyze the trend of the implicit rating by the grouped explicit rating such as 1 and 2 or 4 and 5. Since the median and the mean for some implicit indicators such as the time spent scrolling by the mouse and by the keyboard seems to be cubically proportional to the explicit rating, this analysis will give us interesting results.

With the Curious Browser and the result of the analysis through the data, the next step of this project would be to create a recommendation system that will recommend Web pages where a user will have the most interest based only on implicit interest indicators.

# Appendix A. Flyer for the User Experiment

# Appendix B Data Files in Compact Disc

The Compact Disc is on the last page. It contains:

- MQP Report (under "Report" directory") as Microsoft Word format.
- Visual Basic 6.0 code for Curious Browser program, "Curious Browser.vbp" (under "CuriousBrowser" directory).
- The executable code for the Curious Browser, "Curious Browser.exe" (under "CuriousBrowser" directory).
- All data collected in the user experiment (under "Data" directory) and under this directory, we have:
  - ♦ Two Microsoft Excel files: One contains all data without error, and the other contains all data without error and the explicit rating 0.
  - ♦ "Wine" directory: all data from the Wine Lab (Excel format).
  - ♦ "Adp" directory: all data from the Adp Lab (Excel format).
- Flyer for the user experiment (under "Flyer" directory) as postscript format and Microsoft Publisher 2000 format.

# References

Avery, C. and Zeckhauser, R. (1997). Recommender Systems for Evaluating Computer Messages. *Communications of the ACM.* Vol. 40, No. 3, pp. 88-89. Internet Site: http://www.acm.org/pubs/citations/journals/cacm/1997-40-3/p88-avery/

Goecks, J. and Shavlik J. (1999). Automatically Labeling Web Pages Based on Normal User Actions. *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*. Stockholm, Sweden: Internet Site: http://www.cs.wisc.edu/~shavlik/abstracts/goecks_IJCAI_ML_IF_workshop99.pdf.abstract.html

Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM.* Vol. 40, No. 3, pp. 77-87. Internet Site: http://www.acm.org/pubs/citations/journals/cacm/1997-40-3/p77-konstan/

Lam, W. and Yu, K. L. (1999). An Intelligent Adaptive Filtering Agent Based on an On-line Learning Model. *Proceedings on the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. Berkeley, CA: pp. 287-288. Internet Site: http://www.acm.org/pubs/citations/proceedings/ir/312624/p287-lam/

Morita, M. and Shinoda, Y. (1994). Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval. *Proceedings of the seventeenth annual international ACM-SIGIR conference on Research and development in information retrieval*. Dublin, Ireland: pp. 272-281. Internet Site: http://www.acm.org/pubs/citations/proceedings/ir/188490/p272-morita/

Nichols, D. M. (1997). Implicit Rating and Filtering. *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*. Budapest, Hungary: pp. 31-36. Internet Site: http://www.ercim.org/publication/ws-proceedings/DELOS5/index.html

Oard, D. W. and Marchionini, G (1996). A conceptual Framework for Text Filtering. *Technical Report CAR-TR-830*. Human Computer Interaction Laboratory, University of Maryland at College Park Internet Site: http://www.ece.umd.edu/~oard/research.html

Watson, A. and Sasse, M. A. (1998). Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications. *Proceedings of the 6th ACM international conference on Multimedia*. Bristol, United Kingdom: pp. 55-60. Internet Site: http://www.acm.org/pubs/citations/proceedings/multimedia/290747/p55-watson/