Virtual Reality Based End-User Assessment Tool for Remote Product/System Testing and Support

Burcu Dolunay

A Thesis

in

the Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

May 2006

Your file Votre référence
ISBN: 978-0-494-34589-4
Our file Notre référence
ISBN: 978-0-494-34589-4

ABSTRACT

Virtual Reality Based End-User Assessment Tool For Remote Product / System Testing and Support

Burcu Dolunay

Today, it is known that a through understanding of the end-user is the most valuable information to improve design, training, maintenance and assembly/disassembly processes for products or systems. The most widely used method for this purpose, user experiments, requires a product prototype, a test environment and a researcher to watch and collect the data. This results in a procedure that is time consuming and limited by geographical constraints. Although remote testing is being used today by recording or transmitting the test situations, these still depend on audio or video data that require a researcher to watch and analyze. Therefore there is a need for tools that collect and analyze user data automatically from actual interactions of the user with the product. This thesis proposes an approach to achieving that goal. The approach proposed integrates existing virtual reality technology with a four-phase analysis method that uses techniques from data mining and human-computer interaction researches. In our approach, we first cluster subjects into clusters such that subjects with similar performance are placed into the same cluster. Then, for each cluster, we extract paths that have been followed by users during their interaction with the product. Finally we demonstrate these paths together with the statistical results obtained for a cluster. As result the researchers see if the users are following the paths that they were expected to follow, what are the common paths followed by users with a better or relatively poor performance, which aspects of the design can be changed to direct the users to better paths or how can the users be trained better to follow desired paths which leads to easiest, safest, and most effective working conditions. The thesis describes the model constructed for the approach and then presents an example application: simulation study in the evaluation of a

simple assembly process. Initial results support the usefulness of method as an automated tool to detect problems.

Dedicated to my family, Hatice, Kamil and Ebru Dolunay...

ACKNOWLEDGEMENTS

This thesis would not have been possible without Dr. Ali Akgunduz, my thesis supervisor. I would like give him my special thanks for his invaluable guidance, academic and financial support throughout this research. I am forever grateful to him for his endless understanding and patience.

I would like to thank Dr. Brandon Gordon, Dr. Mingyuan Chen and Dr. Yong Zeng for taking part in my thesis committee and providing valuable comments for my further research.

I wish to express my deepest gratitude to my parents, Kamil and Hatice Dolunay and to my sister Ebru Dolunay, for their never ending support and affection. I would be much less than what I am today without their support.

I wish to thank my dearest *"L'honneur"* for being with me and being so supportive throughout my last years. I am thankful to my friends for their care and friendship.

Burcu Dolunay

May,2006

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

DMU	Digital Mock-up
HCI	Human Computer Interaction
HPA	Hypertext Probabilistic Automaton
HMD	Head Mounted Display
VE	Virtual Environment
VR	Virtual Reality
3D	Three Dimensional

CHAPTER 1

1 INTRODUCTION

Today designers and researchers are continuously seeking optimal products design solutions that lead to more user friendly products which are safer to use, easier to maintain, and easier to assembly and disassembly. Designing such products is only possible by the collaborative participation of both designers and end-users of the products (customers) in the design process. In other words understanding the interaction of end-users with any product/system is an essential step in design and/or improvement of products or systems. This is predominantly achieved through usability tests where designers/experts observe the user performing some tasks with the product and gathers information about the problems that arise etc. However, such data collection technique is only possible once a prototype of the product is build.

Data collected during these tests reveals the difficulties customer had with the product, whether they could accomplish a task within certain criteria, the functionalities that were utilized most often, the aspects that they liked and their comments about the product. This data provides the designers with insight about the aspects that are strong or that need improvement as well as information about most frequently used features so that limited resources can be used in the optimization of these features. Analysis of product usage data is also important after the design stage, since ongoing data collection during the life of product is the most important way to detect changes in the usage patterns of costumers.

So in all stages of a product life cycle, superior understanding of user behaviour and evaluation of product performance is one of the key steps in research and development

activities. Increasing the quality and efficiency of this process, in other words providing high-quality and more efficient tools for user assessment is of great importance.

1.1 PROBLEM AREA

First and the vital stage of understanding user-product interaction is to come up with a prototype of the product and create the suitable test environment to bring the prototype and customer together for evaluation purposes. Although, designers can work on the details of a product before its prototype is built, gathering information from the customers usually requires some level of working prototype.

In traditional test environments while users perform certain given tasks with the system, testers observe them to detect problems in the product. Through these tests on the actual prototypes under the supervisions of the testers (observers), crucial user data can be collected for further analysis. However the problem with this kind of tests is that it is based only on the observational data. Furthermore an experienced design team member is required each time a test is performed to observe and interpret the test session. Product assessment tests, performed using physical prototypes introduce high-cost and consume long time during the product evaluation process. Another major drawback of such practices is that only a small number of subjects, limited to certain geography and time scale can be assessed.

However, today with the improvements in virtual reality technology, prototypes for products come in both physical and virtual forms which make it possible to run remote tests through networked virtual systems. Even supporting VR technologies like haptic devices that require certain settings are being pushed towards low-end 3D models that can be distributed over the

internet. These virtual prototypes and remote tests fulfill an important need in the sense that they enable to evaluate products or systems in a geographically wide area where products may have several different potential customer groups, or companies have product development in several places. Since 3D computer models are less expensive, more flexible and demand less time compared to physical mockups, the popularity of VR for prototyping is increasing. Current trends show that VR will be taken on board in place of physical prototypes in most design and manufacturing facilities in the near future.

In remote usability tests where subjects use virtual prototypes of new products, the results can be either interactively followed through communication network or they can be recorded for afterwards analyzes. However many remote testing settings depend on recorded or transmitted video and audio records of the test situation (Kuutti et al., 2001, Hammontree et al., 1994). This means although the test is conducted in a computer environment the data analysis still requires an expert interaction which is still time consuming and prevents the application of test to a larger number of subjects. Moreover, observational data analysis performed without software support tends to analyze only a few dimensions of the world. On the other hand if several dimensions are being captured electronically, investigators will have a better position to look deeper on the behavioural influences on the product being designed (Sanderson et al., 1994). Multi-dimensional analysis may also reveal behavioural patterns that designers are not aware off. Yet, the problem with automated data collection and the analysis of log files is that since a log record system can not work as a filter like a human observer does; there generally exists a huge amount of data for analysis to extract useful information. In other words there are no mechanisms to imitate filtering, analysis and interpretation capabilities of a natural observer to detect usability, assembly, disassembly, and maintenance problems in products or processes. Hence there are needs for methodologies that analyze

observational data without sacrificing from the accuracy when statistical summaries are attained, yet minimize the necessity of human interactions during data collection and analysis process.

1.2 MOTIVATION AND GOAL

Design and assessment of an automated data collection system that records the user's interactions with the proposed product design into log files and a smart data processing tool that examines these files in order to derive meaningful insights about the designed product for its usability, maintainability and ease-of-assembly etc. are interests of this thesis. The main objective of this study is to propose a virtual reality based information collection and analysis method to understand user behaviour over the designed product and suggest appropriate design changes to better comply with the various user groups so the final product is easy to use, maintain or assemble. This kind of an approach is important in the sense that it enables designers to gather data from a much larger sample-space without time and location limitations, also at a considerably lower cost.

1.3 SOLUTION APPROACH

The proposed approach in this research work draws on two main efforts for the above problem: i) design of a virtual reality based prototype testing simulator (e.g. a prototype, an assembly or a maintenance scenario) and conduct experimental studies with various end-users to capture human interaction with the designed product in terms of sequence of events; ii) development of an intelligent and automated data analysis technique to extract useful qualitative and quantitative information from voluminous raw data.

Today, with the help of advanced computer visualization systems, designing necessary simulation environments and collecting data are not a challenge. VR has already been successfully applied to the simulation of many complex systems such as art, medicine, entertainment, communications, computer science, space exploration, etc. In most of the VR applications, interactions of the subject with the environment, users actions can be readily captured. There are numerous existing and emerging studies that enable tracking and recognizing various gestures, eye or head movements and voice. Hence the current VR technology is capable of capturing the interactions of users with the virtual environment at the highest details using various available components. The captured data provides comprehensive information to search, find, count or analyze the system. There are also available synchronization and search techniques to combine data from different resources like video, logs, voice etc to provide supplementary information. Hence current VR technology is capable of capturing the interactions of users with the virtual environment at the highest level of details using various available components.

Today, the greater challenge is in controlling, analyzing and deriving meaningful conclusions from the available vast amount of data. When it comes to the analysis stage, there are not widely accepted generally applicable techniques. While techniques to collect data in all details have advanced to an important point, automated methods to analyze observational data have not kept pace. In the proposed work, our main objective is to propose better methods to collect data and analyze them to help product/system designers, so that they can better address the needs and requirements of the end-users. We address this problem by putting emphasis on smart analysis of voluminous observational data. For this purpose, we propose use of methods from the field of human computer interaction studies in conjunction with data mining techniques. However we are not using techniques from data mining as a net to cast into the

data in the hope to draw all the "possible fish". Rather we are seeking the answer for the question "whether the system is functioning in the way it is supposed to be functioning in the real world?" and "are there any emerging behavioural differences between different user profiles?". Because, in the design of all products or systems there are certain paths that designers assume end-users of the system would follow to perform the desired functionalities of the product/system. However most of the time users do not follow the paths they were supposed to follow and at least several different ways emerge to accomplish a certain task. The improvements or changes in the products/systems are done according to these implicit assumption or predictions like a shortcut button to a certain menu or a high friction cover to a certain area of a product from where the users are supposed to grab the product during the course of action. It happens quite often that people do not like to use or even understand the utility of a very accessible feature in a product/system and even sometimes it bothers them having to go through it every time they are trying to accomplish a different task. Thus a good way of understanding end-users behaviours or identifying problematic parts in a system is to know the paths that users follow to perform certain tasks, or the paths that enable highest level of effectiveness in terms of time and energy usage. Furthermore identifying the characteristics of users who follow similar patterns during the task provide valuable insights to the designers. This kind of information brings in the possibility of better differentiation in product types, more efficient marketing policies for different groups of end users or in case of complex products or systems better maintenance and training opportunities.

In this work, to accomplish our goal of finding answers to above questions; we are integrating several existing methods from different fields. The way we handle the problem of data mining in user or system evaluation is different in the sense that, we do not focus on an esoteric problem or algorithm in data mining context to come up with an improvement like an

improved solution or a different version. Instead we start from a need for a tool with certain functionalities and figure that these functionalities call for methods from different disciplines, thus we come up with an application of these different techniques for our purposes. Also we propose a method for path comparison between designers and users.

To sum up, the developed methodology aims to incorporate various approaches, with the objective of using knowledge extracted from one step of analysis to feed the execution of subsequent step. As a whole, this research focuses on discovering and displaying usability issues with any system for which we have user interaction data. To the best of our knowledge; both tools from human-computer interaction area and data mining domains were not being used in the test or analysis of a physical product or system before as utilized in the proposed research work.

