Modeling How Users Interact with Windows Outlook to Create Realistic Email Traffic

by

Lisa Hsu

Submitted to the Department of Electrical Engineering and Computer Science on September 7, 2006 in partial fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology

September 7, 2006

Author_____
Department of Electrical Engineering and Computer Science
September 7, 2006

Certified by_____
hard Lippmann
ln Laboratory
is Supervisor

Accepted by____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Modeling How Users Interact with Windows Outlook to Create Realistic Email Traffic

by

Lisa Hsu

Submitted to the
Department of Electrical Engineering and Computer Science

September 7, 2006

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

The ever-present and increasing threat of abuse requires a systematic approach to information assurance to protect the security of systems and data. The Lincoln Adaptable Real-time Information Assurance Testbed (LARIAT) was developed to simplify and address problems that surfaced from DARPA evaluations on intrusion detection systems (IDS) development. LARIAT emulates the network traffic produced from one or more organizations connected to the internet. This thesis work focuses on developing the Outlook email model in WinNTGen, which simulates native Windows traffic in LARIAT. To accurately characterize email network traffic, data from seven real users is collected using an Outlook add-in built on the Microsoft .NET Framework for analysis to produce a more realistic usage behavior model. The analysis determined that users behave differently. Therefore, a state machine of the 20 prevailing user actions, and the 76 prevailing transitions was created for each user, to model each user separately.

Thesis Supervisor: Richard Lippmann
Title: Senior Scientist, M.I.T. Lincoln Laboratory

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This thesis could not have been possible without the unwavering support and expert guidance of a few dedicated, patient, and trusting individuals both in and out of Lincoln Laboratory. Together, they have provided a collaborative environment that made research challenging yet fruitful, well-scoped yet accommodating, and demanding yet rewarding.

At Lincoln, I was fortunate to find a thesis supervisor in Richard Lippmann, who always made it a point to be readily available for solid direction in all aspects of my project, from the theory behind statistical modeling based on limited data to the language-specific software implementation details of certain project requirements. Doug Stetson, as my research advisor, further improved the experience by providing me thorough and patient assistance on any and every problem or need I encountered along the way. I cannot fathom how he endured the persistent snags and setbacks with me, demonstrating, at each occasion, his extraordinary professionalism and ability to adapt through the glitches and changes in my research. I would also like to thank the participants who suffered software inconsistencies for months at a time so that I could collect their usage data.

I am so thankful to have had the opportunity to work at a prestigious and eminent research institute with such distinguished and knowledgeable staff members as MIT Lincoln Laboratory.

I would also like to thank my MIT mentors, family, and friends for their support throughout this time. Professor Paul Gray and George Kocur, thank you for supporting me in my endeavors. My mother, though she never once questioned my ability to finish the Master's program, also never pushed me to do so. My sister, having gone through the experience herself, did the only thing that could possibly help, barring writing my thesis for me. She was always there to provide a shoulder, a quiet working environment,

groceries, or diversions. Sally, Christine, Paul, Brian, Jack, Max, along with a multitude of others, were there to provide the requisite uncritical but effective "get it done" speech I always needed to hear. My commiserating thesis companions, Felicia and Mingyan, also deserve thanks for joining me in late night writing sessions.

Finally, no thesis experience is complete without the unwavering support and calm assurances from Anne Hunter and Vera Sayzew to whom I am forever indebted for guiding my hand through the many ups and downs in the past five years.

Chapter 1

INTRODUCTION

Each day, the global network is prey to thousands of costly attacks as hackers exploit both old and new system vulnerabilities. In 2003, network attacks cost industry an estimated $55 billion in productivity loss, double the amount in 2002 [1]. Single-handedly, the Slammer worm caused between $950 million and $1.2 billion in lost productivity worldwide during its first five days in debut [2]. As systems and networks grow, they are also becoming increasingly vulnerable to electronic attacks. Internet security statistics garnered from a six-month period in early 2004 showed a 12% increase in the number of security vulnerabilities and a 19% increase in the quantity of worms and viruses from the previous year [3]. The same study found that, on average, companies are subject to 38 attacks per week, with 64% of all new attacks exploiting vulnerabilities within one year from the time they are either discovered in existing software or introduced through patches or upgrades.

Concern about information assurance pervades the government as much as industry. In October, 2001, the President issued an Executive Order on Critical Infrastructure Protection, stating the following:

> The information technology revolution has changed the way business is transacted, government operates, and national defense is conducted. Those three functions now depend on an interdependent network of critical information

> infrastructures. The protection program authorized by this order shall consist of continuous efforts to secure information systems for critical infrastructure. Protection of these systems is essential to the telecommunications, energy, financial services, manufacturing, water, transportation, health care, and emergency services sectors. It is the policy of the United States to protect against disruption of the operation of information systems for critical infrastructure and thereby help to protect the people, economy, essential human and government services, and national security of the United States [4].

To detect, isolate, and mitigate costly attacks, the intrusion-detection system (IDS) product market has become a requisite and profitable industry, burgeoning from $40 million in 1997 to $120 million in 2004 [5,6].

1.1 Information Assurance and Intrusion Detection

Given the ever-present and increasing threat of abuse, a significant need exists for a systematic approach to protecting the security of systems and data. The field of Information assurance (IA) focuses primarily on protecting network devices, applications, or data. This includes the development, use, and evaluation of information technology (IT) products and systems, as well as the development of IT security standards, such as security requirements, test methods, tools, techniques, and metrics. Intrusion detection, which falls under the realm of IA, focuses on security tools and products designed to protect networks, system resources, and data against harmful or unwanted attacks. Typically, intrusion detection systems (IDSs) attempt to fulfill this role by identifying system misuse through the examination of network traces and audit logs from individual hosts [7].

1.1.1 *Challenges in IA and ID Development*

There are many challenges to the development of IA products, given the exponential growth of information, data, and technology, compounded by the complexity of networks, and the rapidly changing nature of cyber threats [8].

1.1.2 *Challenges in IA and ID Evaluation*

In 1998 and 1999, DARPA conducted off-line intrusion detection evaluations using realistic background traffic and many examples of realistic attack. The objectives for these evaluations were to assess the performance of DARPA-funded ID technology and to support the researchers developing that technology. Four important lessons emerged from the DARPA evaluations regarding the testbed upon which the ID technologies were measured [7]. First, no testbed can be complete. While the evaluation used a reasonably representative set of attacks, the set was by no means an exhaustive representation of existing attacks, nor did those attacks characterize the full range of actions that could be performed as part of each attack. Second, the testbed did not accurately model the network, using instead a simple network topology with a non-restrictive security policy. Third, the testbed did not accurately characterize background traffic. The probabilistic low-volume background traffic was generated by a limited number of victim machines. Finally, the testbed configuration was inflexible and non-adaptable, requiring extensive instrumentation to provide inputs to intrusion detection systems.

1.2 LARIAT Evaluation Testbed

Without a scalable and comprehensive testbed to provide realistic user-modeled network traffic, it is impossible to provide the requisite consistency to thoroughly test and evaluate different ID technologies, especially those technologies monitoring intrusions at the host level. The Lincoln Adaptable Real-time Information Assurance Testbed (LARIAT) was developed to address the problems that surfaced from the DARPA evaluations and to simplify IDS development and evaluation [9]. LARIAT allows researchers and operational users to configure and run real-time intrusion detection and

correlation tests with robust background traffic and attacks in their laboratories. Successful models of attacks, IDSs, and IDS alerts reduce the cost of experimentation, allow proof-of-concept analysis and simulations, and form the foundation of a theory of intrusion detection [10].

1.2.1 *LARIAT Network Traffic Generator Components*

The current version of LARIAT emulates the network traffic from one or more organizations connected to the internet. LARIAT functionality includes an internet emulator, Unix/Linux traffic generators for both a single client and virtual hosts, Windows traffic generators simulating native Windows traffic (WinNTGen), scheduled and controlled attack scenarios, and a GUI to facilitate experiment configuration and control [10].

1.2.2 *Windows Network Traffic Generator*

Traffic generated by Windows OS users is particularly important because it dominates many networks. Microsoft Windows holds a roughly 90% share of the client operating systems market [11]. An effective emulation of the network would therefore need to provide a rich and expansive Windows simulator.

Jesse Rabek's thesis work initiated the development of WinNTGen, which simulates the actions of a user controlling Windows applications that in turn use network resources [7].

On a Windows host, LARIAT consists of two primary components, LariatGina and WinNTGen. LariatGina communicates with the LARIAT director, obtains experiment parameters from the database, and logs in a simulated virtual user during an experiment. WinNTGen controls applications via application use state machines

(AUSMs) according to the experiment parameters, and logs out the virtual user at the appropriate time.

Individual users are modeled as a collection of AUSMs. Each AUSM encapsulates a user's behavioral pattern for a particular class of network traffic presence such as browsing the web, exploring the file system, or sending email. In WinNTGen, five AUSMs were created that control Internet Explorer, Windows Explorer (file browsing), Word, Chat, and Outlook. The transition parameters in the Internet Explorer and Windows Explorer AUSMs, which govern the likelihood that one action follows another, were set from the modeling of actual recorded user data [7]. The Outlook transition parameters were extrapolated from the Internet Explorer data, not modeled after email usage data [7].

1.3 Creating Accurate Outlook User Models

This thesis work focuses on developing the Outlook AUSM of WinNTGen. To improve the Outlook AUSM functionality, I collected Outlook usage data from MIT Lincoln Laboratory staff members and students for analysis. To collect data, I wrote a program called OutlookMonitor. This program monitors events generated by Outlook, such as sending, receiving or deleting mail, and automatically logs the data in a file on a user's local machine. To begin analysis, I developed a parsing program to filter and extrapolate the set of defined actions from a user's event log. Then, I analyzed the data to obtain the action stream, which contain user actions, action sequences, and timing between actions. I then analyzed the action stream to generate Outlook usage models that enrich the existing model with a wider range of actions and a more accurate representation of actual usage.

The model of usage that is formed from the data collected from users will be established from the answers to two questions. First, do users exhibit behavioral

similarities as far as the actions, sequence of actions, and timing between actions they perform? Second, what email-related actions are being driven?

If there are behavioral usage similarities, is it possible to generalize behavior across users? If no such generalizations can be made, then the model must attempt to define and understand the differences that exist. The data analysis should lend insight into the task of email message processing, such as the number of messages that are sent, the number of messages that are received, the amount of time spent processing email, and the format of information that is transmitted through email.

1.4 Thesis Outline

This thesis describes the motivation for improving the modeling of users in the Outlook AUSM in Chapter 2. Chapter 3 states the modeling and design goals for this research. Chapter 4 provides a detailed description of the implementation of the instrumentation tools to collect usage data and analyze it to formulate rich and realistic user models. Chapter 5, presents the results from the data collected from Lincoln Laboratory users. Chapter 6 draws conclusions on the updated modeling of the Outlook AUSM and addresses the need for future work.

Chapter 2

MOTIVATION AND BACKGROUND

In this chapter, I describe the motivation and background for continuing development of LARIAT's WinNTGen to create realistic and accurate user models for the Outlook AUSM. To do so, I describe the intrusion detection systems industry, LARIAT's role in intrusion detection, the role of the Windows network traffic generator in LARIAT, and the role of the Outlook AUSM in WinNTGen.

2.1 Intrusion Detection

Each year, approximately $1 trillion is spent domestically by companies, organizations, and individuals on information technology [12]. It is, therefore, not surprising that protecting IT investments is a high priority. And with the mounting threat of abuse and attacks, the need to find an appropriate method of thwarting and protecting against loss from IT vulnerabilities is a high priority.

However, every organization has a unique set of IT requirements and places a different value upon the security and fidelity of their information. The difference between each organization's IT infrastructure and valuation of information means that no single IDS can safeguard all organizations from harm in an equally effective manner.

Because of the specificity and uniqueness of needs, each IT department is generally responsible for assessing how different ID technologies perform on its own network, or simply rely on published results or a marketing sales pitch. The problem with performing a personalized assessment is that an organization must install, configure, and test multiple different ID technologies on a fluctuating and open network, or simply rely on published results or sales pitches. There is no assurance that the tests are accurate or fair, or even that they cover every critical circumstance. This self-assessment strategy results in a lot of excess work with no quality assurance guarantee.

2.2 LARIAT's Role in the Evaluation of Intrusion Detection Systems

In 1998 and 1999, MIT Lincoln Laboratory conducted IDS evaluations for DARPA. The evaluations were intended to provide ID researchers with many examples of attacks and normal traffic and to provide the DARPA program with a thorough assessment of research IDS performance [9].

The 1998 and 1999 DARPA IDS evaluations revealed that the evaluation of different intrusion detection systems is hindered by a lack of standardization and flexibility of customizations in a controlled, yet fully representative network environment. LARIAT grew in response to the need for a reliable, standardized, controlled, and adaptable method for testing IDSs.

LARIAT meets these needs through three primary features. First, LARIAT allows the flexibility for multiple network configurations so that organizations testing on LARIAT can perform experiments tailored to their specific network environments. This customizability means that organizations do not have to spend time developing their own testbeds for IDS evaluation.

Second, LARIAT is a controlled system that allows researchers to create scripted attacks that can be run and re-run with varied background traffic and background

attacks. A controlled environment ensures that organizations have a standard against which multiple IDSs can be measured. Accordingly, the results are less dependent upon nuances in the network environment.

Third, LARIAT focuses on accurately modeling the real network so that the traffic patterns it produces are representative of those that any network might encounter and tuned to the network of interest. In order to provide a realistic network simulator, LARIAT must emulate three primary network components: users, host servers, and background infrastructure against which users and hosts must compete for resources.

2.3 Windows and WinNTGen Role in LARIAT

A successful model of users on the network should mimic the behavior of real users. On a real network, users are often based on different operating systems. An OS-dependent model is essential because each OS produces different network and usage patterns. For example, a Linux user might habitually produce network traffic through SSH while a Windows user would be far less likely to use SSH. The WinNTGen component of LARIAT provides the testbed with Windows users, and is an important element because Windows constitutes 90% of the market share on client operating systems [11].

WinNTGen generates many types of network traffic producible by Windows NT 4.0, Windows 2000, and Windows XP. It is also extensible enough to accommodate the advent of new applications or services which might produce new types of network traffic. Since most user-generated network traffic is created by user-application interactions, the design of WinNTGen is based on models of human-computer interaction (HCI). For the evaluation of host-based IDSs, it is important that usage is modeled at the user level because the series of events a user performs may be used to detect suspicious behavior or to generate second-order traffic such as automatic server queries. By controlling

applications directly, WinNTGen accurately reproduces the network traffic timings and eccentricities of application implementations.

2.4 Application Use State Machines in WinNTGen

WinNTGen controls Windows-based applications through application use state machines (AUSMs). The implementation of each AUSM controls a virtual user's behavior for a particular application. Each state within a particular AUSM represents an action that the user performs. The transition between a pair of states within the AUSM encodes the likelihood of a particular action sequence. The collection of AUSMs as a whole simulates the full user and is itself a state machine consisting of a central transition state and all of the AUSMs in use. Because each AUSM controls applications through high-level user interactions, application-specific implementation details are safely abstracted from the simulations. This abstraction provides a buffer against any minor changes in application implementation or incremental application upgrades. For example, the Outlook AUSM controls the sending of mail by calling events in Outlook, rather than by constructing a network packet. As another example, the simulated user also avoids using the application's user interface, which changes, sometimes dramatically, from version to version.

Another benefit of simulating users through AUSMs is the ability to extend the user modeling. The modularity afforded by the AUSM implementation allows the simulated usage of new applications to be added to WinNTGen without impeding or affecting existing AUSMs. Furthermore, AUSMs can allow for the modeling of different types of users.

2.5 Outlook AUSM Role in WinNTGen

As mentioned in Chapter 1, the proliferation of viruses over the network is responsible for billions of dollars in lost productivity. However, the statistics for email are even more alarming. The Gartner Group reports that over 95% of viruses are spread via junk email [13]. Information Week estimates the worldwide cost of MyDoom, an email virus which struck in 2004, at $4 billion [13]. Osterman Research asserts that more than 30% of spam is sent from computers that have been infected with a worm or Trojan [13]. The statistics are disquieting because, as yet, email viruses and spam have been relatively benign in comparison to other viruses, which erase or corrupt data.

The large proportion of system vulnerabilities exploited through email motivates LARIAT to identify an appropriate and accurate model for email usage. Because 65% of corporate users manage their email through the Outlook client [14], WinNTGen's email simulations are performed in Outlook, through the Outlook AUSM.

The current implementation of the Outlook AUSM includes functionality to receive and send email messages. Within the send functionality, the user can send email to one or more users, attach one or more files of randomly selected file types, and compose an email body by selecting from a catalog of sample message bodies. Within the receive functionality, the user can view the email message, open attached files when provided, and delete the message.

The transition parameters between the aforementioned states are determined by usage data extrapolated from recorded user event streams in Internet Explorer.

2.6 Terminology and Theory Background

This section provides the background for Windows and Outlook specifications that affect the design of the instrumentation to collect usage data.

2.6.1 *Events*

When a user performs an action in a Windows application, the application generates an event to communicate with the operating system and perform certain processes. Multiple events can be generated each time the user performs an action. Some events are triggered by commands performed through the user interface, such as a button press. Other events are triggered by peripherals communicating with the application, such as the keyboard or the mouse. Still other events are triggered by the application itself, such as the case when an application automatically saves a document. Most of the events generated by the application help the application fulfill a user request and can be mapped to a specific user action. In some cases, these mappings correspond 1:1. For example, when Outlook receives a new mail message, that action corresponds to a single new mail event. In most cases, however, a single action is mapped into a series of events, with each event responsible for performing a small part in fulfilling the ultimate request. For example, when a user sends a mail item after he has finished composing it, Outlook generates five events. First, an event indicates that the send button was clicked. A second event indicates that the composition window is being deactivated, meaning it is no longer the topmost window. Another event indicates that the composition window is closed. When the composition window is closed, usually the next topmost window is the Outlook application UI, which now comes into focus and triggers an activate event. Finally, an item_send event indicates that the message was processed and put onto the outgoing queue. All together, these events perform specific functions to complete the act of sending an email message.

By default, Outlook processes each event to fulfill a request. However, for extensibility, developers are allowed to hook into existing events, such that, when an event is generated, Outlook performs some new specified action in addition to or instead

of the default action. While there are certain limitations to the set of actions that a developer can perform through Outlook events, the event model is extremely powerful. Furthermore, observing a user's event stream makes it possible to determine a set of actions that occurred to generate the event stream.

2.6.2 *Outlook Object Model*

In order to determine how Outlook is being used by observing the event stream, it is important to understand the Outlook object model. The Outlook Object model sets up a hierarchy of objects and object properties, specifies the interactions that are possible between objects, and organizes each object's role in fulfilling the user's email processing needs. Each object generates its own set of associated events. To observe events and make sense of the events that are being generated, it is important to look into the layout of the object model (OM) and understand how to work within the constraints of the Windows framework.

At the topmost level of the Outlook OM is the Application object. The Application is the object which encapsulates the conceptual representation of Outlook. This is the root object which gives access to all other objects, including the Explorer(s), MailItem(s), and Selection collection. The Explorer is the main graphical user interface (GUI) that appears when Outlook is in use. The Explorer controls user interface widgets and consists of several panes, a menu bar, toolbars, and a list of email items. Additionally, the currently selected message item(s), characterized as MailItem objects displayed in the preview pane, have an underlying representation as the Selection collection. Like these, many of the visual GUI cues have underlying object representations. Many of the interesting user actions are performed through the use of the GUI. Therefore, accurately observing Outlook usage through the event stream requires that all the objects that are

displayed or managed by the GUI are hooked to provide visibility into the actions taken by the user through the interface.

2.6.3 *Add-In*

An add-in is a software utility or program that can be added to a primary program. Writing an Outlook add-in program allows the add-in to access Outlook specific elements, such as the objects and event hooks described in Section 2.6. The add-in is written in C# and built upon the Microsoft .NET Framework. The .NET Framework is a component of the Windows operating system, a software development platform designed for rapid application development.

Chapter 3

RESEARCH AND DESIGN GOALS

My primary research goal was to develop state models to describe how Windows users process email using Outlook 2003. I focused on observing three major elements to establish how Outlook is used: 1) the actions that a user performs during regular Outlook email processing, 2) the spectrum of data that a user is exposed to through email, and 3) the way other computing tasks complement email processing.

The actions that a user performs during email processing frame the modeling of email usage by contributing knowledge of the set of likely states a user enters and the state transitions a user makes. For example, knowing that a user typically reads, replies, then sends an email in quick succession holds behavioral significance. Such an action sequence should be modeled accurately if it is observed.

Understanding the kind of data and information a user processes through email enriches the modeling of email usage by augmenting the top-level states with inner substates. For example, a user may perform the action of reading an attachment on 10% of the email he receives. Having this kind of data allows the model to break down the *Attachment_Read* event (see Table 1: Objects and Associated Events for a complete list of events) to determine the specific file types of the attachments that are read.

Finally, understanding how applications are used in conjunction with Outlook, whether to complement, supplement, or supplant existing Outlook functionality enables the building of a real usage model. For example, if a user frequently opens a calendar program after reading a new message, this knowledge provides additional context that could explain why the users perform an action sequence. It may also imply that some subset of messages that are sent and received are used for scheduling purposes.

In terms of modeling users, the way applications are used in conjunction with Outlook influences the likelihood that one action follows another. For example, it would be interesting to observe how often a user switches from Outlook to an internet browser after he completes a burst of message processing. Switching frequently to a browser may indicate that recently read messages contain links to web sites or context which requires information from the internet. While the veracity of such conclusions is uncertain, they nonetheless help to identify a set of probable and plausible likelihoods.

Together, these elements serve as a framework for modeling Outlook usage in LARIAT. The framework provides heuristics for understanding observed transition parameters determined via analysis on real usage data. Understanding which parameters are due to behavioral differences driven by relevant usage elements and which parameters are unrelated to these elements enables LARIAT to build a model of usage that is guided by heuristics rather than by numbers alone. For example, the model should include user actions which are qualitatively relevant and meaningful. The data alone cannot indicate whether an action is meaningful. While there are limitations to the extent that these heuristics conclusively determine ground truth regarding user behavior, they are still useful in a model of behavior.

The purpose of observing the above factors is twofold. For one, the LARIAT simulation needs to be based on a more deterministic model of real-user behavior on a

Windows machine running Outlook along with other standard Windows applications. Second, the model would be more realistic if it simulated a larger set of actions commonly performed in Outlook. Developing a richer model will pave the way for future work to extend the current Windows network traffic generation component of the LARIAT system with a more comprehensive model of Outlook usage.

3.1 Extending the Model of Basic Email Tasks

To extend the current Outlook AUSM model for a more realistic network traffic pattern, I will observe user behavior in the following tasks:

- **Compose New Mail**: The current model selects message recipients and message bodies arbitrarily. I will try to determine if there are any patterns for when a user is more inclined to compose a new message. I will also determine the relative frequency that users perform this action over all other email-related actions.
- **Reply/Reply to All/Forward**: There is no simulation for reply, reply to all, or forward in the existing implementation of the Outlook Module. Through the collection of data, I will determine whether a more realistic model should include such actions.
- **Message Recipients**: Messages can be sent to one or more recipients or lists. Through data analysis, I will try to determine if there are any significant patterns for sending email to one or more users.
- **Message Body**: Within the context of sending and receiving email, different types of message bodies can also be modeled through data analysis to contain: links to web sites, a mix of HTML/Rich Text/Plain Text –formatting, embedded images, etc. to enrich the body composition.

- **Attachments Sent**: There is currently no correlation between the types of files that are attached to outgoing email messages, and the type of person (persona) who sends the message, although we empirically observe that certain personas are more likely to attach files of certain types. For example, a researcher or student may be more likely to send PDF files, a manager may be more likely to send PowerPoint presentation files, etc...
- **Attachments Received**: There is no model for the percentage of received attachments that are viewed. Nor is there a model for what determines the types of attachments that are commonly viewed. For example, a user may be more or less likely to view an attachment immediately if an email contains only one or two attachments. However, if the email contains many attachments, or a zip file, the user may be more inclined to finish processing other emails before diverting to the more time-consuming task of reviewing multiple documents.
- **Launch Related Applications**: If the body of the email message contains links to a web page, the user should be able to follow the links. Also, if the body of the email message contains an email address, the user may want to compose an email to that user instead of replying to the sender of that email. This behavior is currently not modeled by WinNTGen.

3.2 Modeling More Advanced Outlook Features

There are also more advanced Outlook features that have not been explored in the simulation. These include sending or receiving email messages that are digitally signed or encrypted, flagged with importance, access controlled with DRM (digital rights management), configured for read/delivery confirmation receipt, contain voting buttons

(which returns email to the sender when each recipient votes), contain calendar events, expire after a certain time, or are sent on delay.

Data collection and analysis will determine the frequency with which users perform these actions. Once the significance of these actions is determined, then an implementation plan for these additional features can be formulated. Significance is measured by the relative frequency with which such actions occur, as well as the extent to which each action generates additional, pertinent network traffic.

3.3 States

To achieve the goal of creating a richer, more realistic model for email usage, I implemented a tool to collect the event stream generated by Outlook during real usage. With the event stream data from real Windows users using Outlook 2003 in conjunction with other Windows applications, I can determine the higher-level actions a user performs, and also try to find a basis for why the user exhibits specific behaviors. In this section, I describe the high-level user actions I aimed to observe.

3.3.1 *Outlook State Model*

After exploring the many features available to users in Outlook 2003, I discovered a rather large set of actions, all of which generate some kind of network traffic. The set of actions a user performs in email processing is generally expressed in Figure 1, which depicts that a user begins by opening Outlook, and proceeds to create, send, read, or delete messages until he decides to close Outlook.