1.4 SCOPE OF THIS THESIS

End user evaluation is a very general problem and also requires different applications for different fields. In the definition presented here we will summarize and draw the frame for our approach explained above.

In summary, this thesis describes an automated usage data analysis approach using techniques from data mining and human computer interaction. The approach introduces integration of existing virtual reality and data capturing techniques with automated data analysis in order to facilitate large scale and remote assessment of products and/or systems to improve design and after sales support like maintenance, assembly or disassembly of the product etc. It relies on the idea that understanding behavioural patterns besides statistical and quantitative records is

the key to more insightful automated usability testing. To extract user behaviour patterns, this work suggests use of two traversal pattern discovery techniques and introduces a sequence comparison method to find user actions that deviate from expected process model. At the end, an experiment is conducted to apply proposed approach. A virtual assembly environment is constructed using 3D graphics to collect data and later this data is analyzed by the proposed techniques. Results and evaluations are presented.

1.5 CONTRIBUTIONS

The contributions of this thesis are as follows. We:

- propose the use of virtual reality techniques to automate data analysis phase in usability tests
- propose the adaptation of techniques from data mining and human computer interaction fields to the domain of physical product/system testing
- propose and implement an algorithm to find deviations between the expectations of designers and actual usage data
- construct an experimental set up for the application of proposed methodology and verify the applicability of techniques

The techniques developed in this research are applicable to any system where usage data can be collected and can be used both in different stages of a product's life cycle with different purposes.

1.6 ROADMAP FOR THE THESIS

This thesis is organized as follows:

- Chapter 2 provides basic information about the VR and other related fields to our study. Later in the literature review, previous related studies on methodology are presented.
- Chapter 3 presents our approach in detail. Four phases followed in the model are described in sections 3.1, 3.2, 3.3 and 3.4. The chapter concludes with a general model for the approach which is a big picture of explained methodology and shows the data flow between the phases.
- Chapter 4 presents the experiment we carried out to verify the effectiveness of the methodology to detect usability problems. Designed experiment is an assembly simulation, which is carried in two steps. Overall description and objective of the experiment is described in section 4.1 while section 4.2 explains the experimental set-up, section 4.3 summarizes the results together with the evaluation of the collected information (data).
- Chapter 5 concludes with final thoughts and evaluations about the study, provides ideas about the potential application areas and directions and future work we can get from this research work.

CHAPTER 2

2 BACKGROUND AND RELATED WORK

As explained previously, the methodology proposed in this research work has two aspects: first developing a virtual reality based automated data collection scheme; and second developing an automated data analysis technique to mine useful information in data. There is a rich literature in each of these areas. Therefore we present background and literature review in two sections focusing on one of the aspects in each section. Section 2.1 gives some background on virtual reality and its use in product testing, section 2.2 provides background on automated data analysis in end user evaluation and finally section 2.3 presents literature review.

2.1 VIRTUAL REALITY IN DESIGN AND MAINTENANCE

Virtual reality is the generation of a real or fictive media in computer environment. It provides three-dimensional visualization of systems as well as interaction with these systems by means of tools like head-mounted displays, gloves or tactile feedback devices. Originally the term virtual reality was used to describe fully immersive virtual systems where users were completely embedded in computer generated virtual worlds. Today this term is also used for systems that are partly immersive or that simply provide 3D visualization. In non-immersive virtual reality users use a normal monitor, and manipulate the virtual environment using a keyboard, a mouse, a joystick or gloves or a similar input device. Useful applications of VR include training in numerous areas like military, medical, operation of complex equipment;

virtual prototyping for design evaluation; human factors and ergonomic studies; simulation of assembly sequences and maintenance tasks; architectural walk-through; medical studies etc.

In product design, use of virtual reality technologies implies creating virtual prototypes of products called digital mock-ups (DMUs) instead of physical mock-ups and/or capturing product user interaction using tracking devices. A virtual prototype most simply provides a three dimensional visualization. This will enables designers to share the vision of the design at very early stages and identify problematic points. So far virtual prototypes have been employed in various practical applications and scientific tests and also their validity have been studied in numerous researches. Related studies are reviewed in literature review. Below figure 2.1 shows an example for the physical prototype, virtual prototype and remote test environment for a product called "bugsyphone" that was used in a study by Kuutti et al. (2001).

Figure 2.1: Picture 1 is the physical prototype of "Bugsy phone", Picture 2 is the virtual prototype created using VRML and Picture 2 shows an instance of unguided remote test using virtual prototype.

Virtual encounter of users and products and data capturing techniques are becoming more effective with the availability of advanced techniques and tools like data-gloves, head mounted displays (HMD), tactile and force feedback devices etc. Today there are numerous commercial off-the-shelf hardware and software available for companies that try to implement virtual reality in their product testing processes. There are studies called "user centered virtual prototyping" that make use of VR technology for improvement of human-computer interaction in product testing, to create interactive three-dimensional stimulus feedback mechanisms, by means of which all natural human behaviour can be recorded. It specifically aims to record and measure naturalistic behaviour in product testing without the use of real product or a physical prototype (Kempter, 2005).

Use of virtual prototypes brings the advantage of carrying out product tests at earlier stages of the product design. Today the data collection using virtual prototypes is not a challenge in the field of product testing since existing devices are capable of capturing data in required detail. Moreover, with continuing improvements in both VR components and intelligent data-processing techniques like geometric reasoning, constraint management and dynamics simulation techniques, virtual prototyping has the potential of being one of the most powerful interactive analysis tools in the near future.

2.2 AUTOMATED ANALYSIS OF USABILITY DATA

Once user interaction with product/system is captured next step is the analysis of data to extract useful information. This is the problem of analyzing sequential data to evaluate a system or product without massive human effort.

There are two main fields that deal with automated analysis of sequential data for evaluation purposes: i) human computer interaction; and ii) data mining. The mainstream studies for automated analysis of usability data come from human-computer interaction (HCI) area and deal with assessment of software interfaces. Automated analysis has been an important need for softwares, since capturing user's interactions is very straightforward however analysis is difficult due to the user interface events that are very detailed and large in volume. Usability data is actually a term used in HCI to describe any information that is useful in measuring the usability attributes of a system (Hilbert et al., 2000). For computer systems usability is defined by two factors i) its ability to enable users to accomplish their objectives efficiently ii) level of satisfaction system provides to the users during their interaction with the system (Ivory et al., 2001) In this thesis, we will use the term usability data to mention the data that is captured during a test session of a user with the virtual product or system under evaluation. In a broader definition of usability, Lecerof and Paterno, (1998) state that usability includes the following aspects:

- The relevance: if the system meets users needs.
- The efficiency: how efficient is the system for users to perform their objectives with the system.
- The users' satisfaction: feelings about the system after using
- The learnability: how user friendly is the system to learn and use both for the first and consecutive trials
- The safety: how error-proof is the system

So the objective of usability evaluation is to measure the performance of system on all or some of these components. The importance of these components changes from one system to

another. In usability evaluation, designers generally aim at measuring the aspects that are more critical in their systems.

Hilbert et al. (2000) group the usability evaluation approaches in three main categories: predictive evaluation, participative evaluation and observational evaluation. In predictive evaluation experts make predictions about the performance of the system based on their past experiences. Psychological techniques like cognitive walk-through which help experts to understand users' experiences with the system are another method of groups used for predictive evaluation. So this technique does not involve actual users. The second group, participative evaluation involves collection of information directly from users using surveys etc. Finally observational evaluation involves measuring usability by observing the users while they are working with the system. In the next phase, the collected data is analyzed. In this research we are interested in observational evaluations: we are extending their use to physical products by employing virtual reality technology and propose a methodology for the analysis of collected data. According to Hilbert et al. (2000), techniques for dealing with data in HCI field can be divided into 8 main categories depending on purposes.

1. Techniques for synchronization and searching collected data,
2. Techniques for selecting, abstracting, and recoding event streams to support human and automated analysis (including counts, summary statistics, pattern detection, comparison, and characterization,
3. Techniques for performing counts and summary statistics.
4. Techniques for detecting sequences.
5. Techniques for comparing sequences.
6. Techniques for characterizing sequences.
7. Visualization techniques

8. Integrated evaluation support.

A detailed review and description of these techniques can be found in Hilbert et al. (2000). Among them techniques that perform counting and collecting summary statistics and techniques for detecting, comparing and characterizing sequences are most relevant to our objective and reviewed in depth later in this chapter.

Techniques developed for evaluation of software provide very good tools for product/system evaluation and are applicable in the evaluation of all products that have a user interface. However the usability data that is collected from a virtual prototype of a physical product is different in structure than software interface data. Hence there exists a need for tools that enable designers to perform usability tests on virtual prototypes. In order to achieve the desired goals, we use techniques from the second related field, data mining.

Other related field, data mining, or in other words knowledge-discovery in databases, is the study of automatically searching large volumes of data for hidden useful information. In order to achieve this, data mining uses techniques from various fields like statistics, software engineering and artificial intelligence Data mining is one of the most extensively studied subjects in recent years and has been proven to be useful. There are numerous methodologies of data mining like decision trees, association rules, Bayesian data analysis, prediction techniques, hidden Markov processes and sequential pattern mining techniques, clustering and factor analysis (Hand et al., 2001).

To the best of our knowledge, data mining methods are not used for analysis of behavioural sequential data in product testing. In our work we use techniques from clustering and

sequential pattern mining. Clustering can be defined as the process of grouping a collection of N data points into numerous segments or clusters based on a suitable measurement of similarity among data points (Ghosh, 2003).

Clustering is the primary goal for applications such as customer and product segmentation but more often it is an intermediate step needed for further analysis. Sequential pattern mining is also an extensively studied area in data mining since in many applications data is represented in the form of sequences. Biomedical studies, performance analysis, customer profiles and consumer behaviour are among the important application domains for sequential pattern mining.

2.3 LITERATURE REVIEW

2.3.1 Applications of Virtual Reality

There are numerous studies and applications in the literature where virtual reality systems are used in different steps of product or system development. These include virtual prototyping for product development, product testing (Kuutti et al., 2001), training, collaboratively reviewing design ideas and also virtual reality models as a form of representation in place of physical models specifically in sectors where building physical models are very expensive and not efficient like furniture industry and architectural design (Oh et al., 2004). Milkova and Ikonomov (2003) introduced a scheme for constructing VR simulation for product design. In their study they affirm that by these VR simulations end-user tests can be carried out during design, manufacturing, maintenance for the end-user and recycle state, and provide feedback to reduce design errors. The overall objective is to contribute for a better design. A question that comes to mind here is if the results of tests

performed by using virtual prototypes are reliable compared to physical prototypes. There are studies regarding that question in the literature that show that the predictive power of virtual-prototypes based concept testing provides nearly the same results as physical prototypes (Dahan et al., 1998). VR features offer users the opportunity to explore virtual objects at a high level of detail that are appropriate for activity evaluation (Hurwicz, 2000). Gamberini et al. (2003) presents an experimental study using virtual fire scenarios. Their work concludes that people show realistic responses when dangerous situations are simulated in VEs. This work proves that VR is suitable for emergency simulations and for user training purposes. Figure 2.2 shows the VR equipment used in this experiment.

Figure 2.2: Figure on the left is a view from inside the building where the fire started and figure on the right is the view of subject navigating using VR tools (Gamberini et al., 2003).

Moreover, in the VR based maintenance and training studies, interactive training that is delivered through a computer has been reported to be more effective, leading to reduced training time and costs when compared with the traditional classroom lectures (Fletcher, 1996). These findings were later supported by the results obtained from studies in several

other areas like manufacturing industry, surgery, space and naval and avionics trainings (Stone, 2001). Finally in their study Kempter et al. (2003) has shown that Virtual Reality and other related technologies allow complete performance recording thus prevents loss of important user data, which is a common negative aspect of traditional pen-pencil based observational methods.

2.3.2 Automated Data Analysis Techniques

Our goal is to propose an automated analysis approach to understand the behaviours of users of a product or system from the data that was collected using virtual prototypes. If we go through existing literature for automated product or system evaluation techniques, we see that several methodologies for evaluating systems or products have been developed up to date. However they are generally specific-purpose studies that aim at analyzing some numerical results like smart data mining system for drop test analysis of electronic products (Zhou et al., 2001), data mining in software metrics to support and enhance researchers' understandings of these metrics and their effect on quality (Dick et al.,2004).The study by Zhao et al. (2004) proposes a new technique based on parallel coordinate visualization and introduce a distance-based technique so that interesting patterns in test results can be detected visually. They also develop a query mechanism so that the researcher can search for a pattern. There is one study by Muehleisen (1996) which is different from the others in the sense that it proposes to make use of data that is retrieved from customer support call tracking database for usability evaluation. The assumption in here is that, there is a usability data resting unutilized in these recorded user calls databases and they are all about certain usability issues. In this study statistical methods are used to calculate certain metrics or measures. It is important to note that, in all above studies, the data mining focuses on extracting or visualizing patterns in

numerical data, namely in quantitative measurements recorded during the test. In our approach we are interested more in the behavioural analysis of the event data in other words patterns followed by users. To the best of our knowledge, there are not available techniques or tools that automate data collection by using VR and mine for both quantitative and qualitative information in the data for system or product evaluation.

So far automation of usage data analysis has been studied and applied by human computer interaction researchers in evaluating user interfaces. Automation has been predominantly used in two ways: automated capture of data and automated analysis of data (Ivory et al., 2001). Automated analysis support involves analysis of log files to extract useful information. There have been different approaches to this problem such as counts and summary statistics, detection and comparison of sequences, generating abstract models from logs to find out the sequential structures in the logs, visualized presentation techniques of the data and integrated support tools that include all or some of the listed methods (Hilbert et al., 2000). Many tools making use of these methods have been developed and some of them are currently in use. An extensive review of literature for tools using either one or several of these approaches can be found in Hilbert et al. (2000) and Ivory et al. (2001). In this thesis, we review two of these tools and some studies that use one of the methods which are closely related to the approaches used in this research work, namely metric based analysis or pattern comparison. Metric-based analysis generates quantitative results from the data mainly by statistical methods like reporting average performance time, error frequency etc. The work by Sanderson et al. (1994) is an example to the tools that provides this kind of statistical information. It is important to note that their software is not developed for only software interface evaluation but for all kinds of "exploratory sequential data analysis" which is defined as an attempt to analyze systems, environments or observational data in which the sequential order of events are taken

into consideration in analysis (Sanderson et al., 1994). So when recording data the order of events are saved. Their software tool (MACSHAPA) provides investigators diversified ways to visualize their data as well as means for collecting statistical reports. It also provides a pattern matching functionality by aligning the actual logs with the process model and displaying the mismatches in the spreadsheet that includes logs. Structure of this tool is shown in figure 2.3. In the figure arrows from the centre show information provided by the tool through the use of data in spreadsheet cells whereas arrows to the centre show data entry and modification functionalities of the tool.

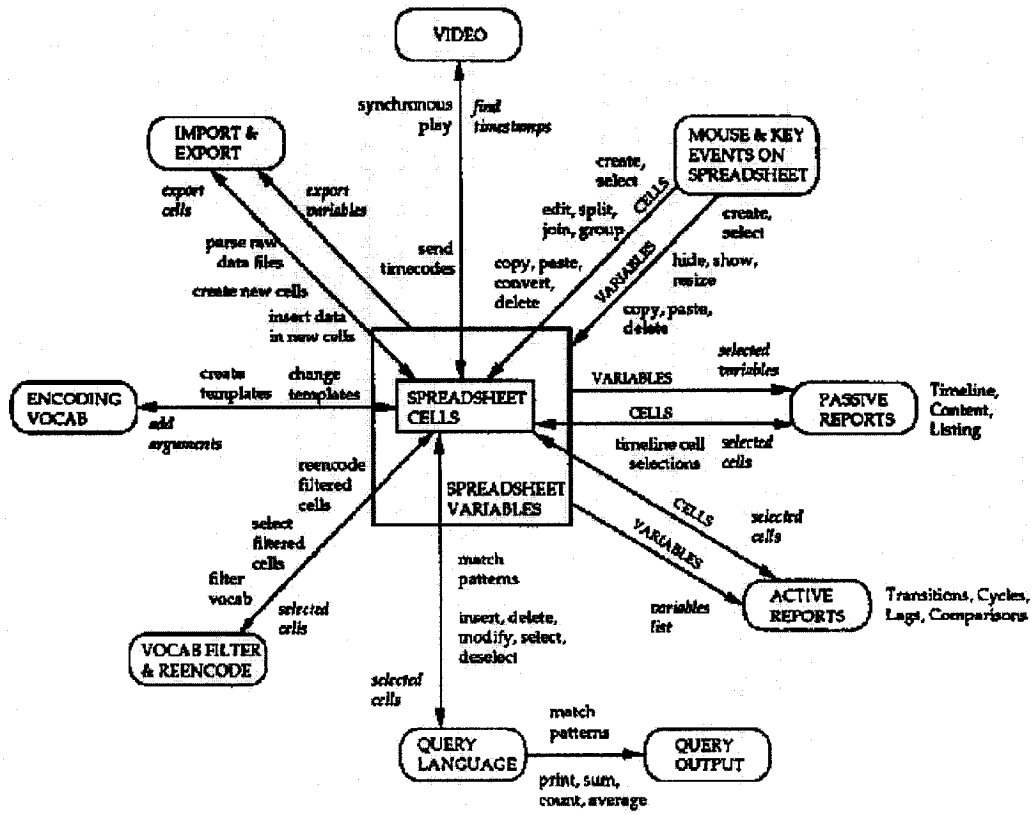

Figure 2.3: Structure of tool MACSHAPA proposed by Sanderson et al., 1994.

Another tool developed by Lecerof and Paterno (1998) provides automatic support for usability evaluation of software interfaces; includes pattern comparison where user logs like

key strokes are first transformed into process events and then compared to the previously prepared process model looking for violations of prerequisites, completed or failed tasks and errors. Then the tool generates reports of these deviations from the model. The results also include an analysis of tasks which are successfully completed, those that are failed or never tried, user errors and their type and execution time. These results are viewed on an interface which is illustrated in figure 2.4. The interface provides several built-in statistical methods and graph drawing functionalities to display results. However, in this study the researcher can observe only user patterns that deviate from process model in other words when there is a rule or precondition violation. Thus the application is not a data mining procedure but rather a scan of user logs for errors.

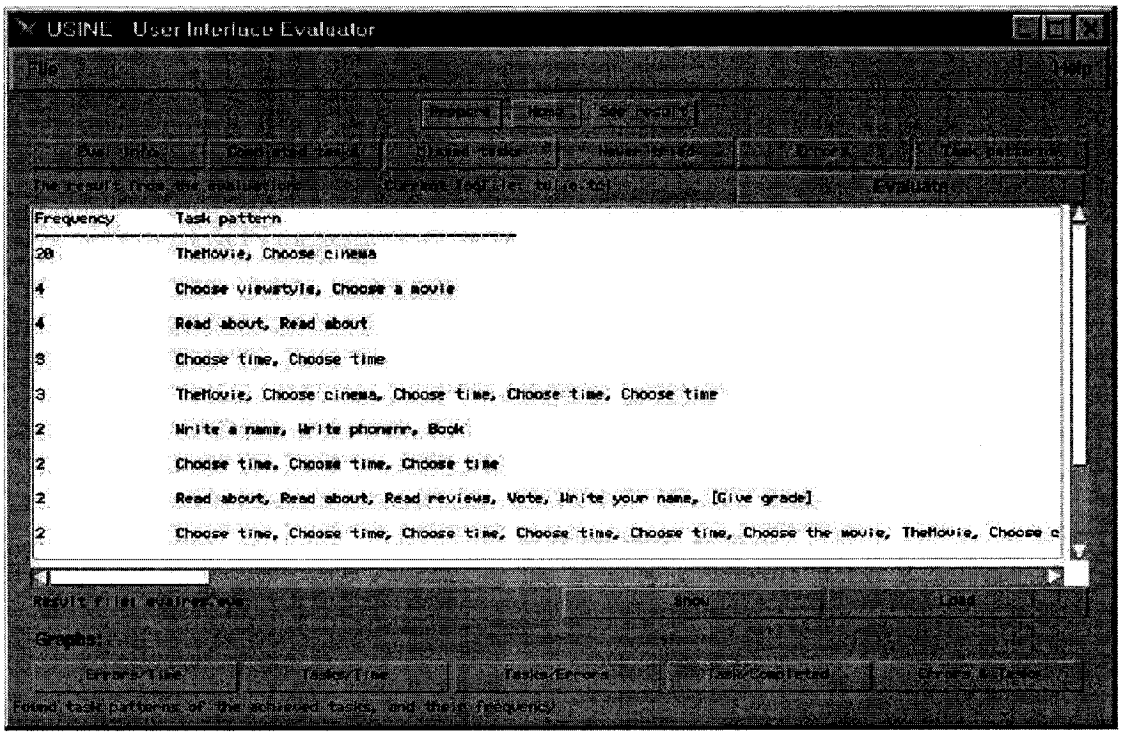

Figure 2.4: Interface of evaluation tool (USINE).

In some other studies like Hilbert and Redmiles (1998) and Zettlemoyer et al. (1998) smart modules called 'agents' were employed to activate and report in case of unexpected user actions. The main idea in these studies is that when developers design systems they have implicit and explicit assumptions about the way the system functions and develop the systems in the light of these assumptions. Thus, it is of a great significance to detect and resolve mismatches between developers' and users' way of dealing with the interface. In Hilbert and Redmiles (1998) these mismatches are reported back to developers. In Maximal Repeating Patterns Analysis (Siochi and Ehrich, 1990), repeating patterns in the logs were mined with the idea that they will provide useful insight about problematic points or most often used paths.