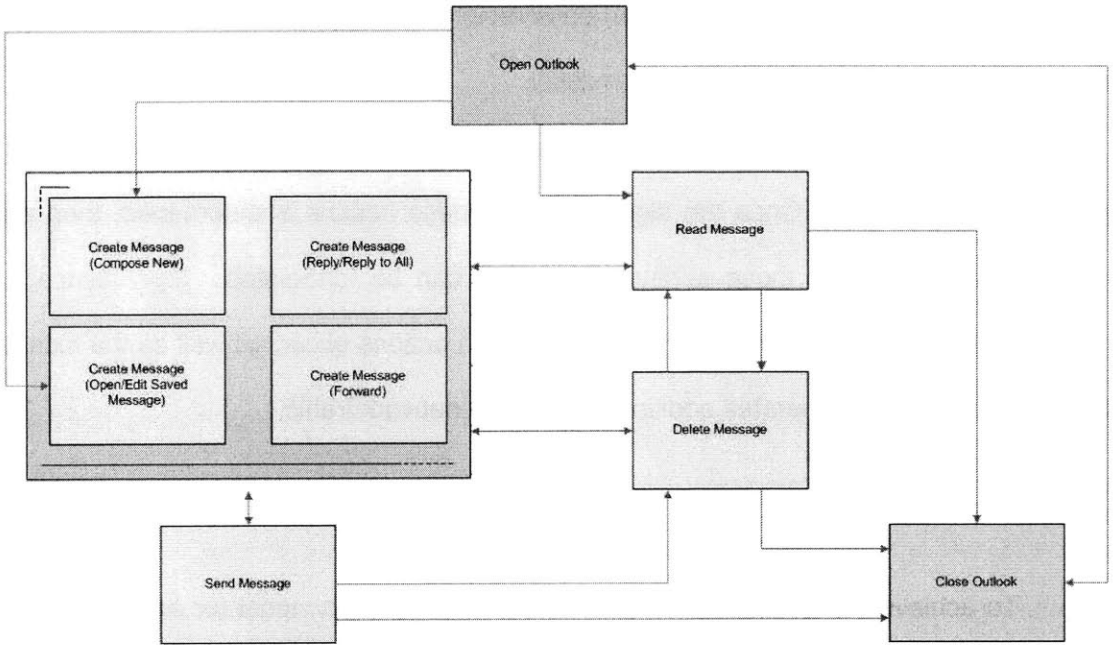

Figure 1: Outlook State Model

For these base states, the graph is nearly fully connected. A few arcs, such as the Open Outlook → Reply/Reply to All/Forward, are not included because, conceptually, a user needs to first read a message before responding to it. The same is true for Open Outlook → Delete Message. Also, notice that the only outgoing arc from the Close Outlook state terminates at the Open Outlook state. When Outlook is closed, no other states can be accessed. Close Outlook is the terminating state.

The state diagram does not include two important states, Leave Outlook, which is a state to represent when the user has temporarily left Outlook to process other information, and Resume Outlook, which is a state for returning to Outlook from another process. This is to maintain clarity in the diagram because all the states in the diagram, with the exception of the terminating state, are fully connected to the Leave Outlook. Similarly, all the states in the diagram are fully connected to Resume Outlook with the exception of the Open Outlook state. Upon returning to Outlook, the user can resume processing in any of the states in the diagram.

3.3.2 *Create Message Substates*

Within the process of creating a message, a user can perform other actions before finally sending the message, and returning to the base states. These actions are categorized as substates of the create message base state, and are depicted in Figure 2. The set of states on the right-hand side are fully linked, because a user can perform each of those actions independently. The column of states connected to the terminating state, Send, differentiate the messages further by the number of message recipients. For the same reason as above, Leave Outlook and Resume Outlook are not depicted.

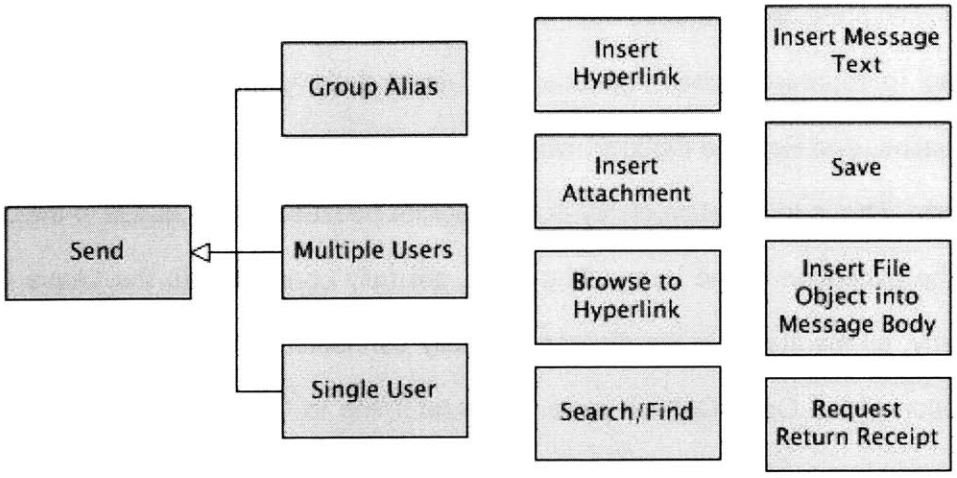

Figure 2: Create Message Substates

3.3.3 *Read Message Substates*

Within the process of reading a mail message, a user can perform more specialized actions. These actions are categorized as substates of the read message state. Figure 3 shows the set of actions a user can perform while in the process of reading a message. When a user reads an attachment, that action can be further classified into opening the attachment or saving the attachment. All of the top level states are fully linked. Some of these substates, such as Open Attachment, takes the user out of the Read state, and into the Leave Outlook state.

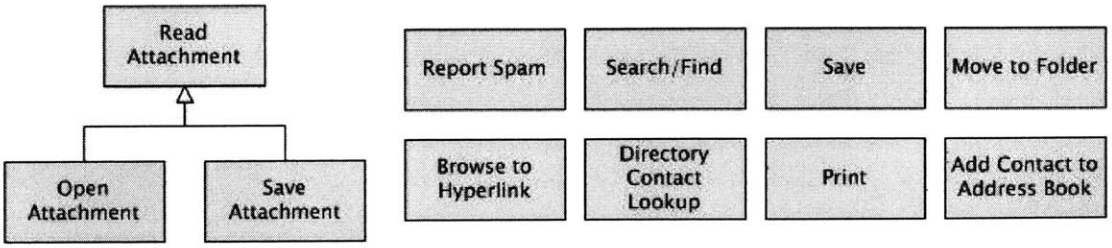

Figure 3: Read Message Substates

Chapter 4

INSTRUMENTATION

In this section, I describe how I collect usage data through the observation of user actions performed on Windows in conjunction with Outlook. To facilitate the discussion, I break up the description into how I collected the events and how I parsed the events into actions. To capture events, I implemented OutlookMonitor. To resolve the event stream into a series of actions, I wrote a parser to read in the entire sequence of events and consolidate events that combine to reveal a high-level user action. I also describe the design architecture, the implementation details, and the format of the output.

The design process can be described by four steps. First, design the tool to collect the raw event data. Next, design the raw data output. Third, design the parser to understand and interpret the output data. And finally, design the output of the parser to facilitate analysis of the final data.

4.1 Collecting Raw Event Stream Data

The OutlookMonitor tool was written in C# to utilize the .NET Framework, which is an environment that provides a notable advance in Windows development. OutlookMonitor is an Outlook program add-in that can be installed by a user on his local

machine. Its function is to listen in on events generated by Outlook 2003. For each event generated, OutlookMonitor automatically logs the event and any additional information related to that event. This section describes the design and implementation of each component class in OutlookMonitor.

4.1.1 *OutlookMonitor: Connect Design & Implementation*

The primary class is Connect. This class allows OutlookMonitor to be an add-in to Outlook, and gets called to perform actions when Outlook is started. This is the first step toward accessing events and components of Outlook. By default, Connect allows the developer to perform certain actions on connection or disconnection from the mail server, on the completion of startup, on the update of add-ins, and on the beginning of shutdown. These are the preliminary events that are open to the developer.

The OutlookMonitor tool begins to hook into more events on startup complete, when Outlook's default initialization procedures have completed. Starting from the topmost level, OutlookMonitor hooks into Application, Explorer, Inspector, Folder, MailItem, button, and keyboard events. On disconnection, it performs a cleanup of the variables that were stored and the events that were hooked into.

The Connect class sets up the infrastructure for the registration of event handlers, the specification of registered event handlers, the extrapolation of useful data related to Outlook objects, and the logging of raw data.

4.1.2 *OutlookMonitor: OutlookEventHandlers Design & Implementation*

When Connect registers itself to listen in on events, it is for the purpose of logging the events and object-specific information. With few exceptions, the OutlookEventHandlers class simply contains all the event handlers that are registered, and specifies that the event should be logged appropriately. It is a minimally

complicated class. It relies on the implementation of OutlookMonitorUtility and OutlookLogger to perform most of the data accessing functions.

One exception is the Deactivate event, which occurs in the Explorer and the Inspector. The Deactivate event occurs when either the Explorer or Inspector windows lost focus. When this occurs, it is an indication the user has temporarily stopped performing email processing tasks. As part of modeling how email processing interleaves with a user's other computational tasks, I sought to determine which processes users generally perform immediately after leaving their email client. Therefore, the Deactivate event creates a thread that attempts to capture the process that is activated immediately after deactivating Outlook, using the GetWindowThreadProcessId event.

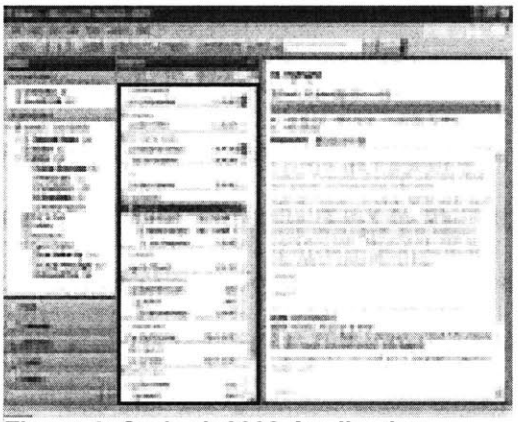

Figure 4: Outlook 2003 Application

The second exception is the SelectionChange event. This event is generated whenever a user selects an item in the Explorer's preview list, outlined in red in Figure 4, of the Outlook application. When a user selects a different item, OutlookMonitor registers item-level event handlers on the selected item and saves a reference to that item so that the item-level event handlers continue to fire even after the SelectionChange method has exited. Without the item reference, registered item-level event handlers are routinely garbage-collected and will cease to fire even though Outlook is still running.

The third exception is the keyPressed event. Outlook provides many keyboard shortcuts for common email-related tasks such as delete, reply, forward, and compose new mail. When the keyboard shortcut is used, the event that usually fires when the

user uses a GUI element, such as a button, does not get triggered. Instead, Outlook fulfills the request along a different code path. Therefore, OutlookMonitor hooks in the keyPressed event. If the keys pressed matches the shortcut for an action that is important to the model of usage, then the event is recorded in the way it would be had the user used the GUI.

4.1.3 *OutlookMonitor: ObjectsWithEvents*

The ObjectsWithEvents class is necessitated by the fact that event handlers are automatically garbage-collected unless at least one reference to the object is retained for the life of the application, until shutdown. Because many objects need to be accounted for, all of the object references are kept in the ObjectsWithEvents class. ObjectsWithEvents is essentially a means to maintain a collection of all the objects to which event handlers are attached. The objects include the Application, all the GUI buttons that fall into the action states outlined in Section 3.3.1 - 3.3.3, the Inspectors collection, the active Explorer, the active Inspector, and a collection for each of the following types of objects: Folders, MailItem, Inspector, and Items.

Table 1 lists the relevant events associated with each type of object. The third column includes a brief explanation for why the event is captured and recorded, and how it helps determine the model for usage.

Table 1: Objects and Associated Events

Object	Event	Usage Context
Application	AdvancedSearch	Indicates that a user is performing an advanced search (programmatically) on a collection of Items.
	NewMailEx	Indicates that the user received new mail.
	ItemSend	Indicates that the user sent mail.
CommandBarButton	Click	Indicates when a button is clicked. Buttons that are

		hooked into include Reply, Print, and Delete.
Inspectors	NewInspector	Indicates when a new Inspector is created, either for message composition, or for opening an Item.
Explorer	Activate	Indicates that the Explorer window is in focus, which implies that the user returns to processing tasks in Outlook.
	Deactivate	Indicates that the Explorer window is defocused, which implies that the user temporarily leaves Outlook.
	BeforeMinimize	Indicates that the Explorer window is minimized, which implies that the user temporarily leaves Outlook.
	FolderSwitch	Indicates when the user has switched to viewing items in a specific folder.
	SelectionChange	Indicates when the user selects a different item to read.
Inspector	Activate	Indicates that the Inspector window is in focus.
	Deactivate	Indicates that the Inspector window is defocused.
	BeforeMinimize	Indicates that the Inspector window is minimized.
	Close	Indicates that the Inspector window is permanently closed.
Folders	FolderAdd	Indicates that a new folder is added to the collection of Folders. Folders are used for sorting mail in Outlook.
	FolderChange	Indicates that the underlying collection of items contained in a Folder is changed – either through addition or removal.
	FolderRemove	Indicates that a Folder is removed from the collection of Folders.
MailItem	AttachmentRead	Indicates that an attachment from a mail item is opened.
	Open	Indicates that an item is

		opened in an Inspector
	Forward	Indicates that the user is forwarding an item.
	Reply	Indicates that the user is replying to an item.
	ReplyAll	Indicates that the user is replying to all recipients of an item.
	PropertyChange	Indicates that a property of the MailItem (i.e. read/unread, importance...), is changed.
Items	ItemAdd	Indicates that an item is added to the Items collection, which represents a Folder's contents.
	ItemChange	Indicates that an item is changed.
	ItemRemove	Indicates that an item is removed.
Keyboard	KeyPressed	Indicates that a key is pressed.

4.1.4 *OutlookMonitor: OutlookMonitorUtility Design & Implementation*

The OutlookMonitorUtility class performs the data retrieval functions for the OutlookMonitor tool. Most importantly, it collects information associated with a MailItem object whenever an item-level event is generated. Given a MailItem, OutlookMonitorUtility determines the following MailItem characteristics: 1) The time the message was sent, 2) The sender, 3) The recipients, 4) The size, 5) The format of the body text, 6) The level of importance, 7) The permission, 8) Any attachments, 9) Any URLs contained within the body, 10) The folder the message is stored in, and 11) the count of the number of unread items in the folder. It assembles each piece of information in a StringCollection. The OutlookLogger then takes the StringCollection and formats the information for output to a file.

Most of the above data elements are properties of a MailItem object. However, to preserve a user's privacy, OutlookMonitorUtility stores the data only as the hashcode. The email addresses of the recipients and senders of the message are hashed. Then, a helper function extrapolates the domain of the email addresses (i.e. ll.mit.edu). The filenames of all attachments are also hashed. Another helper function extrapolates the file extension (i.e. .doc). Finally, OutlookMonitorUtility parses through the body of the mail message to find URLs. When it has found a URL, it determines the protocol (i.e. http) and the filetype (i.e. html).

If OutlookMonitorUtility encounters an Item that is not a MailItem, it returns the type of the Item in a StringCollection. This design is for extensibility. Outlook provides many Item types, such as ContactItem, AppointmentItem, and NoteItem in addition to the MailItem. In the event that future work desires to model users managing other types of tasks in Outlook, the code can be accommodated to that need.

4.2 Logging Raw Data

As described in Section 4.1.4, the data collected for an event is stored in a StringCollection and passed to the OutlookLogger. The OutlookLogger takes the StringCollection and formats the data for output to a file stored on the user's local machine. Each type of event may contain different types of data. In this section, I describe the types of data that are logged and provide a sample output line for each message described in this section and event listed in Table 1.

By default, each message in the output contains the type of message, the category of the data, and the date and time at which the message was recorded. A message is a line in the raw output file. A message can be of three types: Event, Log, or Usage Statistic.

An event is a message that indicates that some action has been performed, either user-driven or programmatically. An event message can pertain to any of the events listed in Table 1. A log message is a message that is unrelated to the usage model. A log message falls into three categories: Error, Comment, or System Message. An error is logged when the OutlookMonitor tool is presented with an unfamiliar situation, or catches an unexpected exception. Errors are used to locate unexpected problems with the tool or output data. A comment is a means for the developer to insert messages in the output, for debugging or clarity purposes. A system message is generally used for messages that are neither errors nor comments. OutlookMonitor typically uses the system message category when it handles an expected exception.

A usage statistic message is a message that yields some data pertaining to usage behavior but does not come directly from an event. This allows the tool to collect additional data even if no events are fired. A usage statistic could have many categories. However, OutlookMonitor only utilizes the usage statistic message for folder statistics.

All of the messages contain the date and time at which the message was logged, with the exception of the Usage Statistic message. The date token contains the day, month, date, and year delimited by spaces. An example of a date token is "Tuesday December 13 2005." The time token contains the one- or two-digit hour (0-23), minute, second, and millisecond, delimited by colons or a period. An example of a time token is "16:51:52.209."

Table 2 provides an explanation of the additional data collected for each kind of message or event that is logged by OutlookMonitor in the raw output. For simplicity, the date and time tokens are removed and replaced by [date] and [time]. The data in a message is comma-delimited.

In Table 2: Data Collected and Sample Output for Messages in Raw Output, the additional data collected often refers to "item information." The item information warrants a more detailed and lengthy description, and is not included in the table for clarity. The item information is a set of data pertaining to the item itself. In Section 4.1.4, I described the kind of data that an item contains. Here, I describe the format of the output of that data. In the example:

> Event,Selection_Change,[date],[time],Thursday October 13 2005,13:34:00.000,
> -1850507293 ll.mit.edu,3,1506349995 ll.mit.edu,-1850507293 ll.mit.edu; 9384729382 mit.edu, none, 3140, olFormatPlain, olImportanceNormal, olUnrestricted,none,none,none,Sent Items,0,619164966

Each token contains a piece of information. Tokens are comma-delimited in the output file. The date (Thursday October 13 2005) and time (13:34:00.000) tokens following the default [date] and [time] entries pertain to when the item was received.

The next token contains two pieces of information relating to the sender. The first piece is a hashcode of the sender's email address (-1850507293). The second piece is the domain of the sender's email address (ll.mit.edu). The next token is an integer that indicates the total number of recipients of the message (3).

The next three tokens each contain the recipient(s) in the "to," "cc," and "bcc" fields of the message, respectively. If there is more than one recipient in any field, the recipients are listed in a semi-colon delimited string. For each recipient, the tool logs the hashcode of the email address, and the domain of the email address. If there are no recipients in any of the fields, the tool logs "none." In the example, there are a total of 3 recipients. The first recipient is a "to" recipient. His email address hashcode is 1506349995, and his domain is ll.mit.edu. The second and third recipients are "cc" recipients. Their semi-colon delimited email address hashcodes and domains are listed as -1850507293 ll.mit.edu;9384729382 mit.edu. Note that one of the email address hashcodes of the "cc" recipient matches the hashcode of the sender. This indicates that

the sender carbon-copied himself on the message. There are no "bcc" recipients, so the token is "none."

The next token indicates the size of the message in bytes (3140). Next, olFormatPlain indicates that the message was sent in plaintext. If the message is sent as HTML, the token would be olFormatHTML. The next token (olImportanceNormal) indicates the message was not sent with any special priority. The next token (olUnrestricted) indicates the message is not permission-restricted.

The next two tokens contain information pertaining to any attachments that are included in the message. If there are no attachments, these token would both read "none" as it does in the example. If there are attachments, the first token would contain a semi-colon delimited list of the hashcodes of the attachment filenames. The second token would contain the file extensions of the files. A sample of these two tokens for a message with two attachments of type txt and doc might look like:

-1349675459;-221345990,txt;doc

The next token contains information pertaining to any URLs that are in the message body, delimited by semi-colons. For each URL, the toll extrapolates the protocol (i.e. https), the file extension (i.e. aspx, or none if the URL is not a reference to a file), and the hashcode of the entire URL. A sample token for a message with two URLs in the body might look like the following:

https aspx -1118270564;http none -1240927154

Following the URL token is the folder in which the item is located (Sent Items). The next number (0) indicates the number of unread messages in the folder. Finally, the last token is a unique identifier for the item (619164966). The unique identifier is calculated using the hashcode of the item's subject line and the hashcode of the time the item was received.

Table 2: Data Collected and Sample Output for Messages in Raw Output

Message	**Additional data collected**
Sample Output	
Error	Exception message
Log,Error,[date],[time],Object reference not set to an instance of an object.	
Comment	Developer comment
Log,Comment,[date],[time],Entering SetInspectors method	
System Message	Exception message – expected
Log,System_Message,[date],[time],Item -721247966 cannot be cast to MailItemClass.	
Usage Statistic	
Folder	Can contain any set of information. It is used in the sample for providing folder statistics for each folder. First, the folder name is listed. The first number following the folder name is the number of total messages in the folder. The second number indicates the number of unread messages in that folder.
Usage_Statistics,Folder,Inbox: 1598 0,Lisa: 94 0,Other Programs: 19 0,Sent: 2062 0	
Event	
AdvancedSearch	Because this event only fires programmatically, no additional data is acquired. OutlookMonitorTool does not handle the Search object that is provided when AdvancedSearch is fired, so it records that it is dealing with an unknown item. This can easily be changed if there is interest in recording data from the Search object.
Event,Advanced_Search_Complete, [date],[time],Unknown Item	
NewMailEx	Contains the item information.
Event,New_Mail,[date],[time],947005624.ll.mit.edu,1,-553432607 ll.mit.edu,none, none,7638,olFormatPlain,olImportanceNormal,olUnrestricted,none,none,http html -1527711690;http none -636078859;http none -636078859,Inbox,1,2120328633	
ItemSend	Contains the item information.
Event,Item_Send,[date],[time],none,1,423889053 harvard.edu,none,none,0,olFormatHTML,olImportanceNormal,olUnrestricted,none,non e,http html 808536231,Inbox,0,-484670472	
Click	Records the button that is clicked, and the item information, see note, pertaining to the currently selected item at the time of the button click.
Event,Button_Click,[date],[time],Reply,[date],[time],-1850507293 ll.mit.edu,1, 416729750 ll.mit.edu,none,none,6187,olFormatPlain,olImportanceNormal, olUnrestricted,none,none,none,Inbox,0,471966902	
NewInspector	No additional information is recorded.
Event,New_Inspector,[date],[time]	
Activate	The event indicates when an Inspector or Explorer is activated. No additional information is recorded.
Event,Inspector_Activate,[date],[time]	
Deactivate	The event indicates when an Inspector or Explorer is deactivated, and the process that is activated as a result.
Event,Explorer_Deactivate,[date],[time],mozilla	

BeforeMinimize	The event indicates when an Inspector or Explorer is minimized. No additional information is recorded.
Event,Explorer_Before_Minimize,[date],[time]	
FolderSwitch	Contains the name of the folder to which the user switches.
Event,Folder_Switch,Tuesday [date],[time],Sent Items	
SelectionChange	Contains the item information.
Event,Selection_Change,[date],[time],[date],[time],164468382 mit.edu,1,-553432607 ll.mit.edu,none,none,12201,olFormatHTML,olImportanceNormal,olUnrestricted,none, none,none,Inbox,1,1886326687	
Close	No additional information is recorded.
Event,Inspector_Close,[date],[time]	
FolderAdd	Contains the name of the folder that is added.
Event,Folder_Add,[date],[time],New Projects	
FolderChange	Contains the name of the folder that is changed.
Event,Folder_Change,[date],[time],lhsu	
FolderRemove	No additional information is recorded. The Folder object is not made available when the event is fired.
Event,Folder_Remove,[date],[time]	
AttachmentRead	Contains the hashcode of the filename and the file extension.
Event,Attachment_Read,[date],[time],2328390,pdf	
Open	No additional information is recorded.
Event,Open,[date],[time]	
Forward	Contains the item information of the open item.
Event,Forward,[date],[time],[date],[time],none,1,157748559 mit.edu,none,none,0, olFormatHTML,olImportanceNormal,olUnrestricted,none,none,none,Inbox,0,254844631	
Reply	Contains the item information of the open item.
Event,Reply,[date],[time],[date],[time],none,1,157748559 mit.edu,none,none,0, olFormatHTML,olImportanceNormal,olUnrestricted,none,none,none,Inbox,0,254844631	
ReplyAll	Contains the item information of the open item.
Event,Reply_to_All,[date],[time],[date],[time],none,1,157748559 mit.edu,none,none,0, olFormatHTML,olImportanceNormal,olUnrestricted,none,none,none,Inbox,0,254844631	
PropertyChange	Contains the name of the property that is changed.
Event,Property_Change,[date],[time],UnRead	
ItemAdd	Contains the folder name to which the item is added.
Event,Item_Add,[date],[time],Inbox	
ItemChange	No additional information is recorded.
Event,Item_Change,[date],[time]	
ItemRemove	No additional information is recorded.
Event,Item_Remove,[date],[time]	
KeyPressed	Specifies the keyboard shortcut that is pressed and the ID of the currently selected item.
Event,Keyboard_Ctrl_Z,[date],[time],1474062654	

4.3 Creating Actions from Event Stream Data

Ideally, an event parser can be designed to programmatically loop through the raw output data and consolidate series of events into user actions. In actuality, three unanticipated implementation flaws introduced complications with parsing. First, the Microsoft API omitted mention of the fact that an event is not guaranteed to fire. Second, the code I implemented did not fully adhere to the Microsoft event-handling guideline stipulating that all event objects should be encased in appropriate object wrappers. Third, events are not guaranteed to fire in a deterministic order.

To evaluate the gravity of the above flaws, I performed a test of the accuracy of the tool for all volunteers participating in the study. Each user was eventually included in the testing phase rather than selecting a representative subset of users because all users behave differently. Therefore, a thorough analysis of each user's behavior was required to ensure that the parser would return accurate results. The first test consisted of three steps. First, the tool was installed on a user's local machine. Second, I manually observed each user's interactions processing email for fifteen minutes, while the tool programmatically recorded the event log. Third, I programmatically parsed the event log and compared the parsed output with the manually recorded, expected results.

The first test found that the code hooked into any particular event is not guaranteed to fire. To address this failure, I added more event support within the data collection tool to provide redundancy in the logging. I then conducted another test to determine the efficacy of providing redundancy. I found several occasions where the original event failed to fire, but where the redundant events fired successfully. After several tests, I was able to match the expected action stream by parsing the recorded event stream.

However, redundancy in event data led to problems in automatic parsing. To manage the parsing, I created a parser written in Perl to evaluate regular expressions and eliminate duplicated or irrelevant events. Then I took the first-pass parsed data and manually parsed the data according to rules that emerged from testing. This was a difficult step, not because the rules were difficult to define, but because the rules were difficult to determine without looking through the events manually to see what users were doing differently and how the events were generated uniquely for each user. This section describes these two steps in parsing the event stream. When parsing is complete, a new file is created that identifies the stream of user actions out of the stream of events. The raw data remains unmodified.

In this section, I describe the programmatic parser and provide the templates, or parsing rules, in Table 3. I also describe the manual parsing and provide the manual parsing rules in Table 4.

4.3.1 *Programmatic Parsing*

The programmatic parser performs the simple parsing of the event stream by extracting the non-redundant events from the raw data. In general, this extraction involves three categories of parsing. The parser: 1) ignores messages that are not a result of user-directed action, 2) removes redundant events, and 3) consolidates series of events that constitute a single user action, maintaining the earliest timestamp for any group of events from the event stream that is consolidated into one user action in the resultant action stream.