Having one or two common ideas with each of these studies the approach we propose here, we do not take the logs as they are but rather find patterns in logs of users in a cluster since comparing logs can be erroneous due to noise in data. We need a method for event abstraction or in other words for extracting meaningful patterns from a large amount of data. In the process of extracting traversal paths from user logs our approach gets closer to sequential patterns mining problem in data mining. Discovering patterns in a sequence of events has been studied extensively in data mining literature for different purposes. In their works, Agrawal and Srikant (1995) propose Apriori Algorithms, where they mine sequential purchase patterns of customers from a large database of purchased items. In this study, potential patterns in item purchase databases were mined and then the patterns that repeated no less than the thresh-hold frequency were listed. In their problem definition the purchases need not be contiguous to form patterns thus requirements listed in Apriori algorithms do not cover most of the sequential path mining needs. However, there are numerous methodologies based on this study for contiguous pattern mining like methods developed to mine web

traversal data in the field of web usage mining. When a user goes back and forth through pages in a web site, these transactions are stored in a variety of places. This stored data forms web usage data and there is a research discipline with the purpose of discovering patterns in this data which is called web usage mining. Its objective is to discover interesting usage patterns, by analyzing the mentioned log data (Karampatziakis et al., 2004). The structure of data collected from web sites and especially traversal pattern mining problem as in the form studied in this discipline provides potentially good techniques for us to use in our approach. In mining web usage data, numerous algorithms have been developed from principles of Apriori algorithms (Agrawal and Srikant, 1995). These algorithms all depend on these two principles: finding all potential sequential patterns in the database then choose the patterns that have support higher than the minimum threshold value. Support in this context is the number of times or frequency of a sequence appearing in the database. There are two studies whose problem definition is close to ours. First one is the work of Chen et al. (1998) where they proposed an algorithm to convert the original sequence of web log data into a set of "maximal forward references" which implies the longest path ever followed by the user without any interruption or cancellation. The main idea is to exclude the effects of pages that are revisited because of their place on the site rather than the content, so they eliminated backward references and repetitions in forming paths. Also a sequence must not be a subsequence of any other sequence to be maximal. In another study, Dunham and Xiao (2001) proposed an algorithm to find traversal patterns in the web usage logs by keeping repetitions and backward moves in paths. Their study is about mining "maximal frequent sequences", which means a pattern can have repetitions and backward traversals. Yet, to be maximal, it must not be a subsequence of any other sequence. Both studies also propose methodologies to improve the efficiency of the scanning process that follows after the extraction of the potential paths to find the patterns with a minimal required frequency. A comparison of Apriori algorithms,

Maximal Forward sequences and maximal frequent sequences are given below (Dunham and Xiao, 2001).

Table 2.1: Comparison of Apriori like web usage mining algorithms to mine traversal pattern

	Ordering	Repetitions	Maximal
Apriori algorithms	N	N	N
Maximal Forward Sequences (Chen et al.)	Y	N	Y
Maximal Frequent Sequences (Xiao and Dunham)	Y	Y	Y

Another approach to extract user behaviour from usage data is to deduce a grammar from logs. For grammar deduction, there have been studies that propose different methods depending on the complexity and characteristics of the system. First –order Markov chains are used to construct probabilistic models for web navigation sessions called Hypertext Probabilistic Automaton (HPA) (Borges, 2000). In constructing Markov models from the user sessions, the frequency of a page is used in calculating its probability of appearing. Similarly, transitions from one page to the other are used in calculating probabilities for different traversals. An example of a user session and its HPA model is given in figure 2.5 (Karampatziakis et al., 2004). Again in the study by Karampatziakis et al. (2004), each user log is assumed to be controlled by the rules of a formal language that is not yet known and each user session is treated as an example of a string belonging to that language. Methodology of using formal languages to infer grammar for a set of strings has been also widely used in biology in DNA sequence analysis. So there is a wide range of methods proposed for

grammar inference from data. Some of the well-known methods are; algorithms that make use of prefix trees to iteratively build the models; statistical inference techniques that make use of Hidden Markov Models and Neural Nets that use neural networks techniques (Cook, 2001)

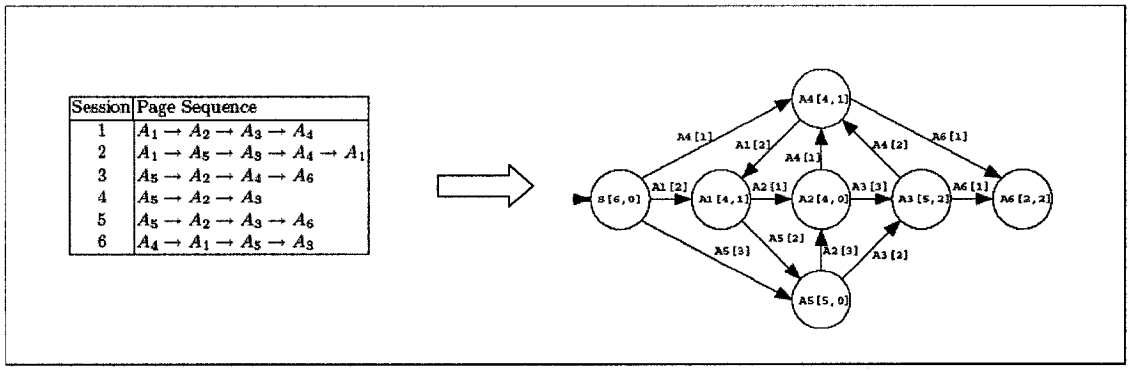

Figure 2.5: Sample user session and its HPA.

Other than web navigation mining studies, in his work, Cook (1996) uses grammar inference approach to discover process models from execution logs and then validates the model by quantitatively finding how good the match is between the model and the logs. Agrawal et al. (1997), presents a methodology to construct process flow diagrams from unstructured past data. In both studies the methodologies were supported by experiments to be useful in constructing process flows for real-life events. Another related field of study is frequent episodes problem in telecommunication alarm analysis where patterns are not event based (Mannila et al., 1997).

As a consequence, considering the wide range of different studies that are related to different parts of the methodology presented in this work, the approach presented here offers distinct advantages in terms of starting point, use of hidden knowledge in the logs and the application area.

CHAPTER 3

3 FOUR-PHASE MODEL

In this work we are concerned with understanding the behaviour of users during their interaction with system/product. We are specifically interested in discovering patterns followed by users and getting most out of usability data both in qualitative and quantitative terms. So the objective is to develop a VR based automated end-user assessment tool that can achieve above functionalities without massive human effort. For this purpose we propose use of two techniques:

i) Capture interactions of the user with the virtual prototype of product/system in form of transcripts

ii) Analyze this data with the objective of answering following questions:

- How high is the performance of users with the system? Does it conform to expected standards?
- What are the problematic points that affect performance and satisfaction?
- What are paths users follow to accomplish certain tasks in the system?
- Do the paths followed by different users match with designer's expectations and with each other?
- What are the patterns that deviate from expected process model?
- What are the performance characteristics of users following a certain path?

In an attempt to answer these questions we suggest a four-phase analysis approach:

i) Automated data collection using log files: Recording user interaction with the system as the set of events.

ii) Scanning log files for potential indicators of performance and obtaining performance parameters for users using statistical techniques and/or other measurements

iii) Clustering users with respect to the observed results

iv) Extracting followed paths in each cluster and comparing these paths with the designer data and/or among user clusters

Our aim is to use the first three phases as a preparation for path comparison phase besides the direct results they provide and then integrate the results at the end. In the rest of the chapter we present each phase in detail and conclude with a general model for the overall process.

3.1 PHASE 1: AUTOMATED DATA COLLECTION

In phase 1, user session data is captured while user exercises with the product/system. All user inputs and relevant system output are recorded in a file from beginning to the end of user session. This forms the log file or transcript for that user. Capturing two kinds of data is required at this stage:

1. User logs and system outputs that define acts
2. Required statistical records or measurements of the session.

In our methodology we adopt an event-based approach. That is, user actions and system outputs are recorded as a set of events where events are well-defined, instantaneous actions that characterize a change in the system. Events are instantaneous in the sense that they need relatively very short time duration to be accomplished like, grabbing an object etc. Activities that span a period of time are represented as the interval between two events. For example the whole session itself is the activity between "begin session" and "end session" events. To preserve all necessary information about a user action, events are typed to the system beforehand and can have attributes. Attributes of an event can be its happening time, or results associated to that event if there are any (e.g. whether it fails or not). So a user session

is a sequence of events, an "event stream" recorded with predetermined attributes together with user identification information and desired statistics about the session such as duration. User attributes if there are any are also recorded in the log files. As a result, at the end of each user session, a log file contains three main pieces of information: User identification number, sequence of events (event stream) for the session and records of the session and user. If we denote the log file of user *i* by R_i; R_i is obtained in the form of:

$$R_i = \langle (i), (E_i), (A_i) \rangle$$

If we write E_i and A_i in an open form:

$$R_i = \langle (i), (e_1, e_2 \ldots\ldots\ldots e_n), (a_1, a_2 \ldots\ldots\ldots a_m) \rangle$$

where

R_i = File for user i

i = Identification number of the user

E_i: $(e_1, e_2 \ldots\ldots\ldots e_n)_i$ = Event stream of user i

A_i : $(a_1, a_2 \ldots\ldots\ldots a_m)_i$ = Set of recorded statistics for the session of user i

a_j = Record j

e_j = jth event in a log file

Thus the input for Phase1 is the user-system interaction and the required equipment is a data recording tool with required built-in functionalities such as time measurement. Output of Phase 1 is a set of user log files (R_i) collected in a file or a simple database as shown in figure 3.1. As explained earlier, numerous tools are available in the market to capture user interactions with a prototype of a system.

$$R_1 = \langle (1), (e_1, e_2 \ldots e_n), (a_1, a_2 \ldots a_m) \rangle$$

$$R_2 = \langle (2), (e_1, e_2 \ldots e_n), (a_1, a_2 \ldots a_m) \rangle$$

$$\vdots$$

$$R_n = \langle (n), (e_1, e_2 \ldots e_n), (a_1, a_2 \ldots a_m) \rangle$$

Figure 3.1: Output of Phase 1

3.2 PHASE 2: PERFORMANCE PARAMETERS

The aim of phase 2 is to determine and calculate the usability related numerical performance parameters from the session log file, so the input of phase 2 comes from log files recorded in phase 1. One of the key benefits of collecting the interaction data with its all details as explained in phase 1 is that we can calculate every numeric value that we think is closely related to the usability of the product. In every system there exist parameters that are good indicators of system or product performance. These parameters give insight on two points:

1. How good a *user* is in the accomplishment of tasks with the product/system compared to the expected performance standards and other users
2. How good the *system-user interaction* is in terms of the potential indicators, i.e., number and frequency of problematic points, users' ease of executing tasks, satisfaction etc.

In HCI literature there are many metrics that were used in studies and fond to be useful to get more insight about the performance of a user when using an interface like task completion time, repetitions, cancellations and undo, the frequency and patterns in use of help, time spent in a window etc (Hilbert et al., 2000). In one study, the analysis of Maximal Repeated

Patterns (MRP) was studied and shown to be useful for finding problems in a complex user interface (Siochi and Ehrich, 1990). Most of these metrics like task completion time, repeating certain tasks frequently or cancelling are valid for other kinds of applications as well as software interfaces or can be easily adapted. Possibilities for alternative indicators are larger with VR systems since these systems are able to keep track of different aspects of interface usage data like the duration and magnitude of pressure on a button coordinates of body parts at a certain time etc. Potentials are increasing as tracking systems advance and enable capturing physiological parameters as well as visual and haptical feedback. For any product or system, the designers collaboratively decide on potential indicators that they think apply best to their situation and construct their data record systems accordingly.

To compute performance parameters, once the test is finished, log file for each subject is scanned and these predetermined metrics are calculated from his/her data. Some of the metrics are taken directly from the collected records set $(A_i : (a_1, a_2 \ldots\ldots\ldots a_m))$ like task completion times. Other metrics may require calculations from the event stream after the session (Ex: number of repetitions for an event). General statistics packages or custom calculation methods can be preferred depending on the need. In the rest of the thesis we will call these predetermined parameters as performance indicators. Table 3.1 shows the performance indicators calculated for each user where I_i is the indicator i and i_{kj} is the value of indicator j for user k.

Table 3.1: Output of Phase 2: User performance indicators

USERS (U_i)	INDICATORS			
	I_1	I_2	$\cdots$	I_m
U_1	i_{11}	i_{12}	$\cdots$	i_{1m}
U_2	i_{21}	i_{22}	$\cdots$	i_{2m}
$\vdots$		$\vdots$		
U_n	i_{n1}	i_{n2}	$\cdots$	i_{nm}

So the output of Phase 2 is an m-dimensional vector containing values of calculated parameters for each user where m is the number of performance indicators to be calculated for the system.

3.3 PHASE 3: CLUSTERING USERS WITH RESPECT TO THE PERFORMANCE INDICATORS

3.3.1 Background on Clustering

The objective of phase 3 is to group subjects based on their performance indicators. In other words, we define groups of users that show high similarity within each other compared to members of other groups in terms of performance indicators. This process is called clustering in data mining and it is one of the most extensively studied subjects in literature.

Clustering can be defined as the process of grouping a collection of N data points into segments or clusters based on a suitable measurement of similarity among data points (Ghosh, 2003). It is a very broad problem since a data point is a very general concept; it can be

numerical values, images, signals, patterns etc. Cluster analysis has long been used in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining. Numerous methods with different properties have been proposed so far and some of them are currently being used.

There is not a precise definition for what conditions are required for a cluster which makes it difficult to come up with a general purpose algorithm. In fact the definition for "well separated clusters" states that: A cluster is a set of points such that any point in a cluster is more similar to any other point in the same cluster than to any point that is in another cluster. However this is not a very applicable definition because in most data sets the points that are close to the boundaries of a cluster get closer to the points on the boundaries of other cluster at least they get further away from some points in their own cluster. So most of the algorithms use "center-based definition" which states that points in a cluster must be closer to the center of that cluster than the center of a different cluster (Kaufman and Rousseeuw, 1990).

Generally following steps are followed in a clustering process:

1. *Data Normalization:* This is the transformation of data to standardized scales before using so all the dimensions will be on the same scale. It is necessary because it is very probable that different aspects are measured on different scales, and in cases of large difference between measured aspects, the dimensions with larger values affect the results of classification as if they have much weight or importance than other dimensions.
2. *Specifying a Distance (or Similarity) Metric:* This is one of most important steps since the determination of similarity metric for the clustering is defining the conditions for a cluster. This step requires integration of knowledge about the

system, data and feature extraction studies to identify the important attributes to consider in a distance metric.

3. *Choice of Clustering Method:* Clustering methods can be divided to two main categories: hierarchical clustering and partitional clustering.

Partitional methods partition the data set into k clusters. K-means and K-medoid methods (Jain and Dubes, 1988) are the most popular representatives of this approach. Given a set of n data points where each data point is a point in d-dimensional space and an initial number of clusters k, these algorithms partition the data points into k clusters. In this process each cluster has a representative data point and distances of all data points in the set from those representatives are minimized. (Ghosh, 2003). A problematic point with this approach is to determine the best value for k. This is done either by making educated guesses about the system, by application requirements or by trying for various values of k and choosing the best k depending on requirements of the system.

Hierarchical methods operate on an input set described in terms of similarity or distance coefficient between each pair of individuals .Then they produce a nested set of partitions, with a single, all inclusive cluster at the top and singleton clusters of individual points at the bottom .The results are presented on a tree-like structure called dendogram as shown in figure 3.2. At the last leaves of a dendogram, each object is a cluster and at the top there is a cluster that contains all the cases. Hierarchical approaches are further divided into two as agglomerative or divisive methods. Divisive methods start from a single cluster containing all objects and end up with the leaves where each cluster contains a single object whereas agglomerative methods starts from numerous clusters and end up with the single cluster that contains all the objects.

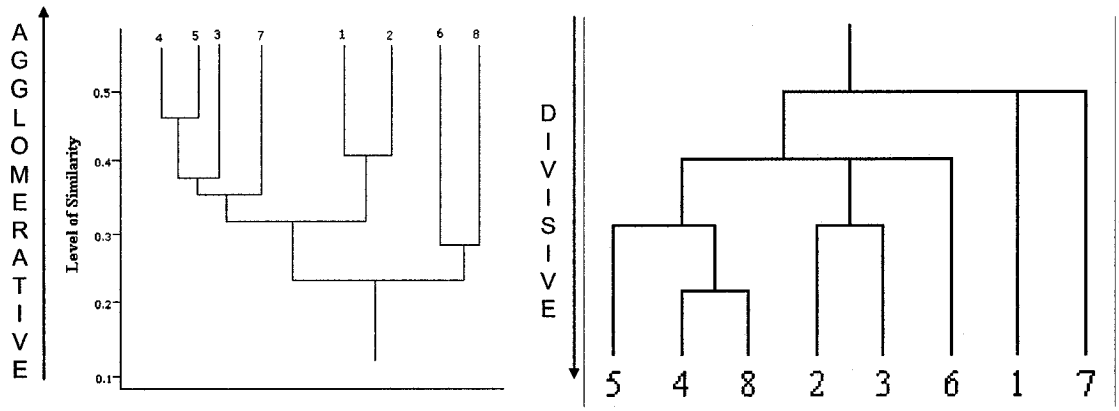

Figure 3.2: Dendogram representation of results for divisive and agglomerative techniques

There are numerous algorithms using each of these approaches and they have specific weaknesses and strengths. The most commonly used algorithms in this class are k-means and k-medoids algorithms. In general the advantages of these algorithms are: they are effective and simple compared to other techniques (time complexity: O (nm) where n is the number of data points and m is the number of dimensions for each data point (Barbara, 2000)) and their weaknesses are: we need to specify k -the number of clusters- in advance, they are very sensitive to noise and outliers and an element belongs to exactly one cluster. If we look at hierarchical clustering techniques, their main advantages are: they are conceptually simple and easy to implement, however the weakness of agglomerative clustering methods is that they have time complexity of at least $O(n^2)$, where n is again the number of data points (Barbara, 2000). Also both of the algorithms above have the risk of ending at local optimums and can be improved by the use of some other techniques like genetic algorithms.

3.3.2 Approach

In our problem:

1. We have a multidimensional indicator vector for each user. If we have n subjects and m indicators for each, then we can represent each subject as a point in m dimensional space.
2. Clustering problem in this context reduces to determining groups with high density around a certain point and finding nearest neighbours of points.
3. We will have a space of numeric attributes and we can use Euclidean distance between points as our similarity measure.

These properties being stated, different methods can be adopted at this phase for different cases depending on the needs of the case, namely the number of subjects and number included dimensions etc. Figure 3.3 shows grouping of 7 subjects with a system of two indicators.

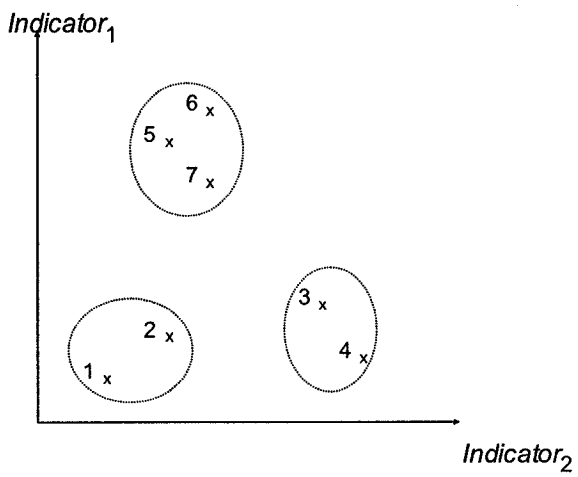

Figure 3.3: Grouping of seven users represented in a two-dimensional attribute space into three clusters

To sum up, the aim of clustering in this phase is to come up with the clusters that represent the sample space in terms of performance indicators in the least costly way. In the outcome different groups of subjects who ends-up with similar results are collected in a cluster. When

comparing emerging patterns with the expected designer patterns or among the subjects, this information provides insight about the relative performance of the users of a certain path. In the end, this information enables designers to consider ways to channel lower performance users to the paths that provide better satisfaction. Moreover, at this step clustering enables us to omit individual 'outliers' in terms of experience and performance.

Once the clusters are determined, log files of members of a cluster are collected in separate files. So if clustering process ends up with k clusters, we have k files containing logs of its member subjects as shown in figure 3.4. In the figure event streams (set E) of subjects in the same cluster are collected in separate files and event stream of a user is the set that contains sequence of events executed by that user.