There are two kinds of messages that fall into the first category of parsing: messages of any type other than event and messages that are automatically generated by Outlook sub-processes. To eliminate non-event-typed messages, the parser walks through the log file line by line, checks the message type and copies all event-type

messages to an array for further parsing. The parser then continues parsing the array, removing Event messages that are automatically generated by Outlook sub-processes, such as the NewMailEx event, which occurs if Outlook is configured to automatically check for new messages at a pre-specified time interval. Automatic events are logged for statistical purposes only. For example, the NewMailEx event reveals when the inbox receives new mail, which provides relevant statistical, rather than behavioral, information. Unlike the NewMailEx event, which is always an automatic event, some events can either fire automatically or indicate a user-directed action. In most instances, the SelectionChange event fires when the user actively selects a message to read. However, it also fires automatically on the first message in the open mail folder when the user activates Outlook. Because some events such as SelectionChange occasionally fire automatically, the parser is forced to form an unambiguous action stream from an ambiguous event stream. In these non-deterministic cases, the parser always makes the same decision so as not to skew the results with a non-definitive decision rule.

The second category of parsing, removing redundant events, also handles two kinds of messages. First, the simplest kind of redundant event occurs in the case of the keyboard keyPressed event, which fires multiple times successively during the short duration for which key is held down. Only the first instance of any series of keyPressed events is necessary, so the parser removes the subsequent keyPressed events from the array. However, most redundant events do not occur successively. The second kind of redundant parsing is, therefore, a more difficult procedure. Redundant events do not necessarily share the same event name, or occur in a deterministic order. Therefore, the parser checks the event stream against several templates. If a match to a template is found, it removes the events specified by the template.

Consolidating events, the third category of parsing, is also a difficult procedure. The main difference between redundant parsing and consolidation parsing is that the latter

requires a combination of information provided by all the events involved, whereas redundant parsing merely ignores the subsequent redundant messages. Because of the difficulty in determining consolidation heuristics, the programmatic parser performs only simple consolidation. For example, when Outlook is minimized, the BeforeMinimize and Deactivate events both fire, respectively. Both of these events, however, indicate the same action, that the user has temporarily left Outlook. The parser consolidates these events by keeping the Deactivate event name and using the BeforeMinimize timestamp. In the model for usage, Deactivate is then used to represent the leave Outlook state.

Table 3 outlines a detailed specification of the rules and templates that the programmatic parser follows. The first column indicates the category of parsing that is performed. The category (Cat) can be one of the following:

(NE) = Non-Event Parsing (type 1)
(A) = Parsing of Automatic Events (type 1)
(K) = Keyboard Shortcut Parsing (type 2)
(R) = Redundant Parsing (type 2)
(C) = Consolidation Parsing (type 3)

The second column contains the stream of events that appear in the raw data. The third column contains either a description of the output and an explanation for why the parser modifies that particular subset of data or the parser output for the combination of events, whichever is easier to understand. Note that in many cases, parsing is not complete and the output is not ready to be mapped directly to user actions. The final step still requires manual parsing. Also note that the table does not contain the full message, only the tokens of data that are used for parsing. Where possible, the data has been abstracted and shortened for simplicity.

Table 3: Programmatically Parsed Output from Raw Output Event Stream

Cat	Raw Data Event Stream	Parsed Output Event Stream
NE	Log Usage_Statistic	Ignore all of these message types.

A	New_Mail Folder_Change Property_Change Item_Add Item_Change	Ignore all of these event types.
A	Selection_Change,[UniqueID=A] Deactivate Activate Selection_Change,[UniqueID=A]	Ignore the second Selection_Change event for the same item as indicated by UniqueID. It is an automatic response to the user activating Outlook.
A	Selection_Change,[UniqueID=A] Selection_Change,[UniqueID=A]	Successive Selection_Change events are also automatic. The first event is retained.
A	Keyboard_[shortcut=A] Keyboard_[shortcut=A]	Repeated KeyPressed events are ignored. The first event is retained.
A	Folder_Change,[Mailbox=Drafts] Item_Remove	Item_Remove event following a change in the Drafts folder is an automatic response when Outlook sends a message that has been automatically saved.
A	Folder_Change,[Mailbox=Outbox] Item_Remove	Item_Remove event following a change in the Outbox folder indicates when a message on queue to be sent is sent successfully.
A	Folder_Change,[Mailbox=Drafts] Item_Add	Item_Add event following a change in the Drafts folder indicates when a message is automatically saved.
A	Folder_Change,[Mailbox=Sent Items] Item_Add	Item_Add event following a change in the Sent Items folder indicates when a message is added to the queue to be sent.
A	Activate Activate	Neither an Explorer nor an Inspector can conceptually be activated back-to-back without being an intermediary deactivate event. This may be an error in the tool, the Outlook API, or both. The first event is retained.
A	Deactivate Deactivate	Neither an Explorer nor an Inspector can conceptually be deactivated back-to-back without an intermediary activate event. This may be an error in the tool, the Outlook API, or both. The first event is retained.
A	New_Mail ... Item_Add	When a New_Mail message is received, it is added to a folder, and a Item_Add event automatically fires. Ignore this pair of events.

A	New_Mail ... Folder_Change	When a New_Mail message is received, an Item is added to a folder, and the folder is automatically changed. Ignore this pair of events.
A	Reply Inspector_Activate	An Inspector is automatically opened to compose a reply. Ignore the Activate event.
A	ReplyAll Inspector_Activate	An Inspector is automatically opened to compose a reply. Ignore the Activate event.
A	Forward Inspector_Activate	An Inspector is automatically opened to compose a reply. Ignore the Activate event.
A	Explorer_Deactivate Inspector_Activate	When an Inspector is activated, the Explorer must be deactivated. Ignore the Deactivate event.
A	Inspector_Deactivate Explorer_Activate	When the Explorer is activated from the Inspector, the Inspector is automatically deactivated. Ignore the Deactivate event.
A	Click,[Button=Print] Deactivate Activate	The print command pops up a dialog box, which automatically deactivates the window. Dismissing the print dialog automatically re-activates the window. Ignore the Deactivate and Activate events.
A	Click,[Button=Save As] Deactivate Activate	The save as command pops up a dialog box, which automatically deactivates the window. Dismissing the save dialog automatically re-activates the window. Ignore the Deactivate and Activate events.
A	Click,[Button=Purge Deleted Messages] Folder_Change	The purge command purges the mail server of messages marked for deletion and automatically removes the item from the folder. Ignore the Folder_Change event.
A	Reply New_Inspector Open	Reply
A	Reply_All New_Inspector Open	Reply_to_All

A	Forward New_Inspector Open	Forward
A	Click,[Button=Compose New Mail] New_Inspector Open	Compose_New_Mail
A	Click,[Button=New Mail Message] New_Inspector Open	Compose_New_Mail
R	Click,[Button=Reply] Reply	Reply
R	Click,[Button=Reply to All] Reply_All	Reply_to_All
R	Click,[Button=Forward] Forward	Forward

C	Click,[Button=Purge Deleted Messages] ... Item_Remove Item_Remove ...	Count the number of Item_Remove events following a purge event, and append the number to the purge event: Purge_Deleted_Messages,n
C	Folder_Change Item_Remove ...	Purge_Deleted_Messages,n
C	Folder_Switch,[Folder=Inbox]	Folder_Switch_to_Inbox
C	Folder_Switch,[Folder=x]	Any switch to a folder other than the Inbox becomes: Folder_Switch_to_Other
C	Inspector_Deactivate,[date1],[time1] ... Close,[date2],[time2]	Close,[date1],[time1]
C	Deactivate,[process=p]	If p is not the Outlook process, or a process constantly running in the background (i.e. xwin32): Deactivate_to_p
C	Deactivate,[date1],[time1] Disconnect,[date2],[time2]	Disconnect,[date1],[time1]
C	Before_Minimize,[date1],[time1] Deactivate,[date2],[time2]	Deactivate,[date1],[time1]

C	New_Inspector,[date1],[time1] Open ... Item_Send ... Deactivate,[date2],[time2]	The timestamp for an Item_Send event is accurate only to the minute. Replace with the more accurate timestamp of the deactivate event that fires immediately after the message is sent and the Inspector window id about to close: Compose_New_Mail,[date1],[time1] Item_Send,[date2],[time2]
C	New_Inspector,[date1],[time1] Open,[date2],[time2] ...	No Item_Send event indicates that the series of events pertains to a message that is being read rather than composed: Open,[date1],[time1]
K	Keyboard_Delete	Delete
K	Keyboard_Ctrl_D	Delete
K	Keyboard_Ctrl_P	Print
K	Keyboard_Ctrl_E	Search
K	Keyboard_Ctrl_Z	Undo
K	Keyboard_Ctrl_N	Compose_New_Mail
K	Keyboard_Ctrl_S	Save
K	Keyboard_Ctrl_R	Reply
K	Keyboard_Ctrl_F	Forward
K	Keyboard_Ctrl_Shift_R	Reply_to_All
K	Keyboard_Ctrl_Shift_F	Advanced_Find

4.3.2 *Manual Parsing*

The rules followed by the programmatic parser are reliable when the events occur in a specific sequence or when the user performs a very simple series of actions. A simple series of actions may involve the user reading a message, deleting the message, composing a new message, and sending the message before processing any other tasks. However, in reality, users multi-task a great deal. While reading a

message, the user may decide to reply to the message before he has finished reading. He may begin the reply, return to the message for more context, return to the reply, leave Outlook to do some other process, return to Outlook, follow a link from the mail message to a web site, then finally return to complete the reply and send. Templates are much harder to create in this case, because the context that is needed to correctly parse the event stream is indeterminately long.

For these special cases, I manually parsed each user's data. Prior to this manual parsing, it was difficult to generate a complete set of rules that applied for all the users. I found that each user behaved differently and therefore displayed unique traits.

Because the programmatic parser was unable to recognize these unique traits, it would occasionally output an event stream that was obviously incorrect. To understand this problem, consider the case where the correct behavior is to remove a pair of events. However, the parser can only recognize one event out of the pair. It therefore removes the one and leaves the other, which can be attributed to the limitations of the parser. Several manual parsing cases are necessitated by the limitations of the programmatic parser. In retrospect, many of these manual parsing rules could have been parsed automatically. However, at the time of analysis, due to limited time, transfer of the manual parsing rules to the programmatic parser was considered inefficient.

Table 4 outlines a detailed specification of my manual parsing rules. The first column (Event Stream) lists the event stream found in the programmatically parsed output. The second column contains an explanation and/or example of the resulting action stream created after manual parsing.

Table 4: Manually Parsed Output from Programmatically Parsed Output Event Stream

Event Stream	Output Action Stream
Connect Selection_Change,[UniqueID=A] Explorer_Activate Selection_Change,[UniqueID=B]	Selection_Change fires automatically before the Explorer is actually activated because of filters on incoming messages:

	Connect Selection_Change[UniqueID=B]
Attachment_Read,[ext=>processID] Deactivate,[processID] Activate Deactivate,[processID] Activate	Opening an attachment automatically launches the application. In this context, the user is still processing message-related tasks, and should not be categorized as temporarily leaving Outlook: Attachment_Read
New_Inspector Open Explorer_Deactivate Inspector_Activate New_Inspector Open Inspector_Deactivate Inspector_Activate Close Inspector_Activate Close	Opening a message within a open message causes an interleaving of the Close events in the wrong order. The first close operation is tied to the first Inspector_Deactivate, which is replaced accordingly: Open Open Close Inspector_Activate Close
New_Inspector Open Explorer_Deactivate Inspector_Activate New_Inspector Open Inspector_Deactivate Inspector_Activate Close Close	This sequence differs from the previous example, but this difference is due to the same issue: Open Open Close Inspector_Activate Close
Attachment_Read,[ext=msg] Deactivate,[processID=Outlook] Inspector_Activate Close	Attachment_Read on a MailItem, parse out all subsequent events, including the Close: Attachment_Read
Open Deactivate Inspector_Activate Deactivate Inspector_Activate Deactivate Inspector_Activate Deactivate Inspector_Activate ...	Remove all Deactivate and Inspector_Activate pairs. In this case, the context of the event stream prior to the Open event indicates that there are no other open items. Therefore, none of the Deactivate/Inspector_Activate pairs could possibly indicate that another Inspector was focused.
Deactivate,[timestamp=t] Inspector_Activate,[timestamp<t+30ms]	The timestamp this pair of Deactivate/Activate events differs by less than 30 ms. This indicates an automatic, not user-driven, process.
Open Reply	Manually, I maintain a running count of the number of open Inspectors at any given

Item_Send ... Close	time. The number of Close events must match. However, when a message composition is sent, the inspector from which the composition command was received is automatically closed too. Therefore, the Close event cannot belong to the Open event
Open Reply Deactivate Inspector_Activate Deactivate Close Inspector_Activate Item_Send	There are two open Inspectors in this example. Which Inspector is activated at what time? The first activate must belong to the original message. It is being closed. The second activate belongs to the reply message. It is being sent. Open Reply Deactivate Inspector_Activate Close Inspector_Activate Item_Send
BeforeMinimize,[timestamp1] Deactivate,[timestamp2],[processID]	When the Explorer or Inspector is minimized, the programmatic parser fails to extrapolate the processID from the Deactivate event. The parser uses the correct timestamp, which is the timestamp from the minimize event: Deactivate,[timestamp1],[processID]
BeforeMinimize,[timestamp1] Deactivate,[timestamp2],[processID] Explorer_Activate,[timestamp3] Explorer_Activate,[timestamp4]	When the Explorer is activated after it has been is minimized, Outlook generates two Activate events. The programmatic parser fails to parse out the duplicate Activate event. Deactivate,[timestamp1],[processID] Explorer_Activate,[timestamp3]
Deactivate Folder_Switch	This is not possible. The programmatic parser incorrectly parses out an Activate event. Return to the raw data to find the correct event stream. Deactivate Explorer_Activate Folder_Switch
Folder_Change,[Parent Folder!=Inbox] Purge_Deleted_Messages,n	The programmatic parser allows to automatic purging of deleted messages. However, unless the parent folder is the Inbox, the automatic purge is an Outlook artifact, rather than user-driven. Both

	events are manually parsed out.
Compose_New_Mail Deactivate,[processID=Winword] Open	Some users use Word to compose mail. Using word to compose mail should not be construed as leaving Outlook: Compose_New_Mail

4.4 Final Data Output for Analysis

After parsing the raw data, I created several additional data files to facilitate analysis. This section describes how the logged data was parsed into the final action stream.

The data is analyzed to model usage in four ways: 1) a state diagram of user actions and transitions between states, 2) histograms of the timing between most frequent transitions, 3) statistical analysis of the event stream (including the total counts of visits to any state, the breakdown of the types of attachments that are read, the number of days of usage data collected per user, and the number of actions performed daily), and 4) a statistical measure (z-score) of the similarity between users.

4.4.1 *State Diagram*

A state diagram is a graph in which the nodes are the states that a user visits, and the arcs are drawn between nodes that occur consecutively in time. An arc is labeled with the total number of times the user has traveled from the start node to the end node.

To obtain the state diagram from the action stream requires three steps. First, a file of the list of states (separated by the newline character), states.txt, is extracted from the action stream using a function, GetStates, Second, a program, count_xisitions (written by my thesis supervisor, Richard Lippmann), formats states.txt into a dot file,

[username].dot, where the [username] is replaced by each volunteer user. The dot file is formatted so that is can be passed into the dot graphing program by Graphviz, which reads the file, automatically-generates the layout of the directed graph, and outputs the image to a gif file, [username].gif. Figure 5 shows a simple example of the process diagrammatically. Actual results can be found in Figure 8-Figure 10.

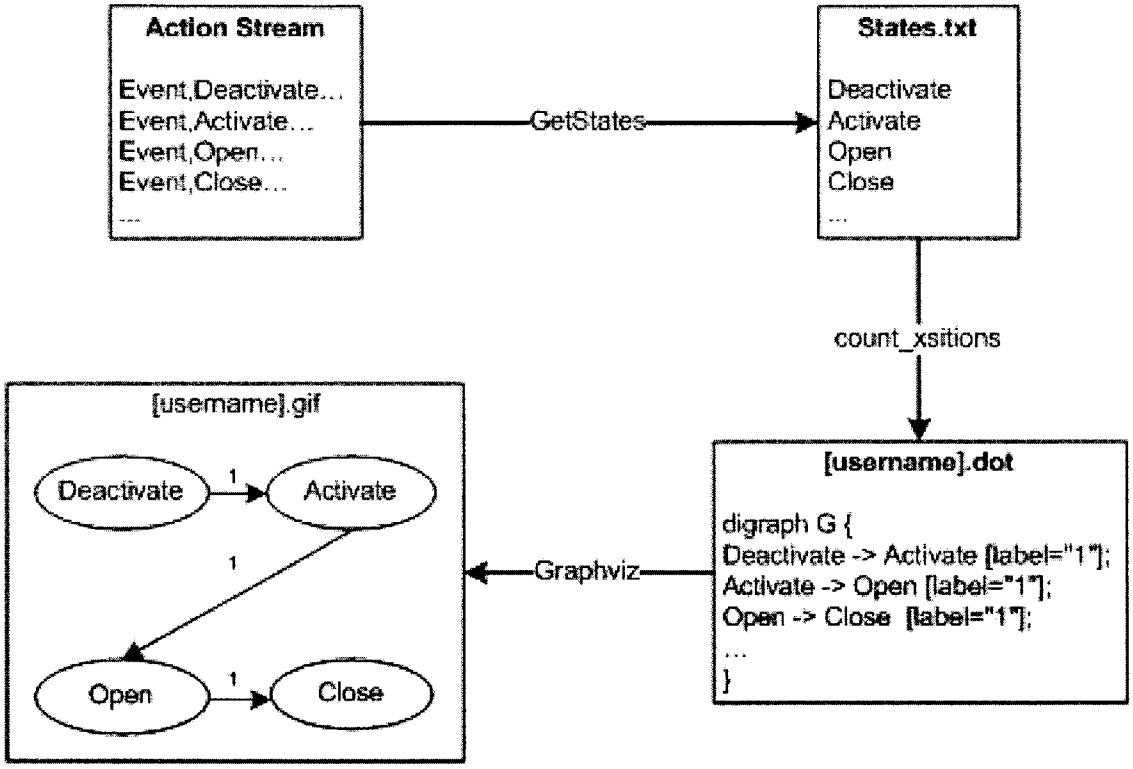

Figure 5: Process of Obtaining State Diagram from Action Stream

The count_xsitions program iterates through the States.txt file and counts up the number of direct transitions from one state (source) to the next (destination). It then sorts each source-destination by the total transition count in descending order. The program, count_xsitions, can also take two parameters to facilitate data filtering. The links parameter records data in the [username].dot file only if the total number of incoming links to a particular node exceeds the number of links specified. The transitions parameter records data in the [username].dot file only if the total count of

transitions for a source-destination entry exceeds the specified number. Both data filtering switches are useful because they allow analysis to focus on more common actions.

The count_xsitions syntax is specified by:

count_xsitions.pl –fin [input filename] –links m –transitions n > [output filename]

If –links and –transitions flags are omitted, or if m and n are set to zero, the output contains the complete data.

The output of count_xsitions resembles the example in [username].dot of Figure 5. Each source-destination is recorded on its own line. The total number of times the user traverses the source-destination arc is the number tagged by the label keyword ([label="1"). The source to destination arc is indicated by the "->" characters.

Graphviz recognizes the syntax:

source_node -> destination_node [label="1"]

by drawing and labeling the source and destination nodes with the node names, and labeling the arc between the nodes with the specified label number.

4.4.2 *Transition Timing*

A histogram of the timing between popular transitions is used to determine the distribution of interstate timing. To obtain the histogram, I first extract the states, dates, and times, then calculate the difference between the timestamps of successive states. For each source-destination, I list the timing of each instance of the transition in a format that can be easily parsed in Matlab. Finally, I use Matlab to plot the histogram of the popular transitions. Figure 6 shows the process diagrammatically.

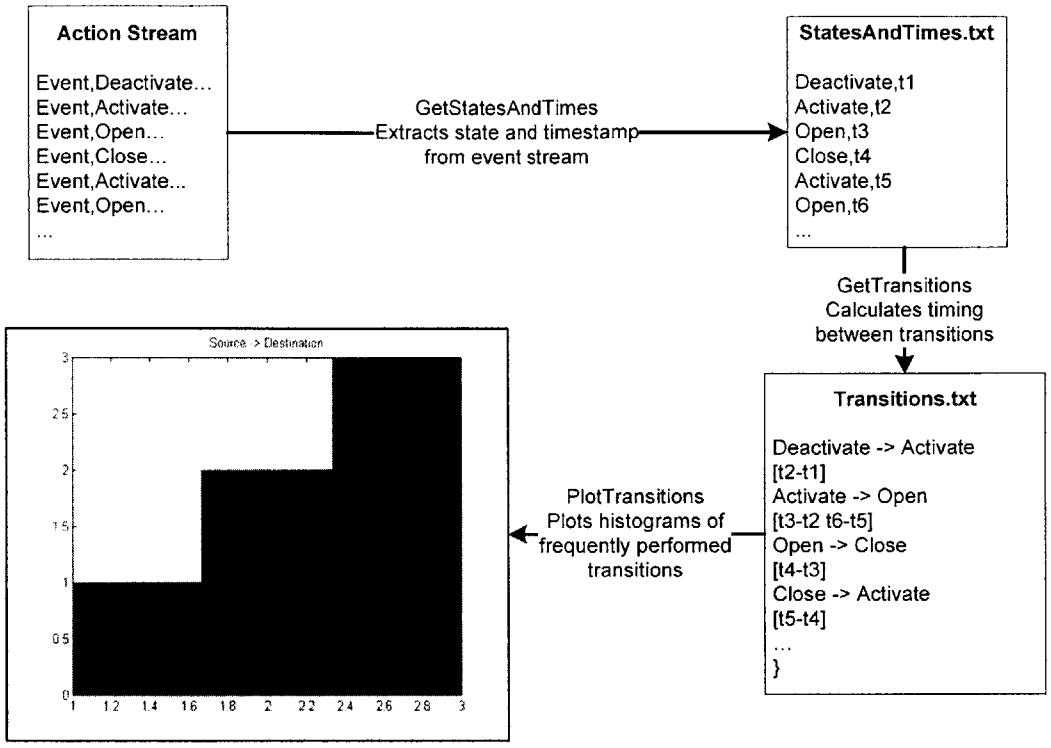

Figure 6: Process of Obtaining Timing Histogram from Action Stream

The PlotTransitions program, written in Matlab, takes the input file (Transitions.txt), and an optional minimum number of transitions. If a minimum number of transitions is entered, PlotTransitions will generate histogram plots for only the arcs between source-destination that have been traversed more than the specified number. PlotTransitions parses Transitions.txt by looking for a line that matches "Source -> Destination" then evaluating the vector string on the next line. It labels the figure with the Transition name. The number of bins used by the histogram can be manually changed.

4.4.3 *Statistical Analysis*

The statistical analysis is easily extracted from charts and pivot tables in Excel. An Excel pivot table performs the totaling of events across a number of axes, including by day and by event. Table 5 provides a sample pivot table. The days included in the

sample data are shown in the first column. The states that are visited are listed in subsequent columns, and tallied by visits per state per day, by total visits per day (Total/day column), and by total visits per state (Total/state row).

Table 5: Sample Pivot Table Used in Statistical Analysis

	Event						
Date	Connect	Disconnect	Explorer_Activate	Folder_Switch	Preview	Print	Total/day
8/31/05	1	1		1		1	4
9/1/05	1				1		2
9/2/05					2		2
9/6/05			1		4		5
Total/state	2	1	1	1	7	1	13

4.4.4 *Z-Score Spreadsheet*

The z-score is a statistical measure of the similarity between different users by comparing their transitions between pairs of states [19]. The significance of the z-score can be determined in five steps. First, count the total number of outgoing arcs from a source node (i) to all destination nodes (all j) for each user (u), $N_i(u) = \sum_j n_{ij}(u)$. Second, calculate the probability, for each user, that a transition occurs from a source node to a destination node, $p_{ij}(u) = n_{ij}(u) / N_i(u)$. Third, calculate the group average transition probability for each source-destination transition by summing across all users, $pg_{ij} = \sum_u n_{ij}(u) / \sum_u N_i(u)$. Fourth, calculate the z-score as $Z_{ij}(u) = [p_{ij}(u) - pg_{ij}(u)] / \sigma_{ij}(u)$. The user's standard deviation is calculated by $\sigma_{ij}(u) = \sqrt{\left[p_{ij}(u)(1 - p_{ij}(u))\right] / N_i(u)}$. Finally, analyze the z-scores based on significance and the measure of similarity. A user's z-score for a particular transition is significant if the user has sufficient data to warrant evaluating his data in that instance for similarity to the group. Of the significant z-scores, if the absolute value of a user's z-

score is less than or equal to 2 (probability of a particular transition falls within 2 standard deviations of the mean), then, based on the assumption of a Gaussian distribution, it can be concluded with 95% confidence that the user behaves similar to the group on average [19]. To facilitate the above calculations, I import the data from all users into Excel. It is important to note that each user has a different but overlapping set of source-destination transitions. To manage the calculations in Excel requires that the transitions which overlap across users are properly aligned in the spreadsheet. For this purpose, I wrote a program in C# to take in any number of user dot files, parse the dot files to align users across matching transitions, and output a tab-delimited table that can be imported into Excel. The table then looks like the sample in Table 6. The first column lists all the unique source-destination transitions. To the right, each user is represented by a column which lists his transition counts. If the user did not perform a particular transition, the cell is filled by zero. A sum across all users is then inserted after all individual users' data is imported in columns with a value $N_u(ij) = \sum_u n_{ij}(u)$.

Table 6: User data aligned across matching transitions in Excel

Source-Destination	$n_{ij}(u)$		$N_u(ij) = \sum_u n_{ij}(u)$
	User1	User2	
Advanced_Find -> Deactivate	0	30	30
Advanced_Find -> Folder_Switch	5	0	5
Advanced_Find -> Open	25	20	45
Advanced_Find -> Preview	10	5	15

The table continues to the right with a total sum down the column for rows which share the same source node, for each unique source node, for each user in the table. The value is $N_i(u) = \sum_j n_{ij}(u)$. Note that this sum places the same value in each row.

Doing so helps simplify the calculations. Next, sum down these columns to obtain $\sum_u N_i(u)$.

Table 6 continued

Source-Destination	$N_i(u) = \sum_j n_{ij}(u)$		$\sum_u N_i(u)$
	User1	User2	
Advanced_Find -> *	40	55	95

Continuing to the right, I add the transition probabilities for each user, $p_{ij}(u) = n_{ij}(u) / N_i(u)$. Note that it is possible to encounter a divide by zero error in this calculation if any user did not visit a source node that was visited by at least one other user. The formula used for Excel avoids the divide by 0 error by checking if the denominator in the calculation is 0, in which case it sets the probability to 0. The group probability is calculated in the next column, $pg_{ij} = \sum_u n_{ij}(u) / \sum_u N_i(u)$.