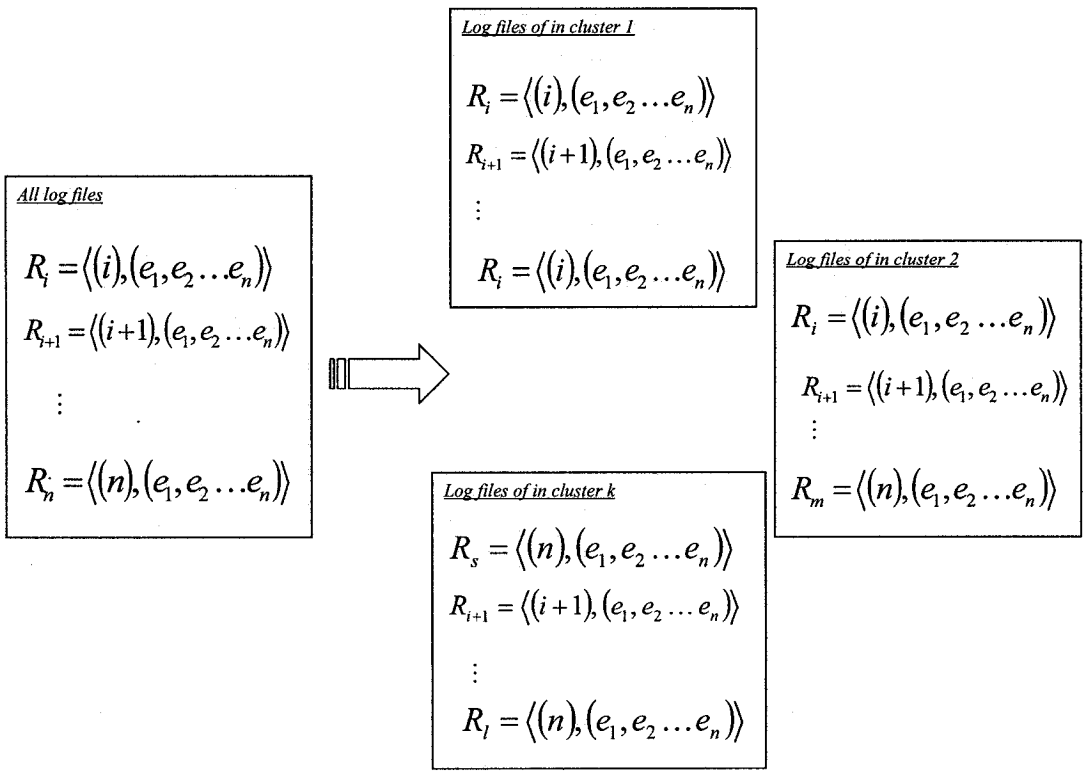

Figure 3.4: Illustration of k-files containing logs of member subjects

3.4 EXTRACTING PATHS

3.4.1 Problem Definition

Once we have the event streams for subjects in cluster files, the next phase to extract traversal paths for each subject from the event streams. Subjects' event streams as they are collected at the end of the experiment are most of the time unnecessarily large and can be misleading due to noise in data. During a test session, a subject works on some tasks. Each of these tasks consists of smaller subtasks that are linked together. In order to accomplish a task, the subject moves among different activities throughout the test. So subjects are likely to traverse activities several times in accordance with the target activity which they are involved at the time. This results in noise in the event stream of a subject as well as meaningful entries. This noise can be interruptions in the paths followed like repetitions, cancellations or some simple basic activities that are necessary to complete a task. However, comparison without filtering can lead to erroneous results, such as giving no common path, since it is improbable that logs of two people are the same even though they follow similar paths. So the problem here is to extract paths that appear to be sufficient enough of interest. Hence we can state the problem as follows:

Given E_i: $((e_1,t_1),(e_2,t_2).....(e_n,t_n))$ set of events for subject *i, where* t_j is the time event *j is executed and* $t_j < t_{j+1}$ *for* $j=1,2,3,.....,n$. Let D_k be the database that includes event streams of subjects in the cluster k. Find traversal paths in this database D_k that repeat at least by the threshold frequency.

3.4.2 Solution Approaches

Approach I

The first approach is casting the problem as one of mining sequential contiguous patterns in noisy data. This problem has been studied extensively in the data mining literature for different purposes. Among the numerous methodologies proposed, most of which are Apriori based approaches. They function in two steps: first finding the traversal paths in each stream, then scanning the path database and choose the paths that are repeated at least by the required number of times. Frequency of a path is called its support and support is the measure of "interestingness" of a path. Apriori algorithms proposed by Agrawal and Srikant, (1995) do not consider order of events thus they are not very suitable to be applied here. However we can adopt their principles as "find the paths for each subject and scan the cluster database for frequent ones" and change properties that define a path considering the features of the system in question. There are studies in web navigation mining field that are good examples to this approach. Most relevant several of these studies, their properties and differences in definitions of paths can be found in detail literature review. We applied the result of one of these studies (Chen et al., 1998) in our experiment which is explained in chapter 4.

The final outcome for this approach is the set of preferred paths as illustrated in figure 3.5.

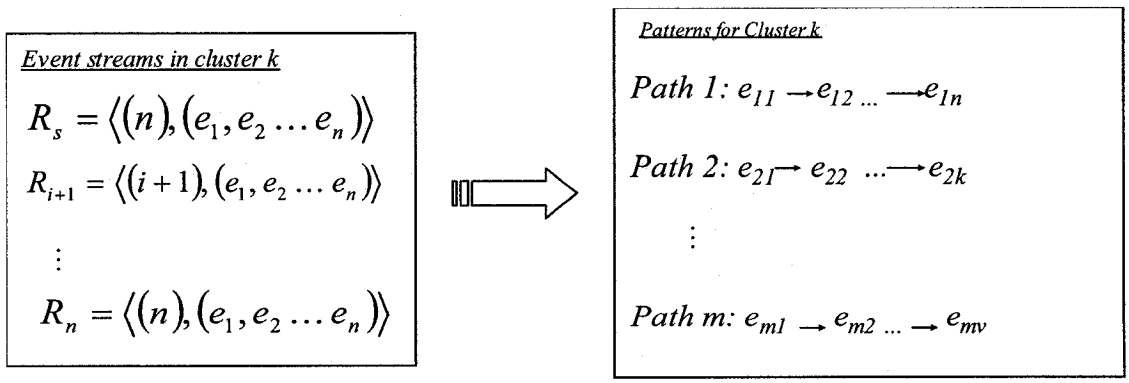

Figure 3.5 : Set of the preferred paths for cluster k mined from the set of event streams

Approach II

Another approach is to build a probabilistic grammar model from the logs or in other words; to deduce a grammar for the user interaction from sets of execution data. This approach is developed based on the method of Hypertext Probabilistic Automata for web navigation mining (Karampatziakis et al., 2004).

Given a set of event streams for cluster k (D_k) the model is constructed as follows:

1. Each event e_i is represented by a state in the model.
2. When a subject executes the event e_j right after the event e_i, this is represented as a transition in the model from e_i to e_j labelled as e_{ij}.
3. Each state and transition's probability is calculated from the repeating number of that state or transition in the event stream,

   p_{ij} = the probability of *eij* (probability of executing event *j* right after event *i*)

Given a process with N events, we obtain a Markov model with N states. Then depending on the desired frequency to accept a transition as a pattern, a threshold probability is determined. Transitions whose probabilities exceed this threshold probability are chosen and the process model is constructed showing these transitions and probabilities associated to them as depicted in figure 3.6.

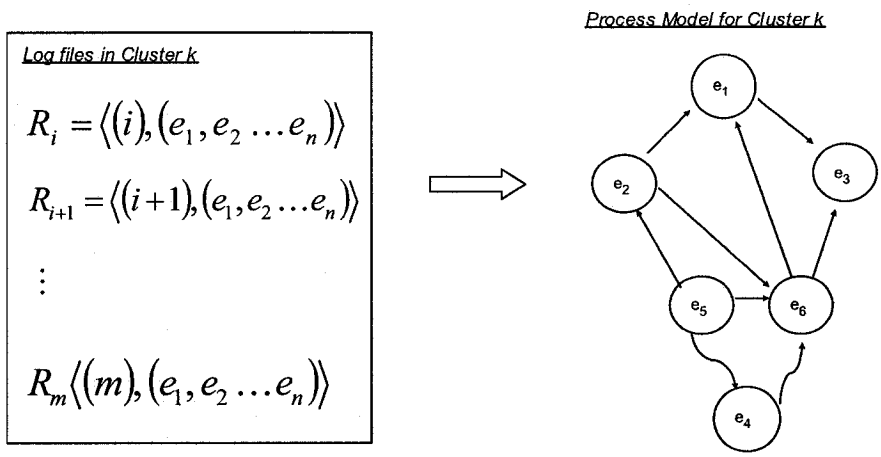

Figure 3.6: Process model for the cluster k

Problem of constructing models out of sequential data has been studied in a wide range of fields like web process discovery, biological pattern researches etc. Thus, there are numerous other approaches and methods proposed for grammar inference from data. A review of related studies is given earlier in the literature review.

Approach III

In Approach III, we propose a sequence comparison method to detect paths that diverge from target sequences and the points at which divergence starts and ends. During a test session, a subject tries to complete a number of tasks. Each of these tasks consists of smaller subtasks that are linked together and to complete a task the subject moves through these subtasks during the test.

In any real life application, people can not and should not be expected to follow a formal process model exactly. In the overall process, they have freedom of choosing the order of activities and kind of activities they want to follow. This does not necessarily imply an

inefficient or problematic situation. However if we break down the whole process to tasks and tasks to subtasks as shown in figure 3.7, we obtain paths under each task that are expected to be followed in a non-problematic session.

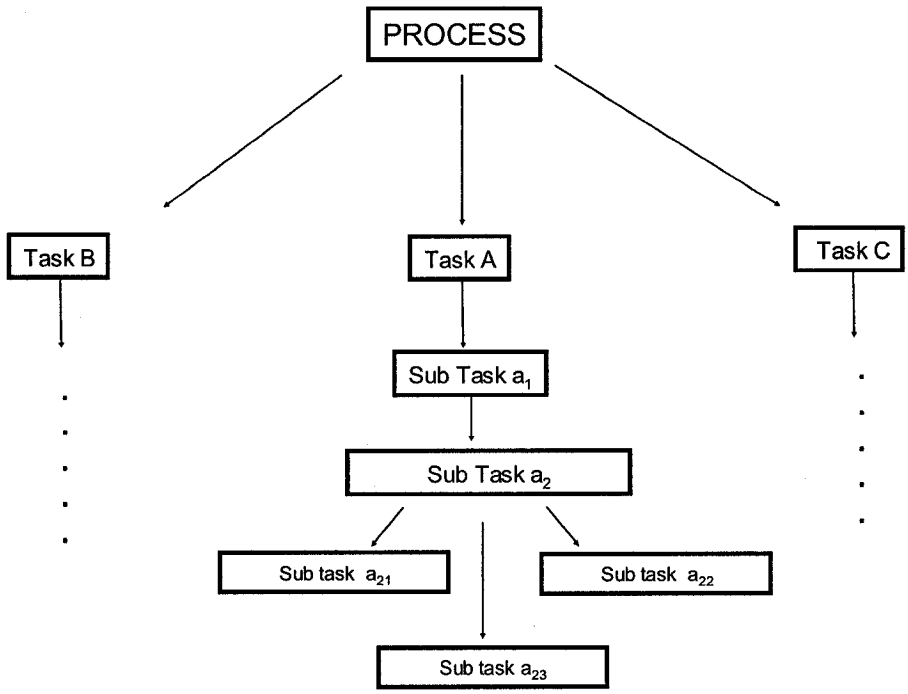

Figure 3.7: Overall process divided into tasks and their subtasks

Then, we can state that, when a subject starts working on task *A*, he/she will try to finish the task and then move to another one. Thus, all the events executed between start and end of event of task *A*, were executed to complete task *A*. Then we can conclude that these tasks are in sub path for task *A*. Finally we can compare these sub paths between tasks with users' sub paths and find the deviations. Since the subject may not be able to finish a task until the end when he/she tries first-time, we capture the end of task *A* from start of another task, say task *B*. This is illustrated in figure 3.8.