Table 6 continued

Source-Destination	$p_{ij}(u) = n_{ij}(u) / N_i(u)$		$pg_{ij} = \sum_u n_{ij}(u) / \sum_u N_i(u)$
	User1	User2	
Advanced_Find -> Deactivate	0/40	30/55	30/95
Advanced_Find -> Folder_Switch	5/40	0/55	5/95
Advanced_Find -> Open	25/40	20/55	45/95
Advanced_Find -> Preview	10/40	5/55	15/95

An intermediate step to calculate the z-score for step 4 is to calculate the standard deviation for each user. To ensure that the standard deviation is strictly positive, the formula in Excel actually adds a small ε = 0.0001 to the equation. Again, it is possible for $N_i(u)$ to equal 0, so handle the divide by 0 error by returning 0 for the division operation. Thus the actual equation is $\sigma_{ij}(u) = \sqrt{\varepsilon + \left[p_{ij}(u)(1 - p_{ij}(u)) \right] / N_i(u)}$. Then, the z-score, $Z_{ij}(u)$ is calculated by $Z_{ij}(u) = [p_{ij}(u) - pg_{ij}(u)] / \sigma_{ij}(u)$.

Table 6 continued

Source-Destination	$\sigma_{ij}(u)$		$Z_{ij}(u)$	
	User1	User2	User1	User2
Advanced_Find -> Deactivate	0.003162	0.067215	-90.3508	3.864306
Advanced_Find -> Folder_Switch	0.052387	0.003162	-3.52234	-97.88
Advanced_Find -> Open	0.0766612	0.064941	4.739414	1.566518
Advanced_Find -> Preview	0.068538	0.038893	1.563255	-1.33568

In the final step, the relevance of the z-score and a measure of similarity between a user and the group are calculated. If $N_u(ij)$, the total number of times a transition is traversed for all users, is below a given threshold, then the z-score is calculated on insufficient data and it is therefore inappropriate to use it as a measure of similarity. In the following sample, and for the actual calculations, I use 15 as the cutoff threshold, which was chosen because it permitted the analysis to ignore a large number of rarely traversed links, but preserved enough data to reasonably compare users. For each statistically significant z-score that remains, if $|Z_{ij}(u)| \leq 2$, then the user is considered to be similar to the group. Otherwise, a score that deviates from a low z-score means that the user probability is significantly different from the group. To make the similar, different, and insignificant scores stand out, insignificant z-scores are set to "_" to indicate that a value is not appropriate, scores that indicate similarity are set to 0, and scores that indicate difference are set to 1.

Table 6 continued

Source-Destination	Modified $Z_{ij}(u)$	
	User1	User2
Advanced_Find -> Deactivate	1	1
Advanced_Find -> Folder_Switch	_	_
Advanced_Find -> Open	1	0
Advanced_Find -> Preview	0	0

The sample data indicates that there is not enough usage for the Advanced_Find→Folder_Switch transition. Also, the z-score indicates that User1 and

User2 are more different than they are similar, because their behavior is significantly different for two of the three remaining statistically relevant transitions. User1 and User2 behave similar to one another for only the Advanced_Find→Preview transition.

Chapter 5

RESULTS AND DISCUSSION

In this chapter, I discuss the data, analysis, and findings for my study of Outlook usage. The results are organized into three categories: 1) the initial data analysis, which shows general patterns of usage and ways that usage differs between users; 2) the analysis of characteristics that differentiate the same action (parameters that vary within a state); and 3) the state machine that results from the sequence of actions users perform in message processing.

5.1 Data Collection

Data was collected from seven volunteers from the Information Systems Technology group at MIT Lincoln Laboratory who hold different positions and display variable email processing habits. Each volunteer supplied some amount of data collected over a period of time. Table 7 shows the total amount of data collected, by user, measured by the total number of days represented in the raw data and by the total number of actions performed over the entire collection period. The information in the table was garnered from the pivot table analysis.

Table 7: Amount of Data Collected By User

User	Total Days	Total Actions Performed
User1	27	2177
User2	28	2397
User3	13	2663
User4	37	8455
User5	7	602
User6	35	3384
User7	26	2939
All Users Average	24.7	3231
All Users Total	173	22617

The range of data collected spans from seven to 37 days. In that time frame, 602 to 8455 actions were performed, for a total of 22,617 recorded actions over a cumulative 173 days. The variation in the number of days collected and the total actions performed per user can be attributed to three factors. First, I set up a rolling installation schedule of the OutlookMonitor tool. User4 and User6 were among the first to install the tool and therefore display a higher action count than the other users. User4 performs an especially large number of actions because she opens and closes each mail message she reads, while other users generally view their messages in the preview pane. Second, each user was observed over the span of several weeks, during which time many users were away from their local machines for travel or holiday. On days when the user did not use the Outlook application on the local machine, no data could be recorded. Finally, users have varying email processing needs and responsibilities. User1 and User5 are graduate students, who have a part-time research schedule to balance with coursework and may not spend as much time in the office communicating with regular staff.

5.2 Action Probabilities

General patterns of usage govern the amount of time each user spends while processing email or performing certain actions related to Outlook. The tables in Section 7.1 show, for each user, the breakdown of the total action counts. In addition to these subtotals, the table also calculates the percentage that each action is performed as a function of the total count of actions performed by day, and over the entire data collection period. Table 8 shows a subset of the action statistics for states commonly visited by each user: the percentage of actions that are devoted to reading mail, sending mail, opening an attachment, and deactivating Outlook (leaving Outlook for another application).

Table 8: Average Usage Statistics for Read, Send, Open Attachment, and Deactivate

User	Read %	Send %	Attachment %	Deactivate %
User1	52.64	2.34	1.19	15.57
User2	56.65	5.67	1.38	5.34
User3	38.04	3.38	1.35	19.3
User4	35.01	3.48	0.83	10.27
User5	22.26	5.48	1.00	29.07
User6	45.51	3.58	2.98	15.16
User7	41.07	1.91	0.41	19.63
All Users Average	41.59	3.69	1.30	16.33

User4 and User5 exhibit a lower percentage of read actions. User5's read percentage is lower because this user's deactivate percentage is much higher than the average for the group, indicating that User5 may perform a lot of context switching between mail processing and other computational tasks. User4 also exhibits a lower percentage of read actions, which is a result of pre-screening messages before reading. This behavior was noted during the tool-parser testing phase, when the first volunteers

were observed for 15 minutes to validate the correct functionality of both the OutlookMonitor tool and the parser. The raw data also corroborates the hypothesis that User4 does not read all of the messages that are received, because this user's Inbox always contained a positive count of unread messages.

The action count serves as a primitive measure of what users are doing. It does not indicate the average amount of time each user spent performing those actions. Any conclusions about timing must be garnered from other analysis metrics, such as inter-state timing histograms.

Table 9 shows the probability that each user performs a message composition related task. There are four ways to compose a message: begin a new composition, reply to a message, reply to all users of a message, or forward the message. Generally, users perform message composition actions infrequently. However, this conclusion does not preclude the possibility that users spend a higher percentage of their time composing messages.

Table 9: Average Usage Statistics for Methods of Sending Mail

User	New %	Reply %	Reply to All %	Forward %	Total %
User1	0.64	1.33	0.23	0.14	2.34
User2	1.42	2.00	1.46	0.63	5.51
User3	0.75	0.90	1.84	0.53	4.02
User4	1.05	2.21	0.09	0.19	3.54
User5	2.16	1.33	0.00	0.66	4.15
User6	0.83	1.80	0.71	0.62	3.96
User7	0.78	1.09	0.10	0.17	2.14
All Users Average	1.09	1.52	0.63	0.42	3.66

5.3 Parameters Within the State

Each state in the LARIAT state machine contains underlying data parameters, which are quantitative traits that further define the state this is visited. When LARIAT

executes a state machine, each state is driven by these underlying parameters. For example, the LARIAT state machine might execute the action of sending a message, but that kind of content in the message and the recipient(s) of the message are further specified by the internal parameters.

LARIAT currently selects internal state parameters at random, such as the total number of recipients for a mail message. The ultimate model that is developed from the data analysis should help to inform the state model from actual observed behavior. Of the many internal parameters that can be estimated from data, this section provides information covering attachments and timing.

5.3.1 *Attachments*

In observing the data for attachments, two questions arise. First, what types of attachments are most commonly received? Second, is there any correlation between the attachment file type and the likelihood that the attachment is read? Table 10 shows the breakdown for the types of attachments that are launched from mail messages for each user listed in decreasing order of popularity across all users.

Table 10: Attachment File Types Opened by Each User

Attach Type	User1	User2	User3	User4	User5	User6	User7	Sum
Pdf	5	12	8	5	38	0	2	70
Mail	2	0	11	9	18	1	3	44
Jpg	2	2	1	35	2	0	0	42
Doc	1	10	8	1	9	4	3	36
Ppt	1	0	3	4	14	0	4	26
Txt	0	10	2	0	11	0	0	23
Xls	1	2	0	4	1	0	1	9
Msg	0	0	0	4	0	0	0	4
Gif	0	0	0	4	0	0	0	4

Vcf	0	0	0	0	2	0	0	2
Zip	0	0	0	0	1	0	0	1
Rtf	0	0	0	0	1	0	0	1
Eps	0	0	0	0	1	0	0	1
Tex	0	0	0	0	1	0	0	1
Log	0	0	0	0	1	0	0	1
Wav	1	0	0	0	0	0	0	1
Xml	0	0	0	0	0	1	0	1
Pps	0	0	0	1	0	0	0	1
Psd	0	0	0	1	0	0	0	1
Bmp	0	0	0	1	0	0	0	1
Sum	13	36	33	69	100	6	13	270

The analysis shows that Acrobat PDF files contribute the highest number of attachment read actions. This result is likely because of the academic environment of Lincoln Laboratory, where staff members diseminate papers and other written documents. Notice that for the JPG file type, User4 contributes 35 of the 42 total read actions. This anomaly can be traced back to the raw data, which indicates that User4 received two messages from the same sender, each containing 15 images, for a total of 30 images, 28 of which were JPEGS. User4 viewed each attachment from those two messages.

5.3.2 *Timing Histogram*

There are many ways to infer timing relations in a state machine designed to model usage behavior. Timing can be observed between two consecutively occurring states, or between any two related states in the model. A simple case would be to observe the timing of the Connect to Disconnect transition, which provides a rough measure of a user's average work day if we assume that a user typically logs into his

mail client first thing in the morning, leaves it running throughout the day, and exits the mail client at the end of his day.

A preliminary observation of the timing histograms for the most frequent transitions for each user did not reveal any simple fit of a standard distribution. Figure 7 shows the cumulative histogram for User4 performing the Deactivate to Explorer_Activate transition. This histogram has a very long tail and covers a large range of interarrival times. For short timing intervals, the distribution appears to be exponential, while for the longer time intervals, it appears to follow a Pareto distribution. Similar behavior was observed in the past for the transition timing measured between button clicks for users interfacing with a web browser.

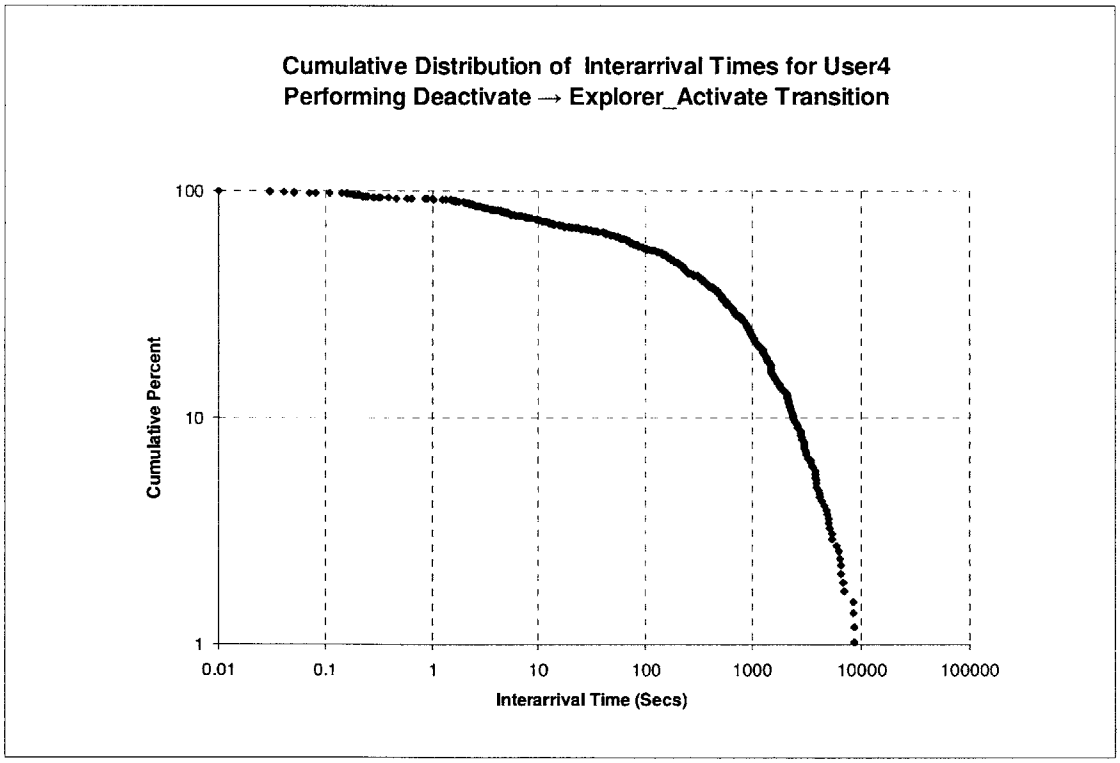

Figure 7: Deactivate → Explorer Activate Transition Timings for User4 Cumulative Histogram

5.4 State Machine

In the model for Outlook usage, there are nearly 30 different states and potentially 30^2 transitions if all the states are fully connected. Fortunately, not all transitions were seen. Aggregating all user transitions, more than 250 arcs were traversed at least once. However, 62 of the arcs were traversed only once among all seven users. By eliminating the transitions which occurred with an aggregate frequency below 15, I created the final state machine with a maximum of 20 common states and 76 transitions. Figure 8 - Figure 10 show the state machines for User2, User4, and User6 who generated the most data.

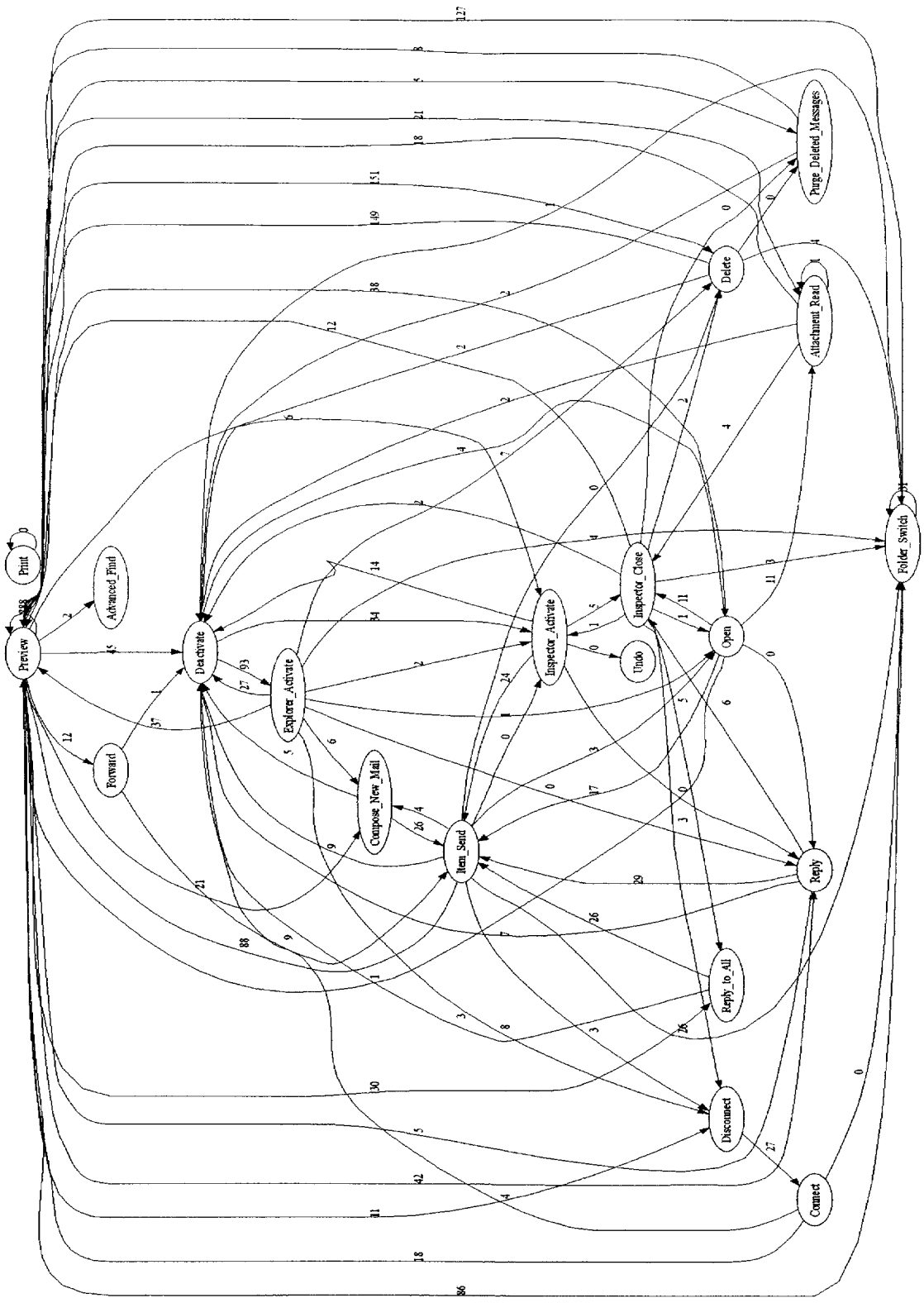

Figure 8: State Machine for User2

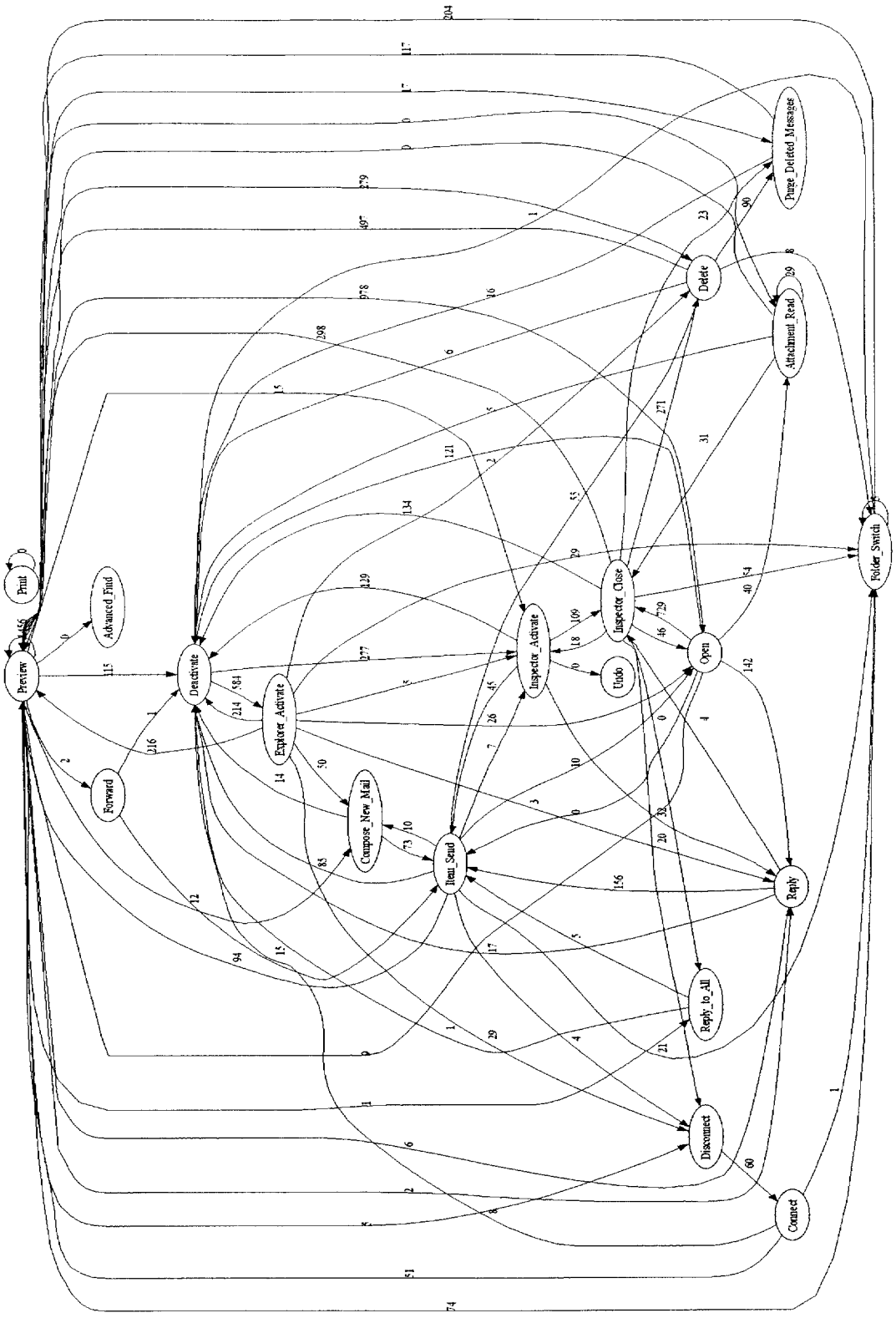

Figure 9: State Machine for User4

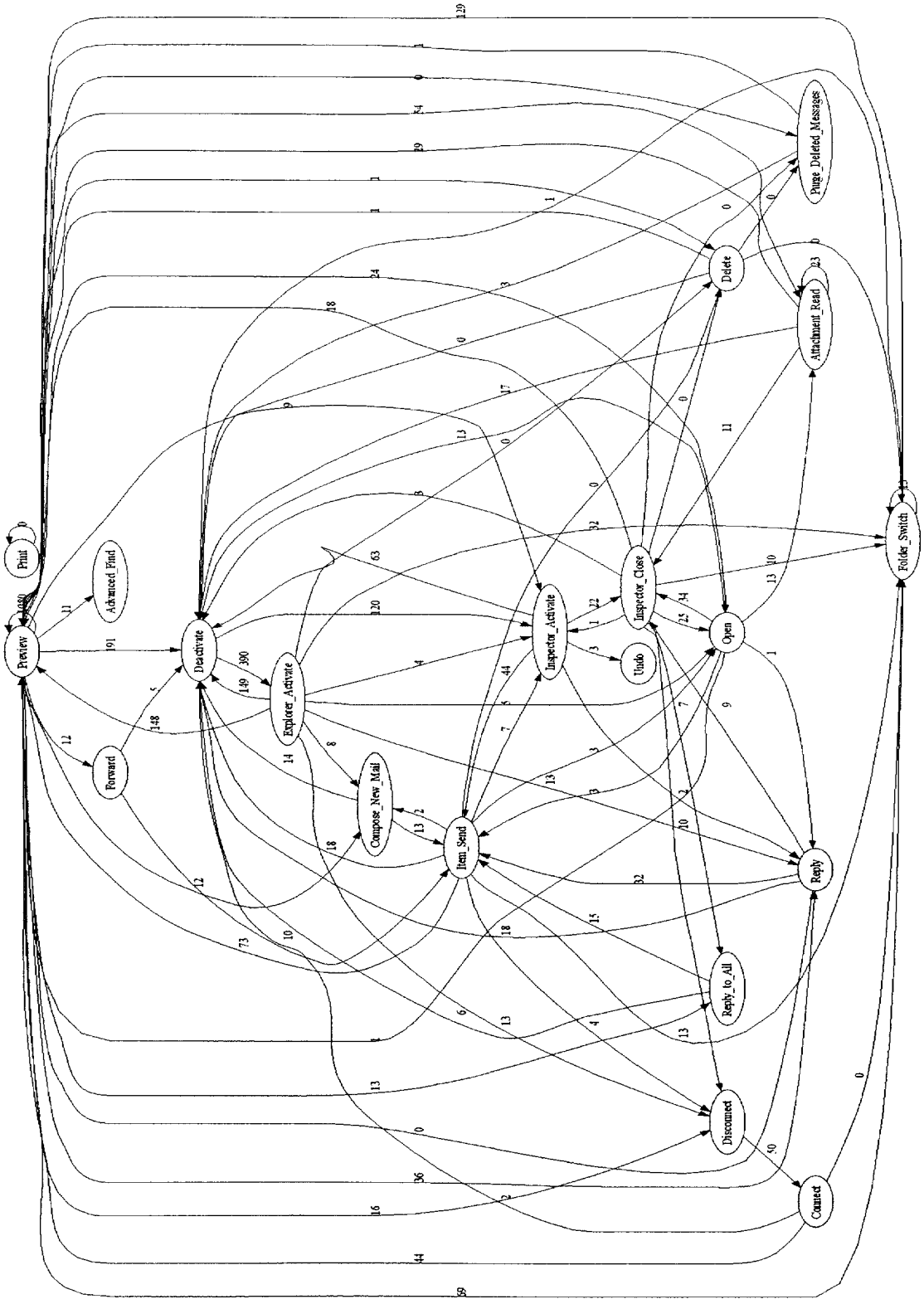

Figure 10: State Machine for User6

Each user's data was mapped to the 20 states and 76 transitions which were observed across all users. However, not all users displayed all the mapped behaviors. For transitions that were not traversed, the arc is drawn and labeled with the value 0. Notice that none of the users pictured in Figure 8-Figure 10 traversed the Print → Print arc. This arc was traversed by only one of the seven volunteers, in excess of the 15 count cutoff. From the state machine diagram, it appears that users behave differently. The z-score analysis, described in the Section 5.5, is used to validate this hypothesis.

5.5 Z-Score Analysis

Table 11 shows the modified z-scores for transitions, calculated as described in Section 4.4.4 from the complete table, which met the aggregate frequency threshold of 15. If there were a sufficient amount of aggregate data, the Transition row is included in Table 11, and shows a value of either 0 or 1 for each user. A score of 0 indicates that a user's behavior for a transition is within 2 standard deviations of the group mean. A score of 1 indicates that the user does not behave like the group.

Table 11: Modified Z-Scores for All Users Across All Transitions

Transition	User1	User2	User3	User4	User5	User6	User7
Attachment_Read -> Attachment_Read	0	1	0	1	0	0	0
Attachment_Read -> Deactivate	0	1	0	1	0	0	0
Attachment_Read -> Inspector_Close	0	0	0	1	0	1	0
Attachment_Read -> Preview	0	1	0	1	1	0	0
Compose_New_Mail -> Deactivate	0	0	0	1	0	1	0
Compose_New_Mail -> Item_Send	0	0	0	1	0	1	1
Connect -> Deactivate	1	0	0	0	1	1	1
Connect -> Folder_Switch	1	1	1	1	0	1	1
Connect -> Preview	1	0	0	1	0	1	0
Deactivate -> Explorer_Activate	1	0	0	1	1	0	1
Deactivate -> Inspector_Activate	1	0	0	1	1	0	1
Delete -> Deactivate	0	0	0	1	0	1	1
Delete -> Folder_Switch	0	0	0	1	1	1	1
Delete -> Preview	1	1	0	0	0	1	0
Delete -> Purge_Deleted_Messages	1	1	1	1	1	1	1

Disconnect -> Connect	0	0	0	0	0	0	0
Explorer_Activate -> Compose_New_Mail	0	0	1	1	0	1	1
Explorer_Activate -> Deactivate	1	1	0	1	1	1	1
Explorer_Activate -> Delete	1	1	0	0	1	1	0
Explorer_Activate -> Disconnect	1	0	1	0	1	0	0
Explorer_Activate -> Folder_Switch	0	1	1	1	1	0	1
Explorer_Activate -> Inspector_Activate	0	0	0	0	1	0	0
Explorer_Activate -> Open	1	0	1	1	0	0	0
Explorer_Activate -> Preview	1	1	0	1	1	1	1
Explorer_Activate -> Reply	0	1	0	1	0	0	0
Folder_Switch -> Deactivate	1	0	0	1	1	0	0
Folder_Switch -> Folder_Switch	1	1	0	1	0	0	0
Folder_Switch -> Preview	1	0	0	1	0	0	0
Forward -> Deactivate	1	1	1	1	1	0	0
Forward -> Item_Send	1	0	1	1	0	0	0
Inspector_Activate -> Deactivate	1	1	1	1	0	0	0
Inspector_Activate -> Inspector_Close	1	0	1	1	1	0	1
Inspector_Activate -> Item_Send	0	1	0	1	0	1	0
Inspector_Activate -> Reply	1	1	1	1	1	1	1
Inspector_Close -> Deactivate	1	0	0	0	0	1	0
Inspector_Close -> Delete	1	1	1	1	1	1	1
Inspector_Close -> Disconnect	1	0	0	1	1	1	0
Inspector_Close -> Folder_Switch	0	0	0	0	1	0	0
Inspector_Close -> Inspector_Activate	1	0	0	0	1	0	0
Inspector_Close -> Open	0	0	1	0	1	1	1
Inspector_Close -> Preview	0	0	0	0	1	1	0
Inspector_Close -> Purge_Deleted_Messages	1	1	0	0	1	1	1
Item_Send -> Compose_New_Mail	1	0	0	0	1	0	1
Item_Send -> Deactivate	1	1	1	1	0	1	0
Item_Send -> Delete	1	1	1	1	1	1	1
Item_Send -> Disconnect	0	0	0	0	0	0	1
Item_Send -> Folder_Switch	0	0	1	1	0	0	1
Item_Send -> Inspector_Activate	1	1	0	0	1	0	0
Item_Send -> Open	1	0	1	0	1	0	0
Item_Send -> Preview	1	1	0	1	0	1	1
Open -> Attachment_Read	1	1	0	1	0	1	0
Open -> Deactivate	0	0	0	0	1	0	0
Open -> Inspector_Close	0	1	1	1	1	1	0
Open -> Item_Send	0	1	0	1	1	0	1
Open -> Reply	1	1	1	0	1	1	1
Preview -> Attachment_Read	0	0	0	1	0	1	0
Preview -> Compose_New_Mail	0	0	0	0	0	0	0

Preview -> Deactivate	0	1	1	1	1	1	1
Preview -> Delete	1	1	1	1	1	1	0
Preview -> Disconnect	0	0	0	0	0	0	0
Preview -> Folder_Switch	1	1	1	1	0	0	0
Preview -> Forward	0	0	0	0	0	0	0
Preview -> Inspector_Activate	0	0	0	0	0	0	0
Preview -> Open	1	1	1	1	1	1	1
Preview -> Preview	1	0	0	1	0	1	1
Preview -> Purge_Deleted_Messages	0	0	0	0	0	0	0
Preview -> Reply	0	1	0	1	0	0	0
Preview -> Reply_to_All	0	1	0	0	0	0	0
Print -> Print	1	1	1	1	1	1	1
Purge_Deleted_Messages -> Deactivate	1	0	0	0	1	0	1
Purge_Deleted_Messages -> Preview	1	0	1	1	1	1	0
Reply -> Deactivate	0	0	0	1	0	1	0
Reply -> Inspector_Close	1	0	0	1	0	0	0
Reply -> Item_Send	0	0	0	1	0	1	1
Reply_to_All -> Deactivate	0	1	0	0	1	0	0
Reply_to_All -> Item_Send	0	0	0	0	1	0	1
Count Statistically Significant Z-Scores	41	34	25	49	39	37	31
Count of Computed Z-Scores	76	76	76	76	76	76	76
5% of Count of Computed Z-Scores	3.8	3.8	3.8	3.8	3.8	3.8	3.8

The last three rows of Table 11 sum up the results of the z-score analysis: 1) the count of statistically significant z-scores; 2) the count of computed z-scores; and 3) 5% of the count of computed z-scores. The count of statistically significant z-scores is a measure of the user's similarity to the group. For each transition, the user is given a value of 0 or 1. A 0 indicates similarity to the group, while a 1 indicates a statistically significant z-score because the user behaves differently from the group average. A higher count of statistically significant z-scores indicates that the user is more dissimilar from the group. Among all the users, User4 displays the most anomalous behavior, with a count of 49, while User3 is most similar, with a count of 25. However, even User3 displays behavior that is markedly different from the group average, because 25 is still a very high count.

The count of computed z-scores measures of the total number of state-to-state transitions for which the group, as a whole, presented "enough" data. Enough, as a measure, is somewhat arbitrary, but is intended to quantify sufficiency of data. I used 15 as the cutoff. If the sum across all users for a particular state-to-state transition was greater than or equal to 15, the transition was included in the final state machine, and each user's z-score for that transition was calculated. This value is 76, meaning that there were 76 transitions for which sufficient data was collected to warrant inclusion into the final state machine.

The 5% of the count of computed z-scores allows for a variation of 5% among users when evaluating finite data. If a user were considered similar to the group, he should display a maximum of 5% variance from the group or alternately, display similar behavior for 95% of the transitions. Therefore, if the users were similar, their count of statistically significant z-scores should be close to 4. The actual computed value is an order of magnitude greater than the expectation.

Further analysis was also performed to determine if any n-subgroup of users displayed similar behavior, where n is the number of users in the subgroup. To do this, each pair of users was compared, calculating a user's z-score based on the average data for each subgroup of two users (2-subgroup). Because User3 differed least from any other single user, User3 was used as a basis for an n-subgroup. Given User3, a user was then added to form an n+1-subgroup and the z-score calculation described in Section 5.5 was performed. It was found that User3 and User6 in a 2-subgroup differed by a maximum of 5 significant z-scores from the group they form. User3 and User7 form a 2-subgroup where each user differs from the group by a maximum of 2 significant z-scores. When User3, User5, User6, and User7 are combined into a 4-subgroup, only one user, User6, differs from the subgroup by more than 10 z-scores. However, no

large subgroups could be found and the results still indicated that each pair of users is statistically different. This analysis corroborates the result that users do not group well.

Therefore, based on user data for significant transitions, the results show that there are fundamental differences between the way users process email in Outlook. Given this finding, the ultimate LARIAT model will implement multiple state machines, each based on an individual user, rather than attempt to combine users based on any similarities. The z-scores calculated for all users corroborate the conclusion that all users considered for analysis behave differently.

5.6 Behavioral Differences

This section describes, for some key transitions, the behavioral habits of users which may have contributed to the statistical differences that were observed. In particular, I discuss the differences related in the usage of Outlook inspectors and in how users manage and sort mail items that have been read.

5.6.1 *Differences with Inspectors*

For the Preview → Open transition, all users display behavior that is different from the group average. This result can be partly attributed to User4, who contributed the vast majority of the total count for this transition, with 978 of 1086. At merely 38 of 1086, User3 contributes the second highest individual user count. Therefore, the group average is greatly skewed by User4 and is not representative of any single user's actions. Since User4 habitually opens all messages for reading, she contributes a disproportionate number of traversal counts to this transition. This user disabled the preview pane, which displays the message body in the Explorer. Only the messages that were opened were actually read. All other users utilized the preview pane, and therefore rarely were compelled to open a message for reading.

Differences displayed in the Inspector_Activate → Inspector_Close and Inspector_Close → Delete transitions are also partially justified by the fact that User4 dominates the data for those transitions, contributing 109 out of 173 and 271 of 276 total traversals respectively. User4 is more likely to perform the Inspector_Activate → Inspector_Close transition for the case when the user performs an email processing related context switch, then returns to the message which triggered the context switch. The complete action series might be: Preview → Open → Deactivate → Inspector_Activate → Inspector_Close → Preview → ... For a similar response, other users' action stream might be: Preview → Deactivate → Explorer_Activate → Preview...

User4 is also more likely to traverse the Inspector_Close → Delete transition, whereas other users traverse the Preview → Delete transition. However, there are other differences involving the delete action, which are further specified in Section 5.6.2.

Yet another difference involving the Inspector actions is the Inspector_Activate → Reply transition. User4 contributes 32 of the total 35 traversals here, dominates the data, and skews the average. Other users demonstrate this behavior in the Preview → Reply transition, where User4 comprises only a small portion of the total traversals, a mere 2 of 155. User2 and User4 are both identified as different from the group, User4 for obvious reasons. User2 contributed the highest count of traversals with 42 of 155 for this transition. This user's count may be higher than the others and than the average because this user, on some occasions, actually intended to reply to all rather than reply only to the sender.

Open → Inspector_Close → Preview is also a notable transition that further reveals the usage differences involving inspectors. This transition for User4 is, in concept, the same as the Preview → Preview action for all other users. When User4 traverses this transition, it is evident that a message is read, closed, and another

message is read in succession. In this case, the users are all performing the same task of reading mail messages successively, but User4 does it in a different way.

In all the above cases, the act of reading a message, and performing functions related to the message such as delete, reply, activate, and deactivate are obscured by the action of opening the message to read in its own inspector window.

The differences between each of the users with respect to the Inspector object became somewhat less pronounced when User4's contribution is removed for those transitions involving inspectors, as indicated by two changes to the results. First, the z-scores based on a group excluding User4 decrease by an order of magnitude. Second, the total count of statistically significant z-scores decreases. However, despite these results, the conclusion remains unchanged. Users still differ from one another by more than two standard deviations. Therefore, while User4's data causes an exaggeration of the differences that are observed, User4 is not solely responsible for them.

5.6.2 *Differences with Delete*

In addition to the difference regarding the delete state mentioned in Section 5.6.1, another difference involving the delete state and the Preview → Delete transition can be attributed to User6. User6 does not delete messages, choosing instead to file all messages into folders. This user contributes 1 of the total 542 traversals for this transition.

The Delete → Purge_Deleted_Messages transition is another behavior wholly dominated by User4. No other user traverses this arc, which indicates that User4 is very mindful of mailbox clean up and that the delete action more often than not evokes this user's habit of regularly purging messages marked for deletion. Adding to this difference in observed behavior is the fact that, for some users, the Purge_Deleted_Messages

action is an automatic, scheduled Outlook task, rather than a deliberate, driven response.

5.6.3 *Other Differences*

The computed z-scores indicate a number of other differences between users that no observed behavior readily explains. For example, all but User3 displayed behavior different from the group for the Explorer_Activate → Deactivate transition. This result indicates that users have a unique way of interleaving email tasks with other computing processes.

The Print → Print transition is another transition which yields high z-scores for all users. This result may be considered an anomaly because the data for this transition comes from User3 exclusively. No other user repeatedly printed the same message. This transition was included in 76 transitions of the final state machine because User3 performed the action 17 times, which is above the 15-traversal cutoff.

Chapter 6

CONCLUSION

The 1998 and 1999 DARPA offline ID evaluations revealed that there was a pressing need for a systematic and scalable approach to testing ID products designed to secure the information sent or accessible across the network. LARIAT was developed in response to the DARPA findings, and Jesse Rabek's thesis work enabled LARIAT to simulate network traffic that models users generating traffic while performing tasks in a number of applications from a Windows machine. Two key goals for future work were to improve the AUSMs for WinNTGen to include the simulation of more complex user actions and to create a model derived from real usage data for the Outlook AUSM. This thesis builds on Rabek's work by addressing the future work he specifies.

Email processing contributes an important feature to network usage for two reasons. First, email is fast becoming a ubiquitous and heavily used medium for communication. Second, email serves as a powerful vehicle for spreading viruses through attachments or auto-executing programs, attacking information through identity theft, and wasting resources through spam. Therefore, an accurate characterization of email network traffic is an important goal of LARIAT.

To accurately characterize email network traffic, data from real users is collected and analyzed to produce a more realistic usage behavior model. To collect data, I

developed a tool called OutlookMonitor, which is an Outlook add-in built on the Microsoft .NET Framework and installed on a user's local machine. The tool collected usage data for seven volunteers over variable periods of time, from 7 to 37 days. The tool logged the event stream, generated by Outlook usage, in a file stored on the user's local machine. 24 primary types of events were hooked.

The data was prepared for analysis by parsing the event stream into an action stream. The parsing was done twice, first automatically by a parser written in Perl, which executed a series of well-defined and simple rules, including the removal of redundant events; then manually by stepping through each individual log file to parse the cases that were left undefined by the programmatic parser, including more complex transmutation of the data. The parsing state enables the final model of Outlook usage to describe user actions, rather than expose the implementation-specific details underlying Outlook usage.

The action stream for each user was analyzed across three axes: the extent to which users behave similarly to the group average, the states and transitions that were visited in regular usage, and statistical internal state parameters. The z-score calculation was the analysis tool used to measure similarity. R. Lippmann's Perl program was used to create a state machine for each user. The Excel pivot table was used to determine the internal state parameters for the Attachment_Read action. Additionally, I wrote a Matlab program to plot the timing for any transition, although a cursory look at the plots did not reveal any information that could enhance the model of usage.

Because more data was collected than was analyzed, one of the future goals for this project is to analyze the data across more internal parameters. For example, the probability distribution for the different numbers of recipients to whom a message is sent could be an informative internal state parameter for the Item_Send action.

Originally, the goals for this project also included the actual implementation of the improved Outlook model in LARIAT, but problems encountered during the development phase of OutlookMonitor precluded the realization of this goal. For future work, the Outlook AUSM developed in this thesis is to be added to LARIAT's WinNTGen.

The analysis determined that each user behaves differently. Therefore, a state machine of the 20 prevailing user actions, out of 30 possible actions, and the 76 prevailing transitions, out of a possible 250, was created for each user, to model each user separately. Also, the probabilities that different types of files are opened, such as PDFs or JPEGs, within the Attachment_Read state were calculated as an internal parameter. Together, these results improve upon the original Outlook AUSM with more accurate transition parameters and more complex, more realistic usage behaviors based on actual observed behavior.

Chapter 7

APPENDIX

7.1 Appendix A: Pivot Table Action Counts

This appendix contains the pivot table analysis that breaks down, for each user, the total action counts and the percentage that each action is performed as a function of the total count of actions performed per day and over the course of the entire data collection period.

7.1.1 *User1 Pivot Table*

Date	Data	Attachment_Read	Compose_New_Mail	Connect	Deactivate	Delete	Disconnect
10/26/2005	Count of Event		4	2	17		2
	Percentage	0.00%	6.45%	3.23%	27.42%	0.00%	3.23%
10/28/2005	Count of Event	1	1	1	7		1
	Percentage	2.38%	2.38%	2.38%	16.67%	0.00%	2.38%
10/29/2005	Count of Event			1	18		
	Percentage	0.00%	0.00%	2.08%	37.50%	0.00%	0.00%
10/30/2005	Count of Event	1		1	6		2
	Percentage	2.94%	0.00%	2.94%	17.65%	0.00%	5.88%
10/31/2005	Count of Event	2		1	5		1
	Percentage	6.45%	0.00%	3.23%	16.13%	0.00%	3.23%
11/2/2005	Count of Event			2	9		2
	Percentage	0.00%	0.00%	1.27%	5.73%	0.00%	1.27%
11/4/2005	Count of Event	1		1	20		1
	Percentage	1.00%	0.00%	1.00%	20.00%	0.00%	1.00%
11/5/2005	Count of Event			1	1	1	1
	Percentage	0.00%	0.00%	5.00%	5.00%	5.00%	5.00%

11/7/2005	Count of Event		1	3	14		3
	Percentage	0.00%	1.37%	4.11%	19.18%	0.00%	4.11%
11/9/2005	Count of Event	1	1	4	40	2	4
	Percentage	0.60%	0.60%	2.38%	23.81%	1.19%	2.38%
11/11/2005	Count of Event	6		1	10		1
	Percentage	10.91%	0.00%	1.82%	18.18%	0.00%	1.82%
11/14/2005	Count of Event	1		1	16		1
	Percentage	1.39%	0.00%	1.39%	22.22%	0.00%	1.39%
11/16/2005	Count of Event	1	1	1	29		1
	Percentage	1.10%	1.10%	1.10%	31.87%	0.00%	1.10%
11/18/2005	Count of Event	2	1	1	15		1
	Percentage	1.83%	0.92%	0.92%	13.76%	0.00%	0.92%
11/22/2005	Count of Event	1		1	3		1
	Percentage	4.55%	0.00%	4.55%	13.64%	0.00%	4.55%
11/30/2005	Count of Event	1	1	1	18		1
	Percentage	0.67%	0.67%	0.67%	12.00%	0.00%	0.67%
12/2/2005	Count of Event	2		3	18		3
	Percentage	2.15%	0.00%	3.23%	19.35%	0.00%	3.23%
12/3/2005	Count of Event			1	2		1
	Percentage	0.00%	0.00%	6.67%	13.33%	0.00%	6.67%
12/4/2005	Count of Event			1	1		1
	Percentage	0.00%	0.00%	7.69%	7.69%	0.00%	7.69%
12/5/2005	Count of Event		1	2	11	1	2
	Percentage	0.00%	1.33%	2.67%	14.67%	1.33%	2.67%
12/8/2005	Count of Event	1		1	1		1
	Percentage	1.28%	0.00%	1.28%	1.28%	0.00%	1.28%
12/9/2005	Count of Event	4	3	1	38	1	1
	Percentage	1.93%	1.45%	0.48%	18.36%	0.48%	0.48%
12/10/2005	Count of Event			1	10		
	Percentage	0.00%	0.00%	0.31%	3.13%	0.00%	0.00%
12/11/2005	Count of Event			1			2
	Percentage	0.00%	0.00%	11.11%	0.00%	0.00%	22.22%
12/12/2005	Count of Event	1		1	25		1
	Percentage	1.00%	0.00%	1.00%	25.00%	0.00%	1.00%
12/13/2005	Count of Event			1	2		1
	Percentage	0.00%	0.00%	11.11%	22.22%	0.00%	11.11%
12/14/2005	Count of Event			1	3		
	Percentage	0.00%	0.00%	4.00%	12.00%	0.00%	0.00%
Total Count of Event		26	14	37	339	5	36
Total Percentage		1.19%	0.64%	1.70%	15.57%	0.23%	1.65%

Date	Data	Explorer_Activate	Folder_Switch	Forward	Inspector_Activate	Inspector_Close
10/26/2005	Count of Event	17	7			1
	Percentage	27.42%	11.29%	0.00%	0.00%	1.61%
10/28/2005	Count of Event	6	3		1	
	Percentage	14.29%	7.14%	0.00%	2.38%	0.00%
10/29/2005	Count of Event	1	2		16	
	Percentage	2.08%	4.17%	0.00%	33.33%	0.00%

10/30/2005	Count of Event	7	2			
	Percentage	20.59%	5.88%	0.00%	0.00%	0.00%
10/31/2005	Count of Event	5	3			
	Percentage	16.13%	9.68%	0.00%	0.00%	0.00%
11/2/2005	Count of Event	9	5			1
	Percentage	5.73%	3.18%	0.00%	0.00%	0.64%
11/4/2005	Count of Event	17	7	1	3	1
	Percentage	17.00%	7.00%	1.00%	3.00%	1.00%
11/5/2005	Count of Event	1	4			
	Percentage	5.00%	20.00%	0.00%	0.00%	0.00%
11/7/2005	Count of Event	14	8			
	Percentage	19.18%	10.96%	0.00%	0.00%	0.00%
11/9/2005	Count of Event	35	11		6	1
	Percentage	20.83%	6.55%	0.00%	3.57%	0.60%
11/11/2005	Count of Event	9	5		1	
	Percentage	16.36%	9.09%	0.00%	1.82%	0.00%
11/14/2005	Count of Event	16	2			
	Percentage	22.22%	2.78%	0.00%	0.00%	0.00%
11/16/2005	Count of Event	27	4		2	
	Percentage	29.67%	4.40%	0.00%	2.20%	0.00%
11/18/2005	Count of Event	15	7			
	Percentage	13.76%	6.42%	0.00%	0.00%	0.00%
11/22/2005	Count of Event	3	3			
	Percentage	13.64%	13.64%	0.00%	0.00%	0.00%
11/30/2005	Count of Event	18	6			
	Percentage	12.00%	4.00%	0.00%	0.00%	0.00%
12/2/2005	Count of Event	18	8			2
	Percentage	19.35%	8.60%	0.00%	0.00%	2.15%
12/3/2005	Count of Event	2	2			
	Percentage	13.33%	13.33%	0.00%	0.00%	0.00%
12/4/2005	Count of Event	1	4			
	Percentage	7.69%	30.77%	0.00%	0.00%	0.00%
12/5/2005	Count of Event	11	11	2		
	Percentage	14.67%	14.67%	2.67%	0.00%	0.00%
12/8/2005	Count of Event	1	2			3
	Percentage	1.28%	2.56%	0.00%	0.00%	3.85%
12/9/2005	Count of Event	32	5		6	
	Percentage	15.46%	2.42%	0.00%	2.90%	0.00%
12/10/2005	Count of Event	2	2		7	
	Percentage	0.63%	0.63%	0.00%	2.19%	0.00%
12/11/2005	Count of Event	1	2			
	Percentage	11.11%	22.22%	0.00%	0.00%	0.00%
12/12/2005	Count of Event	15	8		10	
	Percentage	15.00%	8.00%	0.00%	10.00%	0.00%
12/13/2005	Count of Event	2	2			
	Percentage	22.22%	22.22%	0.00%	0.00%	0.00%
12/14/2005	Count of Event	3	2			
	Percentage	12.00%	8.00%	0.00%	0.00%	0.00%
Total Count of Event		288	127	3	52	9

Total Percentage	13.23%	5.83%	0.14%	2.39%	0.41%

Date	Data	Item_Send	Mark_as_Read	Open	Preview	Reply	Reply_to_All	Grand Total
10/26/2005	Count of Event	3			9			62
	Percentage	4.84%	0.00%	0.00%	14.52%	0.00%	0.00%	100.00%
10/28/2005	Count of Event	1			20			42
	Percentage	2.38%	0.00%	0.00%	47.62%	0.00%	0.00%	100.00%
10/29/2005	Count of Event	1			8	1		48
	Percentage	2.08%	0.00%	0.00%	16.67%	2.08%	0.00%	100.00%
10/30/2005	Count of Event				15			34
	Percentage	0.00%	0.00%	0.00%	44.12%	0.00%	0.00%	100.00%
10/31/2005	Count of Event				14			31
	Percentage	0.00%	0.00%	0.00%	45.16%	0.00%	0.00%	100.00%
11/2/2005	Count of Event	2	1	2	124			157
	Percentage	1.27%	0.64%	1.27%	78.98%	0.00%	0.00%	100.00%
11/4/2005	Count of Event	5		1	38	4		100
	Percentage	5.00%	0.00%	1.00%	38.00%	4.00%	0.00%	100.00%
11/5/2005	Count of Event				11			20
	Percentage	0.00%	0.00%	0.00%	55.00%	0.00%	0.00%	100.00%
11/7/2005	Count of Event	1			29			73
	Percentage	1.37%	0.00%	0.00%	39.73%	0.00%	0.00%	100.00%
11/9/2005	Count of Event	7		1	49	6		168
	Percentage	4.17%	0.00%	0.60%	29.17%	3.57%	0.00%	100.00%
11/11/2005	Count of Event	1			20	1		55
	Percentage	1.82%	0.00%	0.00%	36.36%	1.82%	0.00%	100.00%
11/14/2005	Count of Event	1			33	1		72
	Percentage	1.39%	0.00%	0.00%	45.83%	1.39%	0.00%	100.00%
11/16/2005	Count of Event	2			22	1		91
	Percentage	2.20%	0.00%	0.00%	24.18%	1.10%	0.00%	100.00%
11/18/2005	Count of Event	1			66			109
	Percentage	0.92%	0.00%	0.00%	60.55%	0.00%	0.00%	100.00%
11/22/2005	Count of Event				10			22
	Percentage	0.00%	0.00%	0.00%	45.45%	0.00%	0.00%	100.00%
11/30/2005	Count of Event	1			103			150
	Percentage	0.67%	0.00%	0.00%	68.67%	0.00%	0.00%	100.00%
12/2/2005	Count of Event	3		2	31	3		93
	Percentage	3.23%	0.00%	2.15%	33.33%	3.23%	0.00%	100.00%
12/3/2005	Count of Event				7			15
	Percentage	0.00%	0.00%	0.00%	46.67%	0.00%	0.00%	100.00%
12/4/2005	Count of Event				5			13
	Percentage	0.00%	0.00%	0.00%	38.46%	0.00%	0.00%	100.00%
12/5/2005	Count of Event	4			29	1		75
	Percentage	5.33%	0.00%	0.00%	38.67%	1.33%	0.00%	100.00%
12/8/2005	Count of Event	1		3	63		1	78
	Percentage	1.28%	0.00%	3.85%	80.77%	0.00%	1.28%	100.00%
12/9/2005	Count of Event	11			97	5	3	207
	Percentage	5.31%	0.00%	0.00%	46.86%	2.42%	1.45%	100.00%
12/10/2005	Count of Event	3			291	2	1	319

	Percentage	0.94%	0.00%	0.00%	91.22%	0.63%	0.31%	100.00%
12/11/2005	Count of Event				3			9
	Percentage	0.00%	0.00%	0.00%	33.33%	0.00%	0.00%	100.00%
12/12/2005	Count of Event	2			35	2		100
	Percentage	2.00%	0.00%	0.00%	35.00%	2.00%	0.00%	100.00%
12/13/2005	Count of Event				1			9
	Percentage	0.00%	0.00%	0.00%	11.11%	0.00%	0.00%	100.00%
12/14/2005	Count of Event	1			13	2		25
	Percentage	4.00%	0.00%	0.00%	52.00%	8.00%	0.00%	100.00%
Total Count of Event		51	1	9	1146	29	5	2177
Total Percentage		2.34%	0.05%	0.41%	52.64%	1.33%	0.23%	100.00%
					Total Days Data Collected			27