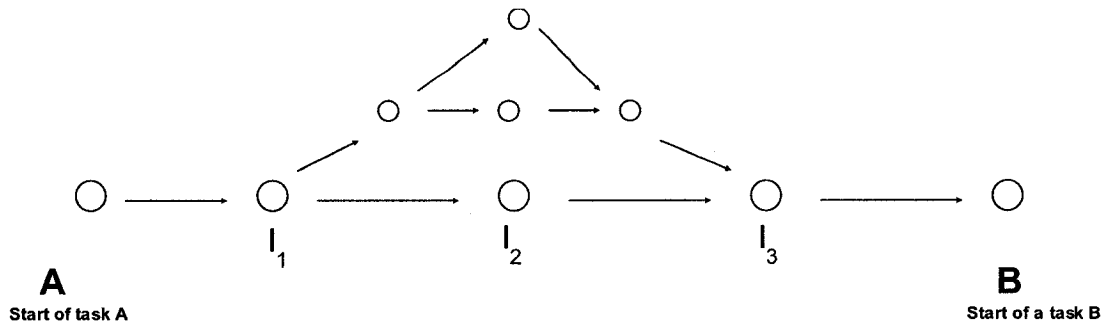

Figure 3.8: Comparison of path between Task A and Task B. I_1, I_2 and I_3 are intermediate tasks for task A.

So the comparison operates as follows:

1. From the process model we extract sub paths that are expected to be followed, in the form of sequence [e_i... ej] where e_i is a starting event and e_j is an ending event for the path. Let *S* be the database containing these sequences.
2. For any sequence [e_i... e_j], compare the sequence [e_i... e_j] with subject's sequence(s) [e_i... e_n[where e_n is the first starting event after e_i.
3. Record the parts of the sequence that do not match until the path turns to normal, record all events that do not match together with the events after which the mismatch starts and with which mismatch ends.
4. For each user repeat this process for all sequences in *S*. Collect all "violating" sequences for subject *i*.

Once we complete comparison for each subject, we have a set of sequences for each subject. Each sequence is an event stream of consisting of "violating" events plus the last "non-violating" event at the beginning and first back-to-normal event at the end.

At the final step, we find frequent deviating paths among the whole set of paths in a cluster as we did in approach I. This is achieved by collecting deviating sequence sets of all subjects of cluster *k*, and then scanning the database for paths that recur at least by a threshold limit.

This method provides us the following information:

1. At what points a certain ratio of users deviate from expected paths, so that we can see the points after which subjects are having problems or ambiguity in the process and can make improvements at this point.
2. What events they execute until they return back to the expected path, so we can understand which activities have been confused most.

As explained above, to apply this approach we need a process model for the system, which describes all the expected paths in terms of events. This can be achieved using task models. A task model describes the set of activities required to reach the users' goals and how these activities are related to each other (Lecerof and Paterno, 1998). Task models are tools that are generally used in the development and the design of computer systems in the area of human computer interaction. We construct and show the use of a task model in our experiment. That process is explained in detail in chapter 4.

3.4.3 Evaluation of Approaches

In approach I as an outcome, we end up with a number of paths whereas in the approach II the outcome is a probabilistic model representing the process with probabilities between states that is actually embedded in the logs. The two techniques are different in terms of visualization and interpretation of results, in the sense that if the previous approach is

adopted, the comparison will be the comparison of numerous paths that emerged for each cluster with each other and with the expected paths; whereas in the second approach, the end result will be a process model for each cluster to be compared with each other and the designer. In systems with large number of events results in approach II may be more difficult to follow and interpret. However it is more informative since it provides the probabilities together with the transition types.

Also it is important to note that in Approach II we assume that system satisfies Markov properties that is, probability of executing an event is independent of all the events before it but the one just before. However this assumption can be changed with the use of higher order Markov models. Also adopting the idea of discovering a grammar for the cluster data, more advanced techniques depending on the features of the system can be developed. In his study, Cook (1996) has reviewed and some of the applied techniques to discover process models for processes that span a long time interval and include different sorts of activities like a project management process in an enterprise.

Approach III provides a comparison method in cases where the objective is to detect the sequences that deviate from optimum or expected paths since its primary objective is to extract and output the differences.

The approach chosen to extract paths at this phase depends on the characteristics of the system/product in question together with the definition of a "path" for the system meaning what information we want to include or exclude in forming the paths. The researchers decide whether being contiguous or being in a certain time frame is critical to form a path in their system, what will be the abstraction level for the events or what kind of comparison is

planned to be used in the end (e. g concrete logs versus abstract models). In our experiment we applied Approach I and III, to extract paths in a cluster. After extracting paths for each cluster, the subset of paths repeated by a certain number were chosen from the whole set. In this path elimination process, a "similarity coefficient for the paths" are used to obtain the paths that are not 100% matching with each other but satisfy a certain proportion of common characters. Figure 3.9 illustrates model architecture for the overall methodology we propose.

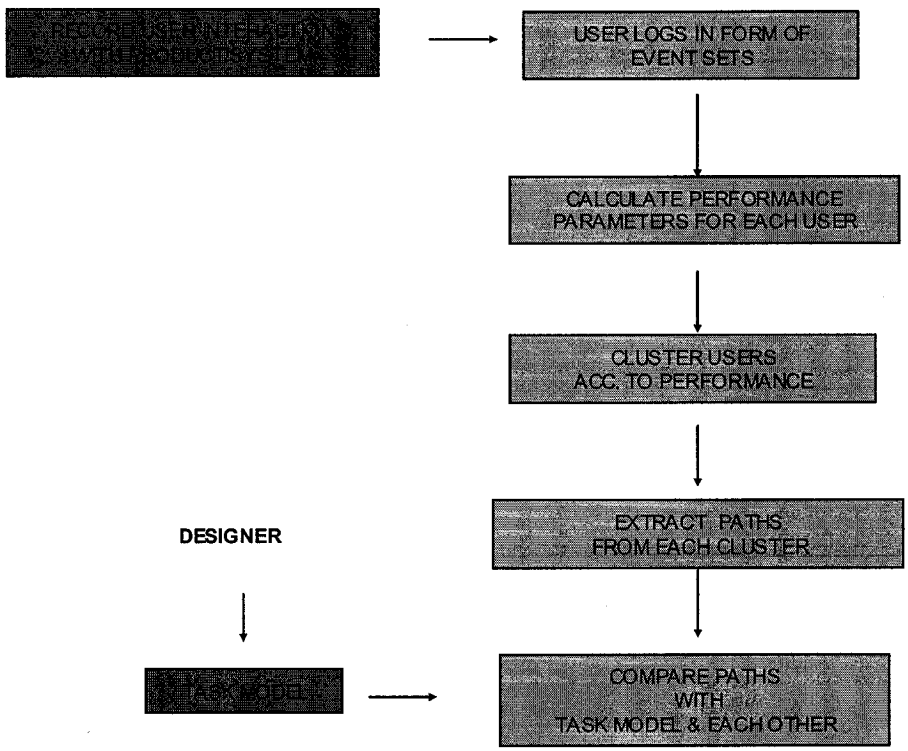

Figure 3.9: The model architecture for the proposed methodology

CHAPTER 4

4 EXPERIMENT

4.1 DESIGN AND OBJECTIVE

To support the proposed methodology in this research, we designed an experimental setup using virtual reality technology. We invited total of 28 subject to conduct the experiment, the subjects were novice people who did not have any specific expertise on the subject. Consent form for using human subjects in the experiments can be seen in Appendix B. Subjects were divided into two groups where groups were asked to accomplish certain tasks in a virtual assembly environment.

The virtual environment used in the test had two versions: in the first version we simulated problematic points as they exist in the real life applications. On the other hand, for the second version we improved the problematic points that were identified as a result of the first test. The first group of participants consisted of 16 subjects and they worked on the first version. The second group of participants had 12 subjects and they worked on the second version. 27 subjects among 28 were students from Concordia University aged 18 to 35. One subject is a university graduate who was running a business currently. Necessary instructions to navigate in the virtual environment, how to use the mouse to pick and move objects etc. were given to each subject before the test session and they were asked to perform once so each subject was comfortable with environment before the test session started. The experiment is carried out in research laboratories in Concordia University. For the experiments two computers (590 MHz Pentium IV, 240 Mb RAM and 2.66 GHz Pentium IV, 1 Gb RAM) were used. Subjects

spent average of 15 minutes in the lab to complete the experiment. We did not provide any financial compensation for the experiments.

Usability evaluation can be performed on systems and/or products with various motives, thus the methodology proposed in this work can be applied for different purposes. Our primary goals in this experiment were to find out:

- Whether our methodology detects the problems of the system that can be detected by direct observation of users
- Whether we could improve the system with insights obtained from the analysis

To answer the first question, we adopted the methodology previously used by Siochi and Hix (1990) in the test of "maximal repeated patterns" study. Their objective in this study was to verify if an anticipated problem in a system is successfully extracted by applying their analysis on experiment results. This constituted the first step of our experiment. In the second step the improved system, where the objective is to eliminate the anticipated problem, is tested to see if problem in the system is resolved. An assembly process was selected as the system to conduct the experiments. The system is an assembly of newly purchased home furniture. It is a simple and realistic process. The problem is faced in the daily life by novice people when they buy new furniture that requires assembly at home. It is a common practice in furniture retail stores to sell the furniture in pieces to facilitate storing and transporting of items. So the customer buys the furniture that comes in pieces with required screws etc and assembles at home after purchase. We found this example well fit in our purpose for the proof of concept in this research work.

The anticipated problem in the selected system is confusion in differentiating between similar pieces in the assembly package, e.g. screws, boards etc. Although it is generally not a very complicated process to finish the furniture, it may be quite annoying depending on the provided details. When we open the package we end up with a bunch of screws and items that look quite similar to each other and a manual that is full of strange names for items. In those cases generally the small pieces like screws, bolts or joints create the biggest problem during the assembly process. In our experiment we first simulate an assembly process that has different kinds of screws, of different sizes. In the first experiment, we asked subjects to complete the task, the assembly process of furniture, in our 3D simulation. During the test we collect user actions and other necessary measurements and store the data to user log files. Next we analyze the data to determine the problematic parts in the system. Once we obtain results of test 1, we analyze whether the methodology detects the problems in the process, which are confirmed to be a problem by the test instructor who observes the tests. Then in the second phase of our experiment, we improve the system according to the problems detected in the first phase. The objective in here is to learn from the user interaction with the initial design and improve the design based on the results. The improved version of the simulation is tested on the different group of subjects. Results are analyzed again using the analysis methodology proposed in this research work. Finally we compare the results of two tests with each other and with the qualitative results that are obtained by direct observation of subjects by the test instructor. Sections 4.2, 4.3 and 4.4 explain the experimental set up, analysis and results in details. Figure 4.1 illustrates the experiment set-up and plan followed in the experiment.