7.1.2 *User2 Pivot Table*

Date	Data	Advanced_Find	Attachment_Read	Compose_New_Mail	Connect	Deactivate	Delete
9/26/2005	Count of Event				1	2	
	Percentage	0.00%	0.00%	0.00%	8.33%	16.67%	0.00%
9/27/2005	Count of Event						4
	Percentage	0.00%	0.00%	0.00%	0.00%	0.00%	9.76%
9/28/2005	Count of Event		2	1		8	3
	Percentage	0.00%	1.89%	0.94%	0.00%	7.55%	2.83%
9/29/2005	Count of Event			1	2	10	1
	Percentage	0.00%	0.00%	1.08%	2.15%	10.75%	1.08%
9/30/2005	Count of Event		14	7	1	22	1
	Percentage	0.00%	4.70%	2.35%	0.34%	7.38%	0.34%
10/5/2005	Count of Event	2	4				20
	Percentage	1.09%	2.17%	0.00%	0.00%	0.00%	10.87%
10/6/2005	Count of Event		2	3	2	8	2
	Percentage	0.00%	1.50%	2.26%	1.50%	6.02%	1.50%
10/7/2005	Count of Event		1	1	1	4	7
	Percentage	0.00%	1.27%	1.27%	1.27%	5.06%	8.86%
10/17/2005	Count of Event		1	4			23
	Percentage	0.00%	0.72%	2.88%	0.00%	0.00%	16.55%
10/18/2005	Count of Event		3			2	2
	Percentage	0.00%	3.37%	0.00%	0.00%	2.25%	2.25%
10/19/2005	Count of Event		2	2			
	Percentage	0.00%	6.67%	6.67%	0.00%	0.00%	0.00%
10/20/2005	Count of Event			1		7	9
	Percentage	0.00%	0.00%	1.10%	0.00%	7.69%	9.89%
10/21/2005	Count of Event			1		2	3
	Percentage	0.00%	0.00%	1.82%	0.00%	3.64%	5.45%
10/24/2005	Count of Event			3	2	6	
	Percentage	0.00%	0.00%	7.50%	5.00%	15.00%	0.00%
10/25/2005	Count of Event		1			6	
	Percentage	0.00%	1.25%	0.00%	0.00%	7.50%	0.00%
10/26/2005	Count of Event			2	1	5	18

	Percentage	0.00%	0.00%	1.60%	0.80%	4.00%	14.40%
10/27/2005	Count of Event			3	3	7	44
	Percentage	0.00%	0.00%	1.22%	1.22%	2.86%	17.96%
10/28/2005	Count of Event					2	3
	Percentage	0.00%	0.00%	0.00%	0.00%	2.47%	3.70%
10/31/2005	Count of Event		2	2	2	10	2
	Percentage	0.00%	1.68%	1.68%	1.68%	8.40%	1.68%
11/3/2005	Count of Event			2	2	10	2
	Percentage	0.00%	0.00%	1.33%	1.33%	6.67%	1.33%
11/4/2005	Count of Event				2	6	7
	Percentage	0.00%	0.00%	0.00%	3.51%	10.53%	12.28%
11/7/2005	Count of Event				3	3	
	Percentage	0.00%	0.00%	0.00%	16.67%	16.67%	0.00%
11/8/2005	Count of Event		1		1		
	Percentage	0.00%	7.69%	0.00%	7.69%	0.00%	0.00%
11/9/2005	Count of Event					2	6
	Percentage	0.00%	0.00%	0.00%	0.00%	4.65%	13.95%
11/10/2005	Count of Event				2	4	2
	Percentage	0.00%	0.00%	0.00%	7.69%	15.38%	7.69%
11/14/2005	Count of Event				1		2
	Percentage	0.00%	0.00%	0.00%	3.33%	0.00%	6.67%
11/15/2005	Count of Event				1		1
	Percentage	0.00%	0.00%	0.00%	7.14%	0.00%	7.14%
11/16/2005	Count of Event			1	1	2	
	Percentage	0.00%	0.00%	16.67%	16.67%	33.33%	0.00%
Total Count of Event		2	33	34	28	128	162
Total Percentage		0.08%	1.38%	1.42%	1.17%	5.34%	6.76%

Date	Data	Disconnect	Explorer_Activate	Find	Folder_Switch	Forward	Inspector_Activate
9/26/2005	Count of Event						2
	Percentage	0.00%	0.00%	0.00%	0.00%	0.00%	16.67%
9/27/2005	Count of Event						
	Percentage	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
9/28/2005	Count of Event		8		5		1
	Percentage	0.00%	7.55%	0.00%	4.72%	0.00%	0.94%
9/29/2005	Count of Event	2	6		3	1	6
	Percentage	2.15%	6.45%	0.00%	3.23%	1.08%	6.45%
9/30/2005	Count of Event	1	11		16	5	14
	Percentage	0.34%	3.69%	0.00%	5.37%	1.68%	4.70%
10/5/2005	Count of Event				10	1	
	Percentage	0.00%	0.00%	0.00%	5.43%	0.54%	0.00%
10/6/2005	Count of Event	3	6		17		2
	Percentage	2.26%	4.51%	0.00%	12.78%	0.00%	1.50%
10/7/2005	Count of Event		2		4		2
	Percentage	0.00%	2.53%	0.00%	5.06%	0.00%	2.53%
10/17/2005	Count of Event				7	1	1
	Percentage	0.00%	0.00%	0.00%	5.04%	0.72%	0.72%
10/18/2005	Count of Event		1		4		1
	Percentage	0.00%	1.12%	0.00%	4.49%	0.00%	1.12%

10/19/2005	Count of Event				2		
	Percentage	0.00%	0.00%	0.00%	6.67%	0.00%	0.00%
10/20/2005	Count of Event		4		9	1	3
	Percentage	0.00%	4.40%	0.00%	9.89%	1.10%	3.30%
10/21/2005	Count of Event		2		4	1	1
	Percentage	0.00%	3.64%	0.00%	7.27%	1.82%	1.82%
10/24/2005	Count of Event	2	6				
	Percentage	5.00%	15.00%	0.00%	0.00%	0.00%	0.00%
10/25/2005	Count of Event		4		10	1	3
	Percentage	0.00%	5.00%	0.00%	12.50%	1.25%	3.75%
10/26/2005	Count of Event	1	4		2		1
	Percentage	0.80%	3.20%	0.00%	1.60%	0.00%	0.80%
10/27/2005	Count of Event	3	7	1	6		
	Percentage	1.22%	2.86%	0.41%	2.45%	0.00%	0.00%
10/28/2005	Count of Event	1	2		40		
	Percentage	1.23%	2.47%	0.00%	49.38%	0.00%	0.00%
10/31/2005	Count of Event	2	9		10	2	1
	Percentage	1.68%	7.56%	0.00%	8.40%	1.68%	0.84%
11/3/2005	Count of Event	1	7		9		3
	Percentage	0.67%	4.67%	0.00%	6.00%	0.00%	2.00%
11/4/2005	Count of Event	2	6		2		
	Percentage	3.51%	10.53%	0.00%	3.51%	0.00%	0.00%
11/7/2005	Count of Event	4	3				
	Percentage	22.22%	16.67%	0.00%	0.00%	0.00%	0.00%
11/8/2005	Count of Event						
	Percentage	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
11/9/2005	Count of Event	1	1			1	1
	Percentage	2.33%	2.33%	0.00%	0.00%	2.33%	2.33%
11/10/2005	Count of Event	2	3				1
	Percentage	7.69%	11.54%	0.00%	0.00%	0.00%	3.85%
11/14/2005	Count of Event	1					
	Percentage	3.33%	0.00%	0.00%	0.00%	0.00%	0.00%
11/15/2005	Count of Event	1				1	
	Percentage	7.14%	0.00%	0.00%	0.00%	7.14%	0.00%
11/16/2005	Count of Event		1				
	Percentage	0.00%	16.67%	0.00%	0.00%	0.00%	0.00%
Total Count of Event		27	93	1	160	15	43
Total Percentage		1.13%	3.88%	0.04%	6.68%	0.63%	1.79%

Date	Data	Inspector_Close	Item_Send	Open	Preview	Purge_Deleted_Messages
9/26/2005	Count of Event		2		3	
	Percentage	0.00%	16.67%	0.00%	25.00%	0.00%
9/27/2005	Count of Event		2		33	
	Percentage	0.00%	4.88%	0.00%	80.49%	0.00%
9/28/2005	Count of Event	4	5	3	61	
	Percentage	3.77%	4.72%	2.83%	57.55%	0.00%
9/29/2005	Count of Event	4	4	3	44	1
	Percentage	4.30%	4.30%	3.23%	47.31%	1.08%
9/30/2005	Count of Event	8	22	12	151	3

	Percentage	2.68%	7.38%	4.03%	50.67%	1.01%
10/5/2005	Count of Event	2	11	5	120	
	Percentage	1.09%	5.98%	2.72%	65.22%	0.00%
10/6/2005	Count of Event	1	12	6	65	
	Percentage	0.75%	9.02%	4.51%	48.87%	0.00%
10/7/2005	Count of Event	2	3	3	47	
	Percentage	2.53%	3.80%	3.80%	59.49%	0.00%
10/17/2005	Count of Event	3	6	3	88	1
	Percentage	2.16%	4.32%	2.16%	63.31%	0.72%
10/18/2005	Count of Event	1	7	1	59	
	Percentage	1.12%	7.87%	1.12%	66.29%	0.00%
10/19/2005	Count of Event		4	3	17	
	Percentage	0.00%	13.33%	10.00%	56.67%	0.00%
10/20/2005	Count of Event		3	1	53	
	Percentage	0.00%	3.30%	1.10%	58.24%	0.00%
10/21/2005	Count of Event		5		33	
	Percentage	0.00%	9.09%	0.00%	60.00%	0.00%
10/24/2005	Count of Event	1	5		12	
	Percentage	2.50%	12.50%	0.00%	30.00%	0.00%
10/25/2005	Count of Event	1	9	2	36	
	Percentage	1.25%	11.25%	2.50%	45.00%	0.00%
10/26/2005	Count of Event		7		78	
	Percentage	0.00%	5.60%	0.00%	62.40%	0.00%
10/27/2005	Count of Event	1	7	1	156	1
	Percentage	0.41%	2.86%	0.41%	63.67%	0.41%
10/28/2005	Count of Event				33	
	Percentage	0.00%	0.00%	0.00%	40.74%	0.00%
10/31/2005	Count of Event		7		67	
	Percentage	0.00%	5.88%	0.00%	56.30%	0.00%
11/3/2005	Count of Event	3	7	2	95	1
	Percentage	2.00%	4.67%	1.33%	63.33%	0.67%
11/4/2005	Count of Event		1		30	
	Percentage	0.00%	1.75%	0.00%	52.63%	0.00%
11/7/2005	Count of Event				5	
	Percentage	0.00%	0.00%	0.00%	27.78%	0.00%
11/8/2005	Count of Event		1		7	2
	Percentage	0.00%	7.69%	0.00%	53.85%	15.38%
11/9/2005	Count of Event		2		28	
	Percentage	0.00%	4.65%	0.00%	65.12%	0.00%
11/10/2005	Count of Event	1	1	1	8	
	Percentage	3.85%	3.85%	3.85%	30.77%	0.00%
11/14/2005	Count of Event		2		21	1
	Percentage	0.00%	6.67%	0.00%	70.00%	3.33%
11/15/2005	Count of Event		1		8	1
	Percentage	0.00%	7.14%	0.00%	57.14%	7.14%
11/16/2005	Count of Event					1
	Percentage	0.00%	0.00%	0.00%	0.00%	16.67%
Total Count of Event		32	136	46	1358	12
Total Percentage		1.34%	5.67%	1.92%	56.65%	0.50%

Date	Data	Reply	Reply_to_All	Undelete	Undo	Grand Total
9/26/2005	Count of Event	1	1			12
	Percentage	8.33%	8.33%	0.00%	0.00%	100.00%
9/27/2005	Count of Event	2				41
	Percentage	4.88%	0.00%	0.00%	0.00%	100.00%
9/28/2005	Count of Event	2	3			106
	Percentage	1.89%	2.83%	0.00%	0.00%	100.00%
9/29/2005	Count of Event	2	2		1	93
	Percentage	2.15%	2.15%	0.00%	1.08%	100.00%
9/30/2005	Count of Event	7	3			298
	Percentage	2.35%	1.01%	0.00%	0.00%	100.00%
10/5/2005	Count of Event	7	1	1		184
	Percentage	3.80%	0.54%	0.54%	0.00%	100.00%
10/6/2005	Count of Event	2	2			133
	Percentage	1.50%	1.50%	0.00%	0.00%	100.00%
10/7/2005	Count of Event	2				79
	Percentage	2.53%	0.00%	0.00%	0.00%	100.00%
10/17/2005	Count of Event		1			139
	Percentage	0.00%	0.72%	0.00%	0.00%	100.00%
10/18/2005	Count of Event	4	4			89
	Percentage	4.49%	4.49%	0.00%	0.00%	100.00%
10/19/2005	Count of Event					30
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/20/2005	Count of Event					91
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/21/2005	Count of Event	1	2			55
	Percentage	1.82%	3.64%	0.00%	0.00%	100.00%
10/24/2005	Count of Event	1	2			40
	Percentage	2.50%	5.00%	0.00%	0.00%	100.00%
10/25/2005	Count of Event	5	2			80
	Percentage	6.25%	2.50%	0.00%	0.00%	100.00%
10/26/2005	Count of Event	4	1		1	125
	Percentage	3.20%	0.80%	0.00%	0.80%	100.00%
10/27/2005	Count of Event		4		1	245
	Percentage	0.00%	1.63%	0.00%	0.41%	100.00%
10/28/2005	Count of Event					81
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/31/2005	Count of Event	2	1			119
	Percentage	1.68%	0.84%	0.00%	0.00%	100.00%
11/3/2005	Count of Event	2	4			150
	Percentage	1.33%	2.67%	0.00%	0.00%	100.00%
11/4/2005	Count of Event		1			57
	Percentage	0.00%	1.75%	0.00%	0.00%	100.00%
11/7/2005	Count of Event					18
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
11/8/2005	Count of Event	1				13
	Percentage	7.69%	0.00%	0.00%	0.00%	100.00%
11/9/2005	Count of Event	1				43
	Percentage	2.33%	0.00%	0.00%	0.00%	100.00%
11/10/2005	Count of Event	1				26

	Percentage	3.85%	0.00%	0.00%	0.00%	100.00%
11/14/2005	Count of Event	1	1			30
	Percentage	3.33%	3.33%	0.00%	0.00%	100.00%
11/15/2005	Count of Event					14
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
11/16/2005	Count of Event					6
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
Total Count of Event		48	35	1	3	2397
Total Percentage		2.00%	1.46%	0.04%	0.13%	100.00%
			Total Days Data Collected			28

7.1.3 *User3 Pivot Table*

Date	Data	Advanced_Find	Attachment_Read	Compose_New_Mail	Connect	Deactivate
10/20/2005	Count of Event		1		2	15
	Percentage	0.00%	2.04%	0.00%	4.08%	30.61%
10/21/2005	Count of Event	1	2	1	1	130
	Percentage	0.17%	0.35%	0.17%	0.17%	22.61%
10/24/2005	Count of Event	1	15	2	1	64
	Percentage	0.30%	4.46%	0.60%	0.30%	19.05%
10/25/2005	Count of Event		2		1	24
	Percentage	0.00%	1.71%	0.00%	0.85%	20.51%
10/26/2005	Count of Event		2	4	1	55
	Percentage	0.00%	0.69%	1.38%	0.35%	19.03%
10/27/2005	Count of Event		6		3	14
	Percentage	0.00%	3.82%	0.00%	1.91%	8.92%
10/28/2005	Count of Event		3	6	1	63
	Percentage	0.00%	0.88%	1.76%	0.29%	18.53%
10/29/2005	Count of Event					13
	Percentage	0.00%	0.00%	0.00%	0.00%	22.81%
10/30/2005	Count of Event		4	3	2	36
	Percentage	0.00%	2.38%	1.79%	1.19%	21.43%
10/31/2005	Count of Event				1	3
	Percentage	0.00%	0.00%	0.00%	5.56%	16.67%
11/3/2005	Count of Event				1	11
	Percentage	0.00%	0.00%	0.00%	3.23%	35.48%
11/4/2005	Count of Event			3	1	36
	Percentage	0.00%	0.00%	1.20%	0.40%	14.40%
11/7/2005	Count of Event		1	1	1	50
	Percentage	0.00%	0.36%	0.36%	0.36%	18.12%
Total Count of Event		2	36	20	16	514
Total Percentage		0.08%	1.35%	0.75%	0.60%	19.30%

Date	Data	Delete	Disconnect	Explorer_Activate	Find	Folder_Add	Folder_Remove
10/20/2005	Count of Event		2	12			

	Percentage	0.00%	4.08%	24.49%	0.00%	0.00%	0.00%
10/21/2005	Count of Event	7	1	102			
	Percentage	1.22%	0.17%	17.74%	0.00%	0.00%	0.00%
10/24/2005	Count of Event	5	1	47			
	Percentage	1.49%	0.30%	13.99%	0.00%	0.00%	0.00%
10/25/2005	Count of Event	7	1	14			
	Percentage	5.98%	0.85%	11.97%	0.00%	0.00%	0.00%
10/26/2005	Count of Event	2		52			
	Percentage	0.69%	0.00%	17.99%	0.00%	0.00%	0.00%
10/27/2005	Count of Event	1	4	11			
	Percentage	0.64%	2.55%	7.01%	0.00%	0.00%	0.00%
10/28/2005	Count of Event	12		59			
	Percentage	3.53%	0.00%	17.35%	0.00%	0.00%	0.00%
10/29/2005	Count of Event	1		7			
	Percentage	1.75%	0.00%	12.28%	0.00%	0.00%	0.00%
10/30/2005	Count of Event	2	3	16			
	Percentage	1.19%	1.79%	9.52%	0.00%	0.00%	0.00%
10/31/2005	Count of Event	1	1	3			
	Percentage	5.56%	5.56%	16.67%	0.00%	0.00%	0.00%
11/3/2005	Count of Event	1	1	11			
	Percentage	3.23%	3.23%	35.48%	0.00%	0.00%	0.00%
11/4/2005	Count of Event	6	1	22	4	5	2
	Percentage	2.40%	0.40%	8.80%	1.60%	2.00%	0.80%
11/7/2005	Count of Event	7	1	42		1	
	Percentage	2.54%	0.36%	15.22%	0.00%	0.36%	0.00%
Total Count of Event		52	16	398	4	6	2
Total Percentage		1.95%	0.60%	14.95%	0.15%	0.23%	0.08%

Date	Data	Folder_Switch	Forward	Inspector_Activate	Inspector_Close	Item_Send
10/20/2005	Count of Event		1	3		1
	Percentage	0.00%	2.04%	6.12%	0.00%	2.04%
10/21/2005	Count of Event	39	2	32	4	18
	Percentage	6.78%	0.35%	5.57%	0.70%	3.13%
10/24/2005	Count of Event	10	5	25	14	12
	Percentage	2.98%	1.49%	7.44%	4.17%	3.57%
10/25/2005	Count of Event	2		10	2	4
	Percentage	1.71%	0.00%	8.55%	1.71%	3.42%
10/26/2005	Count of Event	11	2	5	2	9
	Percentage	3.81%	0.69%	1.73%	0.69%	3.11%
10/27/2005	Count of Event	13	4	3	5	5
	Percentage	8.28%	2.55%	1.91%	3.18%	3.18%
10/28/2005	Count of Event	27	3	5	3	11
	Percentage	7.94%	0.88%	1.47%	0.88%	3.24%
10/29/2005	Count of Event	2		6		2
	Percentage	3.51%	0.00%	10.53%	0.00%	3.51%
10/30/2005	Count of Event	14		25	2	6
	Percentage	8.33%	0.00%	14.88%	1.19%	3.57%
10/31/2005	Count of Event	2				

	Percentage	11.11%	0.00%	0.00%	0.00%	0.00%
11/3/2005	Count of Event					
	Percentage	0.00%	0.00%	0.00%	0.00%	0.00%
11/4/2005	Count of Event	13	4	15	3	12
	Percentage	5.20%	1.60%	6.00%	1.20%	4.80%
11/7/2005	Count of Event	35	3	11	4	10
	Percentage	12.68%	1.09%	3.99%	1.45%	3.62%
Total Count of Event		168	24	140	39	90
Total Percentage		6.31%	0.90%	5.26%	1.46%	3.38%

Date	Data	Move_to_Folder	Open	Preview	Print	Purge_Deleted_Messages	Reply
10/20/2005	Count of Event			12			
	Percentage	0.00%	0.00%	24.49%	0.00%	0.00%	0.00%
10/21/2005	Count of Event	1	2	213		1	10
	Percentage	0.17%	0.35%	37.04%	0.00%	0.17%	1.74%
10/24/2005	Count of Event		8	115			7
	Percentage	0.00%	2.38%	34.23%	0.00%	0.00%	2.08%
10/25/2005	Count of Event		1	42	1		4
	Percentage	0.00%	0.85%	35.90%	0.85%	0.00%	3.42%
10/26/2005	Count of Event		1	138			4
	Percentage	0.00%	0.35%	47.75%	0.00%	0.00%	1.38%
10/27/2005	Count of Event		3	82			3
	Percentage	0.00%	1.91%	52.23%	0.00%	0.00%	1.91%
10/28/2005	Count of Event		2	139	1	2	3
	Percentage	0.00%	0.59%	40.88%	0.29%	0.59%	0.88%
10/29/2005	Count of Event		1	22			1
	Percentage	0.00%	1.75%	38.60%	0.00%	0.00%	1.75%
10/30/2005	Count of Event		2	49		1	3
	Percentage	0.00%	1.19%	29.17%	0.00%	0.60%	1.79%
10/31/2005	Count of Event			7			
	Percentage	0.00%	0.00%	38.89%	0.00%	0.00%	0.00%
11/3/2005	Count of Event			6			
	Percentage	0.00%	0.00%	19.35%	0.00%	0.00%	0.00%
11/4/2005	Count of Event		1	92	19	3	6
	Percentage	0.00%	0.40%	36.80%	7.60%	1.20%	2.40%
11/7/2005	Count of Event		1	96		2	8
	Percentage	0.00%	0.36%	34.78%	0.00%	0.72%	2.90%
Total Count of Event		1	22	1013	21	9	49
Total Percentage		0.04%	0.83%	38.04%	0.79%	0.34%	1.84%

Date	Data	Reply_to_All	Save	Undelete	Undo	Grand Total
10/20/2005	Count of					49

	Event					
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/21/2005	Count of Event	7		1		575
	Percentage	1.22%	0.00%	0.17%	0.00%	100.00%
10/24/2005	Count of Event	4				336
	Percentage	1.19%	0.00%	0.00%	0.00%	100.00%
10/25/2005	Count of Event	1	1			117
	Percentage	0.85%	0.85%	0.00%	0.00%	100.00%
10/26/2005	Count of Event				1	289
	Percentage	0.00%	0.00%	0.00%	0.35%	100.00%
10/27/2005	Count of Event					157
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/28/2005	Count of Event					340
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/29/2005	Count of Event	1			1	57
	Percentage	1.75%	0.00%	0.00%	1.75%	100.00%
10/30/2005	Count of Event					168
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/31/2005	Count of Event					18
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
11/3/2005	Count of Event					31
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
11/4/2005	Count of Event	1			1	250
	Percentage	0.40%	0.00%	0.00%	0.40%	100.00%
11/7/2005	Count of Event				2	276
	Percentage	0.00%	0.00%	0.00%	0.72%	100.00%
Total Count of Event		14	1	1	5	2663
Total Percentage		0.53%	0.04%	0.04%	0.19%	100.00%

Total Days Data Collected 13

7.1.4 *User4 Pivot Table*

Date	Data	Attachment_Read	Compose_New_Mail	Connect	Deactivate	Delete
10/13/2005	Count of Event	7	3	2	38	59
	Percentage	1.14%	0.49%	0.33%	6.18%	9.59%
10/14/2005	Count of Event		2		1	1
	Percentage	0.00%	8.33%	0.00%	4.17%	4.17%
10/17/2005	Count of Event	3	5	2	57	123
	Percentage	0.35%	0.59%	0.24%	6.71%	14.47%
10/18/2005	Count of Event	4	3	1	31	10
	Percentage	1.69%	1.27%	0.42%	13.08%	4.22%

10/19/2005	Count of Event	4	5	1	17	8
	Percentage	1.96%	2.45%	0.49%	8.33%	3.92%
10/20/2005	Count of Event		1	2	56	21
	Percentage	0.00%	0.31%	0.62%	17.28%	6.48%
10/21/2005	Count of Event		1		2	
	Percentage	0.00%	9.09%	0.00%	18.18%	0.00%
10/24/2005	Count of Event	1	1	3	21	3
	Percentage	0.71%	0.71%	2.14%	15.00%	2.14%
10/25/2005	Count of Event		1	1	7	8
	Percentage	0.00%	1.15%	1.15%	8.05%	9.20%
10/26/2005	Count of Event		3	2	27	30
	Percentage	0.00%	1.02%	0.68%	9.18%	10.20%
10/27/2005	Count of Event	1	7	1	57	21
	Percentage	0.27%	1.88%	0.27%	15.28%	5.63%
10/31/2005	Count of Event		5	3	41	12
	Percentage	0.00%	1.36%	0.82%	11.17%	3.27%
11/1/2005	Count of Event	2		1	10	6
	Percentage	1.14%	0.00%	0.57%	5.71%	3.43%
11/2/2005	Count of Event	1		1	7	4
	Percentage	1.19%	0.00%	1.19%	8.33%	4.76%
11/3/2005	Count of Event	2	2	4	18	1
	Percentage	1.10%	1.10%	2.20%	9.89%	0.55%
11/10/2005	Count of Event		1	2	8	9
	Percentage	0.00%	0.67%	1.34%	5.37%	6.04%
11/14/2005	Count of Event	3	1	1	37	22
	Percentage	0.55%	0.18%	0.18%	6.79%	4.04%
11/15/2005	Count of Event		7	2	22	12
	Percentage	0.00%	4.22%	1.20%	13.25%	7.23%
11/16/2005	Count of Event	3	4	1	34	12
	Percentage	1.33%	1.77%	0.44%	15.04%	5.31%
11/17/2005	Count of Event		5	1	24	13
	Percentage	0.00%	1.92%	0.38%	9.23%	5.00%
11/18/2005	Count of Event	1	2	2	10	5
	Percentage	0.96%	1.92%	1.92%	9.62%	4.81%
11/20/2005	Count of Event			1		
	Percentage	0.00%	0.00%	12.50%	0.00%	0.00%
11/21/2005	Count of Event	2	3	1	34	11
	Percentage	0.75%	1.12%	0.37%	12.73%	4.12%
11/22/2005	Count of Event		1	3	41	8
	Percentage	0.00%	0.41%	1.24%	16.94%	3.31%
11/23/2005	Count of Event	3	2	1	3	6
	Percentage	3.06%	2.04%	1.02%	3.06%	6.12%
11/25/2005	Count of Event			1	1	1
	Percentage	0.00%	0.00%	4.00%	4.00%	4.00%
11/28/2005	Count of Event		1	2	7	12
	Percentage	0.00%	0.61%	1.23%	4.29%	7.36%
11/29/2005	Count of Event	1	2	2	39	18
	Percentage	0.30%	0.60%	0.60%	11.64%	5.37%
11/30/2005	Count of Event		3	1	25	82

	Percentage	0.00%	0.84%	0.28%	7.00%	22.97%
12/1/2005	Count of Event		2	1	37	14
	Percentage	0.00%	0.57%	0.29%	10.57%	4.00%
12/2/2005	Count of Event			2	9	7
	Percentage	0.00%	0.00%	1.45%	6.52%	5.07%
12/5/2005	Count of Event		10	5	62	45
	Percentage	0.00%	2.09%	1.04%	12.94%	9.39%
12/6/2005	Count of Event	32	3	2	24	9
	Percentage	16.49%	1.55%	1.03%	12.37%	4.64%
12/7/2005	Count of Event			4	19	6
	Percentage	0.00%	0.00%	3.92%	18.63%	5.88%
12/8/2005	Count of Event		3	1	38	9
	Percentage	0.00%	1.20%	0.40%	15.20%	3.60%
12/9/2005	Count of Event				1	
	Percentage	0.00%	0.00%	0.00%	25.00%	0.00%
12/14/2005	Count of Event			1	3	2
	Percentage	0.00%	0.00%	3.85%	11.54%	7.69%
Total Count of Event		70	89	61	868	610
Total Percentage		0.83%	1.05%	0.72%	10.27%	7.21%

Date	Data	Disconnect	Explorer_Activate	Folder_Add	Folder_Switch	Forward	Inspector_Activate
10/13/2005	Count of Event	1	28		10	1	14
	Percentage	0.16%	4.55%	0.00%	1.63%	0.16%	2.28%
10/14/2005	Count of Event	1	1		6		1
	Percentage	4.17%	4.17%	0.00%	25.00%	0.00%	4.17%
10/17/2005	Count of Event	2	35		10	2	25
	Percentage	0.24%	4.12%	0.00%	1.18%	0.24%	2.94%
10/18/2005	Count of Event	1	20		9	1	13
	Percentage	0.42%	8.44%	0.00%	3.80%	0.42%	5.49%
10/19/2005	Count of Event	1	10		8	2	9
	Percentage	0.49%	4.90%	0.00%	3.92%	0.98%	4.41%
10/20/2005	Count of Event	1	21		4	1	35
	Percentage	0.31%	6.48%	0.00%	1.23%	0.31%	10.80%
10/21/2005	Count of Event	1	3				
	Percentage	9.09%	27.27%	0.00%	0.00%	0.00%	0.00%
10/24/2005	Count of Event	3	18			1	3
	Percentage	2.14%	12.86%	0.00%	0.00%	0.71%	2.14%
10/25/2005	Count of Event	1	7				
	Percentage	1.15%	8.05%	0.00%	0.00%	0.00%	0.00%
10/26/2005	Count of Event	2	18				10
	Percentage	0.68%	6.12%	0.00%	0.00%	0.00%	3.40%
10/27/2005	Count of Event	1	36		8	1	24
	Percentage	0.27%	9.65%	0.00%	2.14%	0.27%	6.43%
10/31/2005	Count of Event	3	21		29	1	24
	Percentage	0.82%	5.72%	0.00%	7.90%	0.27%	6.54%
11/1/2005	Count of Event	1	8		8	2	2
	Percentage	0.57%	4.57%	0.00%	4.57%	1.14%	1.14%
11/2/2005	Count of Event	1	7				1

	Percentage	1.19%	8.33%	0.00%	0.00%	0.00%	1.19%
11/3/2005	Count of Event	4	14		22		5
	Percentage	2.20%	7.69%	0.00%	12.09%	0.00%	2.75%
11/10/2005	Count of Event	2	4		3	1	5
	Percentage	1.34%	2.68%	0.00%	2.01%	0.67%	3.36%
11/14/2005	Count of Event	1	21	1	6		19
	Percentage	0.18%	3.85%	0.18%	1.10%	0.00%	3.49%
11/15/2005	Count of Event	2	21		4		2
	Percentage	1.20%	12.65%	0.00%	2.41%	0.00%	1.20%
11/16/2005	Count of Event	1	22		9		14
	Percentage	0.44%	9.73%	0.00%	3.98%	0.00%	6.19%
11/17/2005	Count of Event	1	12		9		13
	Percentage	0.38%	4.62%	0.00%	3.46%	0.00%	5.00%
11/18/2005	Count of Event	2	7		5	2	5
	Percentage	1.92%	6.73%	0.00%	4.81%	1.92%	4.81%
11/20/2005	Count of Event	1			2		
	Percentage	12.50%	0.00%	0.00%	25.00%	0.00%	0.00%
11/21/2005	Count of Event	1	24		5		12
	Percentage	0.37%	8.99%	0.00%	1.87%	0.00%	4.49%
11/22/2005	Count of Event	3	29				14
	Percentage	1.24%	11.98%	0.00%	0.00%	0.00%	5.79%
11/23/2005	Count of Event	1	3				1
	Percentage	1.02%	3.06%	0.00%	0.00%	0.00%	1.02%
11/25/2005	Count of Event	1	1				
	Percentage	4.00%	4.00%	0.00%	0.00%	0.00%	0.00%
11/28/2005	Count of Event	2	3		2		6
	Percentage	1.23%	1.84%	0.00%	1.23%	0.00%	3.68%
11/29/2005	Count of Event	1	26		6		14
	Percentage	0.30%	7.76%	0.00%	1.79%	0.00%	4.18%
11/30/2005	Count of Event	2	23		4		3
	Percentage	0.56%	6.44%	0.00%	1.12%	0.00%	0.84%
12/1/2005	Count of Event		22		23		15
	Percentage	0.00%	6.29%	0.00%	6.57%	0.00%	4.29%
12/2/2005	Count of Event	3	8				2
	Percentage	2.17%	5.80%	0.00%	0.00%	0.00%	1.45%
12/5/2005	Count of Event	5	48		8	1	15
	Percentage	1.04%	10.02%	0.00%	1.67%	0.21%	3.13%
12/6/2005	Count of Event	2	23		2		2
	Percentage	1.03%	11.86%	0.00%	1.03%	0.00%	1.03%
12/7/2005	Count of Event	4	16				2
	Percentage	3.92%	15.69%	0.00%	0.00%	0.00%	1.96%
12/8/2005	Count of Event		23		8		14
	Percentage	0.00%	9.20%	0.00%	3.20%	0.00%	5.60%
12/9/2005	Count of Event	1	2				
	Percentage	25.00%	50.00%	0.00%	0.00%	0.00%	0.00%
12/14/2005	Count of Event						3
	Percentage	0.00%	0.00%	0.00%	0.00%	0.00%	11.54%
Total Count of Event		60	585	1	210	16	327
Total Percentage		0.71%	6.92%	0.01%	2.48%	0.19%	3.87%

Date	Data	Reply	Reply_to_All	Undelete	Grand Total
10/13/2005	Count of Event	9	1		615
	Percentage	1.46%	0.16%	0.00%	100.00%
10/14/2005	Count of Event				24
	Percentage	0.00%	0.00%	0.00%	100.00%
10/17/2005	Count of Event	6			850
	Percentage	0.71%	0.00%	0.00%	100.00%
10/18/2005	Count of Event	10			237
	Percentage	4.22%	0.00%	0.00%	100.00%
10/19/2005	Count of Event	4			204
	Percentage	1.96%	0.00%	0.00%	100.00%
10/20/2005	Count of Event	13	2		324
	Percentage	4.01%	0.62%	0.00%	100.00%
10/21/2005	Count of Event				11
	Percentage	0.00%	0.00%	0.00%	100.00%
10/24/2005	Count of Event	4			140
	Percentage	2.86%	0.00%	0.00%	100.00%
10/25/2005	Count of Event	3			87
	Percentage	3.45%	0.00%	0.00%	100.00%
10/26/2005	Count of Event	7			294
	Percentage	2.38%	0.00%	0.00%	100.00%
10/27/2005	Count of Event	11			373
	Percentage	2.95%	0.00%	0.00%	100.00%
10/31/2005	Count of Event	8			367
	Percentage	2.18%	0.00%	0.00%	100.00%
11/1/2005	Count of Event	9	1		175
	Percentage	5.14%	0.57%	0.00%	100.00%
11/2/2005	Count of Event	1			84
	Percentage	1.19%	0.00%	0.00%	100.00%
11/3/2005	Count of Event	5	1		182
	Percentage	2.75%	0.55%	0.00%	100.00%
11/10/2005	Count of Event	3			149
	Percentage	2.01%	0.00%	0.00%	100.00%
11/14/2005	Count of Event	19			545
	Percentage	3.49%	0.00%	0.00%	100.00%
11/15/2005	Count of Event	3			166
	Percentage	1.81%	0.00%	0.00%	100.00%
11/16/2005	Count of Event	9			226
	Percentage	3.98%	0.00%	0.00%	100.00%
11/17/2005	Count of Event	4			260
	Percentage	1.54%	0.00%	0.00%	100.00%
11/18/2005	Count of Event	2	1		104
	Percentage	1.92%	0.96%	0.00%	100.00%
11/20/2005	Count of Event				8
	Percentage	0.00%	0.00%	0.00%	100.00%
11/21/2005	Count of Event	7			267
	Percentage	2.62%	0.00%	0.00%	100.00%

11/22/2005	Count of Event	3			242
	Percentage	1.24%	0.00%	0.00%	100.00%
11/23/2005	Count of Event	2			98
	Percentage	2.04%	0.00%	0.00%	100.00%
11/25/2005	Count of Event	1			25
	Percentage	4.00%	0.00%	0.00%	100.00%
11/28/2005	Count of Event	5	1		163
	Percentage	3.07%	0.61%	0.00%	100.00%
11/29/2005	Count of Event	5			335
	Percentage	1.49%	0.00%	0.00%	100.00%
11/30/2005	Count of Event	2			357
	Percentage	0.56%	0.00%	0.00%	100.00%
12/1/2005	Count of Event	7	1	1	350
	Percentage	2.00%	0.29%	0.29%	100.00%
12/2/2005	Count of Event	4			138
	Percentage	2.90%	0.00%	0.00%	100.00%
12/5/2005	Count of Event	6			479
	Percentage	1.25%	0.00%	0.00%	100.00%
12/6/2005	Count of Event	5			194
	Percentage	2.58%	0.00%	0.00%	100.00%
12/7/2005	Count of Event	2			102
	Percentage	1.96%	0.00%	0.00%	100.00%
12/8/2005	Count of Event	6			250
	Percentage	2.40%	0.00%	0.00%	100.00%
12/9/2005	Count of Event				4
	Percentage	0.00%	0.00%	0.00%	100.00%
12/14/2005	Count of Event	2			26
	Percentage	7.69%	0.00%	0.00%	100.00%
Total Count of Event		187	8	1	8455
Total Percentage		2.21%	0.09%	0.01%	100.00%
		Total Days Data Collected			37

7.1.5 *User5 Pivot Table*

Date	Data	Attachment_Read	Compose_New_Mail	Connect	Deactivate	Delete	Disconnect
11/9/2005	Count of Event	2		2	4		1
	Percentage	8.33%	0.00%	8.33%	16.67%	0.00%	4.17%
11/14/2005	Count of Event		3		23	1	1
	Percentage	0.00%	3.16%	0.00%	24.21%	1.05%	1.05%
11/21/2005	Count of Event	1	2	2	28		1
	Percentage	1.41%	2.82%	2.82%	39.44%	0.00%	1.41%
11/28/2005	Count of Event	3			31		
	Percentage	2.10%	0.00%	0.00%	21.68%	0.00%	0.00%
12/2/2005	Count of Event		7	1	59	1	1
	Percentage	0.00%	3.57%	0.51%	30.10%	0.51%	0.51%
12/5/2005	Count of Event			1	14		1

	Percentage	0.00%	0.00%	2.86%	40.00%	0.00%	2.86%
12/12/2005	Count of Event		1		16		
	Percentage	0.00%	2.63%	0.00%	42.11%	0.00%	0.00%
Total Count of Event		6	13	6	175	2	5
Total Percentage		1.00%	2.16%	1.00%	29.07%	0.33%	0.83%

Date	Data	Explorer_Activate	Folder_Switch	Forward	Inspector_Activate	Inspector_Close
11/9/2005	Count of Event	3	4			
	Percentage	12.50%	16.67%	0.00%	0.00%	0.00%
11/14/2005	Count of Event	23	3	1	1	
	Percentage	24.21%	3.16%	1.05%	1.05%	0.00%
11/21/2005	Count of Event	18			9	1
	Percentage	25.35%	0.00%	0.00%	12.68%	1.41%
11/28/2005	Count of Event	27	11	1	5	2
	Percentage	18.88%	7.69%	0.70%	3.50%	1.40%
12/2/2005	Count of Event	49	4	2	12	
	Percentage	25.00%	2.04%	1.02%	6.12%	0.00%
12/5/2005	Count of Event	13	1			
	Percentage	37.14%	2.86%	0.00%	0.00%	0.00%
12/12/2005	Count of Event	17				
	Percentage	44.74%	0.00%	0.00%	0.00%	0.00%
Total Count of Event		150	23	4	27	3
Total Percentage		24.92%	3.82%	0.66%	4.49%	0.50%

Date	Data	Item_Send	Open	Preview	Reply	Save	Undo	Grand Total
11/9/2005	Count of Event			8				24
	Percentage	0.00%	0.00%	33.33%	0.00%	0.00%	0.00%	100.00%
11/14/2005	Count of Event	4		35				95
	Percentage	4.21%	0.00%	36.84%	0.00%	0.00%	0.00%	100.00%
11/21/2005	Count of Event	3	1	3	1	1		71
	Percentage	4.23%	1.41%	4.23%	1.41%	1.41%	0.00%	100.00%
11/28/2005	Count of Event	13	5	38	6		1	143
	Percentage	9.09%	3.50%	26.57%	4.20%	0.00%	0.70%	100.00%
12/2/2005	Count of Event	11	2	45			2	196
	Percentage	5.61%	1.02%	22.96%	0.00%	0.00%	1.02%	100.00%
12/5/2005	Count of Event	1	1	3				35
	Percentage	2.86%	2.86%	8.57%	0.00%	0.00%	0.00%	100.00%
12/12/2005	Count of Event	1		2	1			38
	Percentage	2.63%	0.00%	5.26%	2.63%	0.00%	0.00%	100.00%
Total Count of Event		33	9	134	8	1	3	602
Total Percentage		5.48%	1.50%	22.26%	1.33%	0.17%	0.50%	100.00%

Total Days Data Collected 7

7.1.6 *User6 Pivot Table*

Date	Data	Advanced_Find	Attachment_Read	Compose_New_Mail	Connect	Deactivate
10/6/2005	Count of Event		1	1	2	3
	Percentage	0.00%	5.88%	5.88%	11.76%	17.65%
10/7/2005	Count of Event		5			13
	Percentage	0.00%	5.00%	0.00%	0.00%	13.00%
10/11/2005	Count of Event		3	1	1	18
	Percentage	0.00%	2.63%	0.88%	0.88%	15.79%
10/12/2005	Count of Event					5
	Percentage	0.00%	0.00%	0.00%	0.00%	19.23%
10/13/2005	Count of Event	1		2		25
	Percentage	0.58%	0.00%	1.16%	0.00%	14.45%
10/14/2005	Count of Event		1		2	4
	Percentage	0.00%	2.13%	0.00%	4.26%	8.51%
10/17/2005	Count of Event		4	1	2	8
	Percentage	0.00%	8.00%	2.00%	4.00%	16.00%
10/18/2005	Count of Event	1	4		1	28
	Percentage	0.64%	2.56%	0.00%	0.64%	17.95%
10/19/2005	Count of Event		3	1	1	7
	Percentage	0.00%	4.17%	1.39%	1.39%	9.72%
10/20/2005	Count of Event		5	2	1	10
	Percentage	0.00%	4.00%	1.60%	0.80%	8.00%
10/21/2005	Count of Event				3	4
	Percentage	0.00%	0.00%	0.00%	8.57%	11.43%
10/24/2005	Count of Event		1		2	7
	Percentage	0.00%	2.08%	0.00%	4.17%	14.58%
10/25/2005	Count of Event				2	10
	Percentage	0.00%	0.00%	0.00%	4.08%	20.41%
10/26/2005	Count of Event	3	1	1	2	14
	Percentage	2.59%	0.86%	0.86%	1.72%	12.07%
10/31/2005	Count of Event		9	1	1	42
	Percentage	0.00%	3.83%	0.43%	0.43%	17.87%
11/1/2005	Count of Event				1	4
	Percentage	0.00%	0.00%	0.00%	2.27%	9.09%
11/2/2005	Count of Event	3	4		2	25
	Percentage	1.42%	1.90%	0.00%	0.95%	11.85%
11/3/2005	Count of Event		6		3	12
	Percentage	0.00%	5.94%	0.00%	2.97%	11.88%
11/4/2005	Count of Event		5	1	4	29
	Percentage	0.00%	2.72%	0.54%	2.17%	15.76%
11/9/2005	Count of Event		1		1	2
	Percentage	0.00%	5.88%	0.00%	5.88%	11.76%
11/10/2005	Count of Event			1	2	18
	Percentage	0.00%	0.00%	0.94%	1.89%	16.98%
11/14/2005	Count of Event		4	4	4	48
	Percentage	0.00%	2.02%	2.02%	2.02%	24.24%
11/15/2005	Count of Event		2	2	1	23

	Percentage	0.00%	1.69%	1.69%	0.85%	19.49%
11/16/2005	Count of Event		1	2	1	13
	Percentage	0.00%	1.20%	2.41%	1.20%	15.66%
11/17/2005	Count of Event					10
	Percentage	0.00%	0.00%	0.00%	0.00%	29.41%
11/18/2005	Count of Event		11		1	13
	Percentage	0.00%	13.75%	0.00%	1.25%	16.25%
11/21/2005	Count of Event	1	4		1	18
	Percentage	0.95%	3.81%	0.00%	0.95%	17.14%
11/22/2005	Count of Event	1	8		2	11
	Percentage	1.09%	8.70%	0.00%	2.17%	11.96%
11/28/2005	Count of Event	1	7	3	1	34
	Percentage	0.54%	3.80%	1.63%	0.54%	18.48%
11/29/2005	Count of Event			1	1	10
	Percentage	0.00%	0.00%	1.54%	1.54%	15.38%
11/30/2005	Count of Event		3	3	1	16
	Percentage	0.00%	4.11%	4.11%	1.37%	21.92%
12/1/2005	Count of Event	1	1			13
	Percentage	1.19%	1.19%	0.00%	0.00%	15.48%
12/2/2005	Count of Event		3		1	5
	Percentage	0.00%	3.57%	0.00%	1.19%	5.95%
12/5/2005	Count of Event		4	1	1	5
	Percentage	0.00%	3.60%	0.90%	0.90%	4.50%
12/12/2005	Count of Event				3	6
	Percentage	0.00%	0.00%	0.00%	6.38%	12.77%
Total Count of Event		12	101	28	51	513
Total Percentage		0.35%	2.98%	0.83%	1.51%	15.16%

Date	Data	Delete	Disconnect	Explorer_Activate	Find	Folder_Add	Folder_Switch
10/6/2005	Count of Event		1	2			1
	Percentage	0.00%	5.88%	11.76%	0.00%	0.00%	5.88%
10/7/2005	Count of Event			14			
	Percentage	0.00%	0.00%	14.00%	0.00%	0.00%	0.00%
10/11/2005	Count of Event		1	17			7
	Percentage	0.00%	0.88%	14.91%	0.00%	0.00%	6.14%
10/12/2005	Count of Event			4			2
	Percentage	0.00%	0.00%	15.38%	0.00%	0.00%	7.69%
10/13/2005	Count of Event		1	16	1		15
	Percentage	0.00%	0.58%	9.25%	0.58%	0.00%	8.67%
10/14/2005	Count of Event		2	4			2
	Percentage	0.00%	4.26%	8.51%	0.00%	0.00%	4.26%
10/17/2005	Count of Event		2	8			1
	Percentage	0.00%	4.00%	16.00%	0.00%	0.00%	2.00%
10/18/2005	Count of Event		1	25			16
	Percentage	0.00%	0.64%	16.03%	0.00%	0.00%	10.26%
10/19/2005	Count of Event		1	4			2
	Percentage	0.00%	1.39%	5.56%	0.00%	0.00%	2.78%
10/20/2005	Count of Event			7			13

	Percentage	0.00%	0.00%	5.60%	0.00%	0.00%	10.40%
10/21/2005	Count of Event		4	4			
	Percentage	0.00%	11.43%	11.43%	0.00%	0.00%	0.00%
10/24/2005	Count of Event		2	7			
	Percentage	0.00%	4.17%	14.58%	0.00%	0.00%	0.00%
10/25/2005	Count of Event		2	10			
	Percentage	0.00%	4.08%	20.41%	0.00%	0.00%	0.00%
10/26/2005	Count of Event		1	10	1	1	8
	Percentage	0.00%	0.86%	8.62%	0.86%	0.86%	6.90%
10/31/2005	Count of Event		2	37			11
	Percentage	0.00%	0.85%	15.74%	0.00%	0.00%	4.68%
11/1/2005	Count of Event		1	4			2
	Percentage	0.00%	2.27%	9.09%	0.00%	0.00%	4.55%
11/2/2005	Count of Event		2	15			12
	Percentage	0.00%	0.95%	7.11%	0.00%	0.00%	5.69%
11/3/2005	Count of Event		3	11			
	Percentage	0.00%	2.97%	10.89%	0.00%	0.00%	0.00%
11/4/2005	Count of Event		4	25			18
	Percentage	0.00%	2.17%	13.59%	0.00%	0.00%	9.78%
11/9/2005	Count of Event		1	2			
	Percentage	0.00%	5.88%	11.76%	0.00%	0.00%	0.00%
11/10/2005	Count of Event		2	16			
	Percentage	0.00%	1.89%	15.09%	0.00%	0.00%	0.00%
11/14/2005	Count of Event		4	36		1	6
	Percentage	0.00%	2.02%	18.18%	0.00%	0.51%	3.03%
11/15/2005	Count of Event			16			8
	Percentage	0.00%	0.00%	13.56%	0.00%	0.00%	6.78%
11/16/2005	Count of Event		1	9			
	Percentage	0.00%	1.20%	10.84%	0.00%	0.00%	0.00%
11/17/2005	Count of Event			9			
	Percentage	0.00%	0.00%	26.47%	0.00%	0.00%	0.00%
11/18/2005	Count of Event	1	2	14			2
	Percentage	1.25%	2.50%	17.50%	0.00%	0.00%	2.50%
11/21/2005	Count of Event			16			3
	Percentage	0.00%	0.00%	15.24%	0.00%	0.00%	2.86%
11/22/2005	Count of Event		3	6			2
	Percentage	0.00%	3.26%	6.52%	0.00%	0.00%	2.17%
11/28/2005	Count of Event		1	14			4
	Percentage	0.00%	0.54%	7.61%	0.00%	0.00%	2.17%
11/29/2005	Count of Event		1	7			4
	Percentage	0.00%	1.54%	10.77%	0.00%	0.00%	6.15%
11/30/2005	Count of Event			8			
	Percentage	0.00%	0.00%	10.96%	0.00%	0.00%	0.00%
12/1/2005	Count of Event		1	6			5
	Percentage	0.00%	1.19%	7.14%	0.00%	0.00%	5.95%
12/2/2005	Count of Event		1	5			2
	Percentage	0.00%	1.19%	5.95%	0.00%	0.00%	2.38%
12/5/2005	Count of Event		1	2			
	Percentage	0.00%	0.90%	1.80%	0.00%	0.00%	0.00%

12/12/2005	Count of Event		2	1			3
	Percentage	0.00%	4.26%	2.13%	0.00%	0.00%	6.38%
Total Count of Event		1	50	391	2	2	149
Total Percentage		0.03%	1.48%	11.55%	0.06%	0.06%	4.40%

Date	Data	Forward	Inspector_Activate	Inspector_Close	Item_Send	Open
10/6/2005	Count of Event				1	
	Percentage	0.00%	0.00%	0.00%	5.88%	0.00%
10/7/2005	Count of Event	1		1	2	1
	Percentage	1.00%	0.00%	1.00%	2.00%	1.00%
10/11/2005	Count of Event		1	2	5	5
	Percentage	0.00%	0.88%	1.75%	4.39%	4.39%
10/12/2005	Count of Event		1	1		
	Percentage	0.00%	3.85%	3.85%	0.00%	0.00%
10/13/2005	Count of Event	2	10	2	7	2
	Percentage	1.16%	5.78%	1.16%	4.05%	1.16%
10/14/2005	Count of Event				1	
	Percentage	0.00%	0.00%	0.00%	2.13%	0.00%
10/17/2005	Count of Event			1	2	1
	Percentage	0.00%	0.00%	2.00%	4.00%	2.00%
10/18/2005	Count of Event		4	2	2	2
	Percentage	0.00%	2.56%	1.28%	1.28%	1.28%
10/19/2005	Count of Event	1	3	2	3	2
	Percentage	1.39%	4.17%	2.78%	4.17%	2.78%
10/20/2005	Count of Event	1	3	2	8	1
	Percentage	0.80%	2.40%	1.60%	6.40%	0.80%
10/21/2005	Count of Event		1		1	
	Percentage	0.00%	2.86%	0.00%	2.86%	0.00%
10/24/2005	Count of Event				1	
	Percentage	0.00%	0.00%	0.00%	2.08%	0.00%
10/25/2005	Count of Event	1			1	
	Percentage	2.04%	0.00%	0.00%	2.04%	0.00%
10/26/2005	Count of Event	1	2	7	5	7
	Percentage	0.86%	1.72%	6.03%	4.31%	6.03%
10/31/2005	Count of Event	4	12	5	13	1
	Percentage	1.70%	5.11%	2.13%	5.53%	0.43%
11/1/2005	Count of Event				1	
	Percentage	0.00%	0.00%	0.00%	2.27%	0.00%
11/2/2005	Count of Event		10	23	4	22
	Percentage	0.00%	4.74%	10.90%	1.90%	10.43%
11/3/2005	Count of Event		1	3	7	3
	Percentage	0.00%	0.99%	2.97%	6.93%	2.97%
11/4/2005	Count of Event	1	5	2	7	2
	Percentage	0.54%	2.72%	1.09%	3.80%	1.09%
11/9/2005	Count of Event				2	
	Percentage	0.00%	0.00%	0.00%	11.76%	0.00%
11/10/2005	Count of Event		2	2	4	2
	Percentage	0.00%	1.89%	1.89%	3.77%	1.89%

11/14/2005	Count of Event	2	15	2	7	2
	Percentage	1.01%	7.58%	1.01%	3.54%	1.01%
11/15/2005	Count of Event		8	4	4	3
	Percentage	0.00%	6.78%	3.39%	3.39%	2.54%
11/16/2005	Count of Event		4		3	
	Percentage	0.00%	4.82%	0.00%	3.61%	0.00%
11/17/2005	Count of Event				1	
	Percentage	0.00%	0.00%	0.00%	2.94%	0.00%
11/18/2005	Count of Event				1	
	Percentage	0.00%	0.00%	0.00%	1.25%	0.00%
11/21/2005	Count of Event	1	1	3		3
	Percentage	0.95%	0.95%	2.86%	0.00%	2.86%
11/22/2005	Count of Event	1	7	3	2	2
	Percentage	1.09%	7.61%	3.26%	2.17%	2.17%
11/28/2005	Count of Event	1	20	5	6	2
	Percentage	0.54%	10.87%	2.72%	3.26%	1.09%
11/29/2005	Count of Event	1	4	1	3	1
	Percentage	1.54%	6.15%	1.54%	4.62%	1.54%
11/30/2005	Count of Event		8		6	1
	Percentage	0.00%	10.96%	0.00%	8.22%	1.37%
12/1/2005	Count of Event		11	2	2	1
	Percentage	0.00%	13.10%	2.38%	2.38%	1.19%
12/2/2005	Count of Event	3	1	4	2	
	Percentage	3.57%	1.19%	4.76%	2.38%	0.00%
12/5/2005	Count of Event		3		5	
	Percentage	0.00%	2.70%	0.00%	4.50%	0.00%
12/12/2005	Count of Event		6	4	2	1
	Percentage	0.00%	12.77%	8.51%	4.26%	2.13%
Total Count of Event		21	143	83	121	67
Total Percentage		0.62%	4.23%	2.45%	3.58%	1.98%

Date	Data	Preview	Print	Purge_Deleted_Messages	Reply
10/6/2005	Count of Event	5			
	Percentage	29.41%	0.00%	0.00%	0.00%
10/7/2005	Count of Event	61		1	1
	Percentage	61.00%	0.00%	1.00%	1.00%
10/11/2005	Count of Event	50		1	1
	Percentage	43.86%	0.00%	0.88%	0.88%
10/12/2005	Count of Event	11	1	1	
	Percentage	42.31%	3.85%	3.85%	0.00%
10/13/2005	Count of Event	86			2
	Percentage	49.71%	0.00%	0.00%	1.16%
10/14/2005	Count of Event	29		1	1
	Percentage	61.70%	0.00%	2.13%	2.13%
10/17/2005	Count of Event	19			
	Percentage	38.00%	0.00%	0.00%	0.00%
10/18/2005	Count of Event	66	1	1	2
	Percentage	42.31%	0.64%	0.64%	1.28%

10/19/2005	Count of Event	41			1
	Percentage	56.94%	0.00%	0.00%	1.39%
10/20/2005	Count of Event	66			3
	Percentage	52.80%	0.00%	0.00%	2.40%
10/21/2005	Count of Event	17			1
	Percentage	48.57%	0.00%	0.00%	2.86%
10/24/2005	Count of Event	27			
	Percentage	56.25%	0.00%	0.00%	0.00%
10/25/2005	Count of Event	22	1		
	Percentage	44.90%	2.04%	0.00%	0.00%
10/26/2005	Count of Event	47			4
	Percentage	40.52%	0.00%	0.00%	3.45%
10/31/2005	Count of Event	83	4		8
	Percentage	35.32%	1.70%	0.00%	3.40%
11/1/2005	Count of Event	30			1
	Percentage	68.18%	0.00%	0.00%	2.27%
11/2/2005	Count of Event	82	2		3
	Percentage	38.86%	0.95%	0.00%	1.42%
11/3/2005	Count of Event	45			5
	Percentage	44.55%	0.00%	0.00%	4.95%
11/4/2005	Count of Event	76			4
	Percentage	41.30%	0.00%	0.00%	2.17%
11/9/2005	Count of Event	6			2
	Percentage	35.29%	0.00%	0.00%	11.76%
11/10/2005	Count of Event	54			1
	Percentage	50.94%	0.00%	0.00%	0.94%
11/14/2005	Count of Event	60			
	Percentage	30.30%	0.00%	0.00%	0.00%
11/15/2005	Count of Event	44			3
	Percentage	37.29%	0.00%	0.00%	2.54%
11/16/2005	Count of Event	46			1
	Percentage	55.42%	0.00%	0.00%	1.20%
11/17/2005	Count of Event	13			1
	Percentage	38.24%	0.00%	0.00%	2.94%
11/18/2005	Count of Event	33	1		1
	Percentage	41.25%	1.25%	0.00%	1.25%
11/21/2005	Count of Event	54			
	Percentage	51.43%	0.00%	0.00%	0.00%
11/22/2005	Count of Event	43			1
	Percentage	46.74%	0.00%	0.00%	1.09%
11/28/2005	Count of Event	76	2		3
	Percentage	41.30%	1.09%	0.00%	1.63%
11/29/2005	Count of Event	30			1
	Percentage	46.15%	0.00%	0.00%	1.54%
11/30/2005	Count of Event	24			1
	Percentage	32.88%	0.00%	0.00%	1.37%
12/1/2005	Count of Event	39			2
	Percentage	46.43%	0.00%	0.00%	2.38%
12/2/2005	Count of Event	55			2

	Percentage	65.48%	0.00%		0.00%	2.38%
12/5/2005	Count of Event	86				3
	Percentage	77.48%	0.00%		0.00%	2.70%
12/12/2005	Count of Event	14				2
	Percentage	29.79%	0.00%		0.00%	4.26%
Total Count of Event		1540	12		5	61
Total Percentage		45.51%	0.35%		0.15%	1.80%

Date	Data	Reply_to_All	Save_As	Undo	Grand Total
10/6/2005	Count of Event				17
	Percentage	0.00%	0.00%	0.00%	100.00%
10/7/2005	Count of Event				100
	Percentage	0.00%	0.00%	0.00%	100.00%
10/11/2005	Count of Event	1			114
	Percentage	0.88%	0.00%	0.00%	100.00%
10/12/2005	Count of Event				26
	Percentage	0.00%	0.00%	0.00%	100.00%
10/13/2005	Count of Event	1			173
	Percentage	0.58%	0.00%	0.00%	100.00%
10/14/2005	Count of Event				47
	Percentage	0.00%	0.00%	0.00%	100.00%
10/17/2005	Count of Event	1			50
	Percentage	2.00%	0.00%	0.00%	100.00%
10/18/2005	Count of Event				156
	Percentage	0.00%	0.00%	0.00%	100.00%
10/19/2005	Count of Event				72
	Percentage	0.00%	0.00%	0.00%	100.00%
10/20/2005	Count of Event	3			125
	Percentage	2.40%	0.00%	0.00%	100.00%
10/21/2005	Count of Event				35
	Percentage	0.00%	0.00%	0.00%	100.00%
10/24/2005	Count of Event	1			48
	Percentage	2.08%	0.00%	0.00%	100.00%
10/25/2005	Count of Event				49
	Percentage	0.00%	0.00%	0.00%	100.00%
10/26/2005	Count of Event	1			116
	Percentage	0.86%	0.00%	0.00%	100.00%
10/31/2005	Count of Event	2			235
	Percentage	0.85%	0.00%	0.00%	100.00%
11/1/2005	Count of Event				44
	Percentage	0.00%	0.00%	0.00%	100.00%
11/2/2005	Count of Event	2			211
	Percentage	0.95%	0.00%	0.00%	100.00%
11/3/2005	Count of Event	2			101
	Percentage	1.98%	0.00%	0.00%	100.00%
11/4/2005	Count of Event		1		184
	Percentage	0.00%	0.54%	0.00%	100.00%
11/9/2005	Count of Event				17

	Percentage	0.00%	0.00%	0.00%	100.00%
11/10/2005	Count of Event	2			106
	Percentage	1.89%	0.00%	0.00%	100.00%
11/14/2005	Count of Event	1		2	198
	Percentage	0.51%	0.00%	1.01%	100.00%
11/15/2005	Count of Event				118
	Percentage	0.00%	0.00%	0.00%	100.00%
11/16/2005	Count of Event			2	83
	Percentage	0.00%	0.00%	2.41%	100.00%
11/17/2005	Count of Event				34
	Percentage	0.00%	0.00%	0.00%	100.00%
11/18/2005	Count of Event				80
	Percentage	0.00%	0.00%	0.00%	100.00%
11/21/2005	Count of Event				105
	Percentage	0.00%	0.00%	0.00%	100.00%
11/22/2005	Count of Event				92
	Percentage	0.00%	0.00%	0.00%	100.00%
11/28/2005	Count of Event	2		2	184
	Percentage	1.09%	0.00%	1.09%	100.00%
11/29/2005	Count of Event				65
	Percentage	0.00%	0.00%	0.00%	100.00%
11/30/2005	Count of Event	2			73
	Percentage	2.74%	0.00%	0.00%	100.00%
12/1/2005	Count of Event				84
	Percentage	0.00%	0.00%	0.00%	100.00%
12/2/2005	Count of Event				84
	Percentage	0.00%	0.00%	0.00%	100.00%
12/5/2005	Count of Event				111
	Percentage	0.00%	0.00%	0.00%	100.00%
12/12/2005	Count of Event	3			47
	Percentage	6.38%	0.00%	0.00%	100.00%
Total Count of Event		24	1	6	3384
Total Percentage		0.71%	0.03%	0.18%	100.00%
		Total Days Data Collected			35

7.1.7 *User7 Pivot Table*

Date	Data	Attachment_Read	Compose_New_Mail	Connect	Deactivate	Delete	Disconnect
10/13/2005	Count of Event		2	1	11	4	
	Percentage	0.00%	1.74%	0.87%	9.57%	3.48%	0.00%
10/14/2005	Count of Event	1	1	4	24		4
	Percentage	0.55%	0.55%	2.20%	13.19%	0.00%	2.20%
10/17/2005	Count of Event	1		3	13		4
	Percentage	2.22%	0.00%	6.67%	28.89%	0.00%	8.89%
10/18/2005	Count of Event		4	3	29	2	3
	Percentage	0.00%	2.74%	2.05%	19.86%	1.37%	2.05%

10/19/2005	Count of Event		1	1	21		1
	Percentage	0.00%	1.20%	1.20%	25.30%	0.00%	1.20%
10/20/2005	Count of Event			1	18		
	Percentage	0.00%	0.00%	2.04%	36.73%	0.00%	0.00%
10/21/2005	Count of Event			1	19		2
	Percentage	0.00%	0.00%	1.33%	25.33%	0.00%	2.67%
10/27/2005	Count of Event		3	2	59	1	1
	Percentage	0.00%	1.58%	1.05%	31.05%	0.53%	0.53%
10/28/2005	Count of Event				8		
	Percentage	0.00%	0.00%	0.00%	27.59%	0.00%	0.00%
10/31/2005	Count of Event				6		1
	Percentage	0.00%	0.00%	0.00%	18.18%	0.00%	3.03%
11/1/2005	Count of Event			1	6		
	Percentage	0.00%	0.00%	4.55%	27.27%	0.00%	0.00%
11/2/2005	Count of Event			3	19		4
	Percentage	0.00%	0.00%	5.00%	31.67%	0.00%	6.67%
11/3/2005	Count of Event	1		1	25		
	Percentage	1.20%	0.00%	1.20%	30.12%	0.00%	0.00%
11/4/2005	Count of Event			1	16	1	1
	Percentage	0.00%	0.00%	1.45%	23.19%	1.45%	1.45%
11/7/2005	Count of Event				8		
	Percentage	0.00%	0.00%	0.00%	29.63%	0.00%	0.00%
11/8/2005	Count of Event		1	1	16		1
	Percentage	0.00%	1.61%	1.61%	25.81%	0.00%	1.61%
11/9/2005	Count of Event		1		23		
	Percentage	0.00%	1.11%	0.00%	25.56%	0.00%	0.00%
11/10/2005	Count of Event	2	5	2	25	5	2
	Percentage	1.11%	2.78%	1.11%	13.89%	2.78%	1.11%
12/1/2005	Count of Event	4		1	32	40	1
	Percentage	0.90%	0.00%	0.23%	7.22%	9.03%	0.23%
12/2/2005	Count of Event				43	6	1
	Percentage	0.00%	0.00%	0.00%	19.55%	2.73%	0.45%
12/5/2005	Count of Event	3	2	1	50	4	
	Percentage	1.21%	0.81%	0.40%	20.16%	1.61%	0.00%
12/6/2005	Count of Event			2	10		2
	Percentage	0.00%	0.00%	6.25%	31.25%	0.00%	6.25%
12/7/2005	Count of Event		2	3	27	2	3
	Percentage	0.00%	1.79%	2.68%	24.11%	1.79%	2.68%
12/8/2005	Count of Event		1	1	38	5	1
	Percentage	0.00%	0.50%	0.50%	18.81%	2.48%	0.50%
12/9/2005	Count of Event				15		
	Percentage	0.00%	0.00%	0.00%	42.86%	0.00%	0.00%
12/12/2005	Count of Event				16	6	
	Percentage	0.00%	0.00%	0.00%	14.95%	5.61%	0.00%
Total Count of Event		12	23	33	577	76	32
Total Percentage		0.41%	0.78%	1.12%	19.63%	2.59%	1.09%

Date	Data	Explorer_Activate	Folder_Add	Folder_Switch	Forward	Inspector_Activate

10/13/2005	Count of Event	7		17		4
	Percentage	6.09%	0.00%	14.78%	0.00%	3.48%
10/14/2005	Count of Event	22		3		2
	Percentage	12.09%	0.00%	1.65%	0.00%	1.10%
10/17/2005	Count of Event	14		1		
	Percentage	31.11%	0.00%	2.22%	0.00%	0.00%
10/18/2005	Count of Event	23		17		9
	Percentage	15.75%	0.00%	11.64%	0.00%	6.16%
10/19/2005	Count of Event	10		8		11
	Percentage	12.05%	0.00%	9.64%	0.00%	13.25%
10/20/2005	Count of Event	17		2		
	Percentage	34.69%	0.00%	4.08%	0.00%	0.00%
10/21/2005	Count of Event	17		8		4
	Percentage	22.67%	0.00%	10.67%	0.00%	5.33%
10/27/2005	Count of Event	33		11		26
	Percentage	17.37%	0.00%	5.79%	0.00%	13.68%
10/28/2005	Count of Event	8		1		
	Percentage	27.59%	0.00%	3.45%	0.00%	0.00%
10/31/2005	Count of Event	7		5		
	Percentage	21.21%	0.00%	15.15%	0.00%	0.00%
11/1/2005	Count of Event	5				2
	Percentage	22.73%	0.00%	0.00%	0.00%	9.09%
11/2/2005	Count of Event	19		3		
	Percentage	31.67%	0.00%	5.00%	0.00%	0.00%
11/3/2005	Count of Event	22		8		2
	Percentage	26.51%	0.00%	9.64%	0.00%	2.41%
11/4/2005	Count of Event	15		1	1	1
	Percentage	21.74%	0.00%	1.45%	1.45%	1.45%
11/7/2005	Count of Event	8		4		
	Percentage	29.63%	0.00%	14.81%	0.00%	0.00%
11/8/2005	Count of Event	16		7		
	Percentage	25.81%	0.00%	11.29%	0.00%	0.00%
11/9/2005	Count of Event	22		7		2
	Percentage	24.44%	0.00%	7.78%	0.00%	2.22%
11/10/2005	Count of Event	22		20		5
	Percentage	12.22%	0.00%	11.11%	0.00%	2.78%
12/1/2005	Count of Event	28		3		5
	Percentage	6.32%	0.00%	0.68%	0.00%	1.13%
12/2/2005	Count of Event	28		13	1	23
	Percentage	12.73%	0.00%	5.91%	0.45%	10.45%
12/5/2005	Count of Event	41		8	3	11
	Percentage	16.53%	0.00%	3.23%	1.21%	4.44%
12/6/2005	Count of Event	10				
	Percentage	31.25%	0.00%	0.00%	0.00%	0.00%
12/7/2005	Count of Event	23		9		5
	Percentage	20.54%	0.00%	8.04%	0.00%	4.46%
12/8/2005	Count of Event	29	5	28		8
	Percentage	14.36%	2.48%	13.86%	0.00%	3.96%
12/9/2005	Count of Event	15				

	Percentage	42.86%	0.00%	0.00%	0.00%	0.00%
12/12/2005	Count of Event	15				2
	Percentage	14.02%	0.00%	0.00%	0.00%	1.87%
Total Count of Event		476	5	184	5	122
Total Percentage		16.20%	0.17%	6.26%	0.17%	4.15%

Date	Data	Inspector_Close	Item_Send	Open	Preview	Print	Purge_Deleted_Messages
10/13/2005	Count of Event	2	5	2	57		
	Percentage	1.74%	4.35%	1.74%	49.57%	0.00%	0.00%
10/14/2005	Count of Event	1	3		113		1
	Percentage	0.55%	1.65%	0.00%	62.09%	0.00%	0.55%
10/17/2005	Count of Event	1		1	7		
	Percentage	2.22%	0.00%	2.22%	15.56%	0.00%	0.00%
10/18/2005	Count of Event	5	7	4	33		
	Percentage	3.42%	4.79%	2.74%	22.60%	0.00%	0.00%
10/19/2005	Count of Event	1	2	1	24		
	Percentage	1.20%	2.41%	1.20%	28.92%	0.00%	0.00%
10/20/2005	Count of Event				11		
	Percentage	0.00%	0.00%	0.00%	22.45%	0.00%	0.00%
10/21/2005	Count of Event		1		22		
	Percentage	0.00%	1.33%	0.00%	29.33%	0.00%	0.00%
10/27/2005	Count of Event	1	4	1	44		2
	Percentage	0.53%	2.11%	0.53%	23.16%	0.00%	1.05%
10/28/2005	Count of Event				12		
	Percentage	0.00%	0.00%	0.00%	41.38%	0.00%	0.00%
10/31/2005	Count of Event	1	1	1	10		
	Percentage	3.03%	3.03%	3.03%	30.30%	0.00%	0.00%
11/1/2005	Count of Event	1		1	5		
	Percentage	4.55%	0.00%	4.55%	22.73%	0.00%	0.00%
11/2/2005	Count of Event		1		9		1
	Percentage	0.00%	1.67%	0.00%	15.00%	0.00%	1.67%
11/3/2005	Count of Event		1		21		
	Percentage	0.00%	1.20%	0.00%	25.30%	0.00%	0.00%
11/4/2005	Count of Event		2		28	1	
	Percentage	0.00%	2.90%	0.00%	40.58%	1.45%	0.00%
11/7/2005	Count of Event				7		
	Percentage	0.00%	0.00%	0.00%	25.93%	0.00%	0.00%
11/8/2005	Count of Event		1		19		
	Percentage	0.00%	1.61%	0.00%	30.65%	0.00%	0.00%
11/9/2005	Count of Event		2		28		
	Percentage	0.00%	2.22%	0.00%	31.11%	0.00%	0.00%
11/10/2005	Count of Event	9	9	8	60		
	Percentage	5.00%	5.00%	4.44%	33.33%	0.00%	0.00%
12/1/2005	Count of Event	2	2	1	318		2
	Percentage	0.45%	0.45%	0.23%	71.78%	0.00%	0.45%
12/2/2005	Count of Event	6	4	5	81		
	Percentage	2.73%	1.82%	2.27%	36.82%	0.00%	0.00%
12/5/2005	Count of Event	2	7		111		

	Percentage	0.81%	2.82%	0.00%	44.76%	0.00%	0.00%
12/6/2005	Count of Event				8		
	Percentage	0.00%	0.00%	0.00%	25.00%	0.00%	0.00%
12/7/2005	Count of Event	1	3	1	32		
	Percentage	0.89%	2.68%	0.89%	28.57%	0.00%	0.00%
12/8/2005	Count of Event	4	1	4	75		1
	Percentage	1.98%	0.50%	1.98%	37.13%	0.00%	0.50%
12/9/2005	Count of Event				5		
	Percentage	0.00%	0.00%	0.00%	14.29%	0.00%	0.00%
12/12/2005	Count of Event				67		
	Percentage	0.00%	0.00%	0.00%	62.62%	0.00%	0.00%
Total Count of Event		37	56	30	1207	1	7
Total Percentage		1.26%	1.91%	1.02%	41.07%	0.03%	0.24%

Date	Data	Reply	Reply_to_All	Save	Undo	Grand Total
10/13/2005	Count of Event	3				115
	Percentage	2.61%	0.00%	0.00%	0.00%	100.00%
10/14/2005	Count of Event	3				182
	Percentage	1.65%	0.00%	0.00%	0.00%	100.00%
10/17/2005	Count of Event					45
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/18/2005	Count of Event	3		3	1	146
	Percentage	2.05%	0.00%	2.05%	0.68%	100.00%
10/19/2005	Count of Event	1			1	83
	Percentage	1.20%	0.00%	0.00%	1.20%	100.00%
10/20/2005	Count of Event					49
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/21/2005	Count of Event		1			75
	Percentage	0.00%	1.33%	0.00%	0.00%	100.00%
10/27/2005	Count of Event	1			1	190
	Percentage	0.53%	0.00%	0.00%	0.53%	100.00%
10/28/2005	Count of Event					29
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
10/31/2005	Count of Event	1				33
	Percentage	3.03%	0.00%	0.00%	0.00%	100.00%
11/1/2005	Count of Event				1	22
	Percentage	0.00%	0.00%	0.00%	4.55%	100.00%
11/2/2005	Count of Event	1				60
	Percentage	1.67%	0.00%	0.00%	0.00%	100.00%
11/3/2005	Count of Event	1		1		83
	Percentage	1.20%	0.00%	1.20%	0.00%	100.00%
11/4/2005	Count of Event	1				69

	Percentage	1.45%	0.00%	0.00%	0.00%	100.00%
11/7/2005	Count of Event					27
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
11/8/2005	Count of Event					62
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
11/9/2005	Count of Event	1		2	2	90
	Percentage	1.11%	0.00%	2.22%	2.22%	100.00%
11/10/2005	Count of Event	5		1		180
	Percentage	2.78%	0.00%	0.56%	0.00%	100.00%
12/1/2005	Count of Event	3			1	443
	Percentage	0.68%	0.00%	0.00%	0.23%	100.00%
12/2/2005	Count of Event	3	1		5	220
	Percentage	1.36%	0.45%	0.00%	2.27%	100.00%
12/5/2005	Count of Event	3	1		1	248
	Percentage	1.21%	0.40%	0.00%	0.40%	100.00%
12/6/2005	Count of Event					32
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
12/7/2005	Count of Event	1				112
	Percentage	0.89%	0.00%	0.00%	0.00%	100.00%
12/8/2005	Count of Event				1	202
	Percentage	0.00%	0.00%	0.00%	0.50%	100.00%
12/9/2005	Count of Event					35
	Percentage	0.00%	0.00%	0.00%	0.00%	100.00%
12/12/2005	Count of Event	1				107
	Percentage	0.93%	0.00%	0.00%	0.00%	100.00%
Total Count of Event		32	3	7	14	2939
Total Percentage		1.09%	0.10%	0.24%	0.48%	100.00%

Total Days Data Collected 26

Chapter 8

REFERENCES

1. Schwartau, W. Make Security Personal [WWW Document]. URL http://www.nwfusion.com/columnists/2004/092704schwartau.html (visited 1/27/2006).

2. Lemos, R. Counting the cost of slammer [WWW Document]. URL http://news.com.com/Counting+the+cost+of+Slammer/2100-1001_3-982955.html (visited 1/27/2006).

3. Turner, D. Symantec Internet Security Threat Report. Sept. 2004. Volume VI.

4. Bush, G. Executive Order on Critical Infrastructure Protection [WWW Document]. URL http://www.whitehouse.gov/news/releases/2001/10/20011016-12.html (visited 1/27/2006).

5. AXcessNews. Intrusion Detection Product Revenue Expected to climb in 2004 [WWW Document]. URL http://www.axcessnews.com/technology_030704a.shtml (visited 1/27/2006).

6. Bace, R. An Introduction to Intrusion Detection and Assessment for System and Network Security Management. April 1999.

7. Rabek, J. LARIAT Windows Traffic Generation. Feb. 24, 2003.

8. Information Assurance Technology. Information Assurance Technology Analysis Center (IATAC) [WWW Document]. URL http://iac.dtic.mil/iac_dir/IATAC.html (Visited 1/27/2006).

9. Haines, J.W., Rossey, L.M., Lippmann, R.P., Cunningham, R.K. Extending the DARPA Off-Line Intrusion Detection Evaluations. DARPA Information Survivability Conference and Exposition (DISCEX) II. 2001. Anaheim, CA: IEEE Computer Society.

10. Rossey, L. LARIAT Overview and Capabilities. Feb. 24, 2003.

11. Thurrott, P. OS Market Share: Microsoft Stomps the Competition [WWW Document]. URL http://www.winnetmag.com/Article/ArticleID/40481/40481.html (Visited 1/27/2006).

12. Hamm, S., Burrows, P. Why High Tech has to Stay Humble [WWW Document]. URL http://www.businessweek.com/magazine/content/04_03/b3866081_mz063.htm (Visited 1/27/2006).

13. MXLogic, Advanced Email Defense. Email Defense Industry Statistics [WWW Document]. URL http://www.mxlogic.com/PDFs/IndustryStats.pdf (visited 12/7/04).

14. Directions on Microsoft. Jul. 2003: Exchange Server 2003, Outlook 2003 Enhance Mobility, Scalability, Security [WWW Document]. URL http://www.directionsonmicrosoft.com/sample/DOMIS/research/2003/07jul/0703i.htm (visited 12/7/04).

15. Microsoft Developer Network. Cutting Edge: Windows Hooks in the .NET Framework [WWW Document]. URL http://msdn.microsoft.com/msdnmag/issues/02/10/CuttingEdge/ (Visited 12/7/04).

16. Barabasi, A.-L. (2002). Linked: The new science of networks. Cambridge, MA: Perseus Publishing.

17. Ebel, H. Mielsch, L.-I., Bornholdt, S. (2002). Scale-free topology of email networks. Physical Review, E 66.

18. ITFacts.biz. Email [WWW Document]. http://www.itfacts.biz/index.php?id=C0_8_1 (Visited 11/17/2004).

19. Motulsky, H. (1995). Intuitive Biostatistics. NY: Oxford University Press.