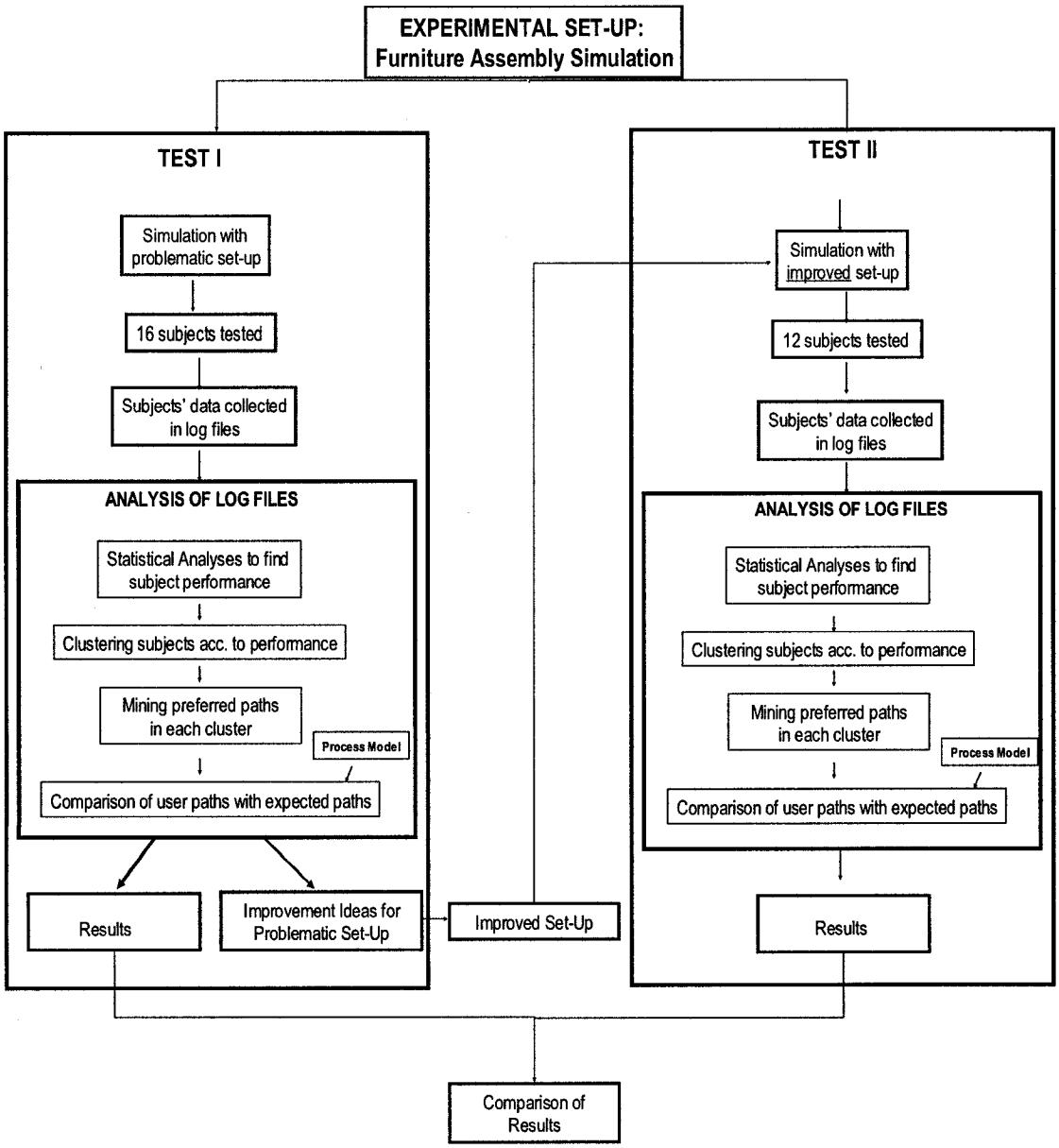

Figure 4.1: Experimental Set-Up

4.2 EXPERIMENTAL SET-UP: THE ASSEMBLY SIMULATION

4.2.1 Data Collection

The virtual prototype used in the experiment was created using the inventor library. As explained earlier, the system used for the experiments is the simulation of a newly purchased home furniture assembly: a wall mounted shelving unit that consists of frames, shelves, and

different types of screws. Selection of a simple scenario eliminated the need to find test subjects with certain skills or expertise. Subjects find the pieces in a virtual room at the center of the environment along with the assembly manual. The manual includes a picture of the completed book shelf, names of the individual pieces and assembly instructions. The instructions are provided using part names and usual wording from assembly manuals of stores. Ex: "Fix the upper frame using one of the 6X screws". Manual for the first test is illustrated in figure 4.2. The assembly environment at the beginning of a test session is shown in figure 4.3.

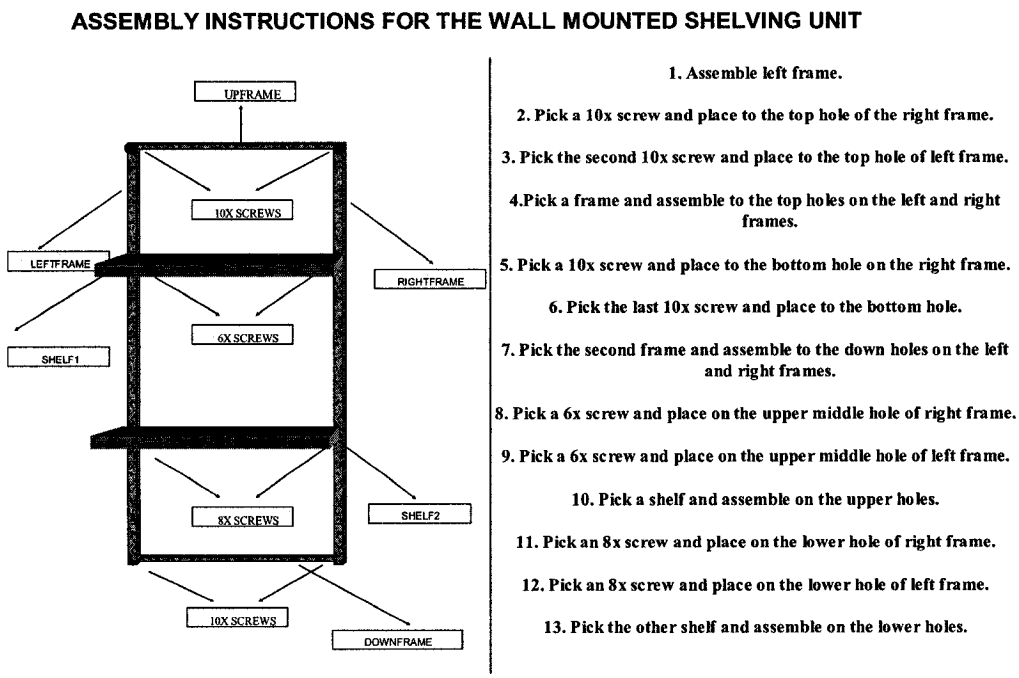

Figure 4.2: Assembly instructions for the shelving unit

The movement in VE is enabled by mouse and keyboard inputs. The virtual environment designed for this simulation enables free movement, scaling and rotation of the pieces. In order to perform the assembly process the subject picks a piece, navigates it freely in the 3D space, positions the piece in the right place and performs the assembly. A message is

displayed when a part is assembled so the user stops trying. In the tested system, for the purpose of simplicity, simulation of actual assembly process (e.g. using screwdrivers, hammer, etc) is omitted. At the beginning of each test session, a training session was given to all subjects on the functioning of the system, how to pick and move the pieces etc so that they could navigate freely in the system when they started to work on the assembly. Figure 4.4 illustrates free navigation of objects in the space and the final state of the assembly.

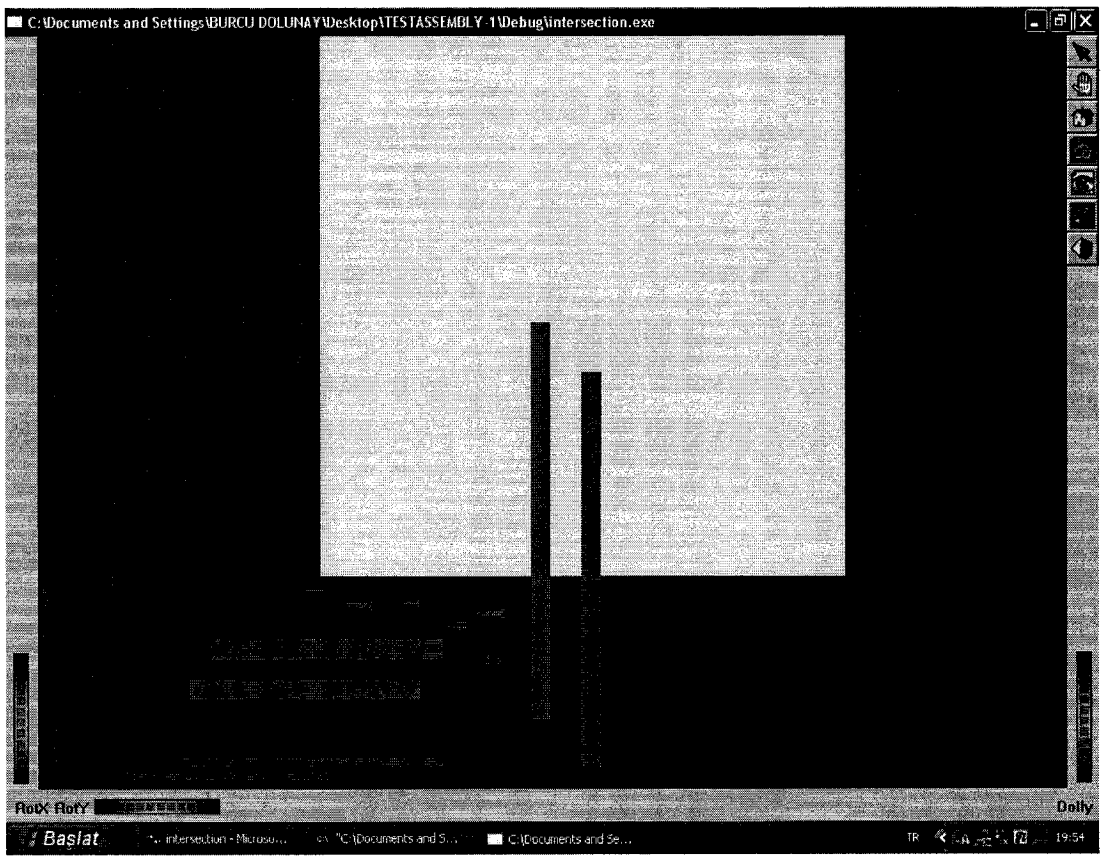

Figure 4.3: Virtual assembly environment at the beginning of the test

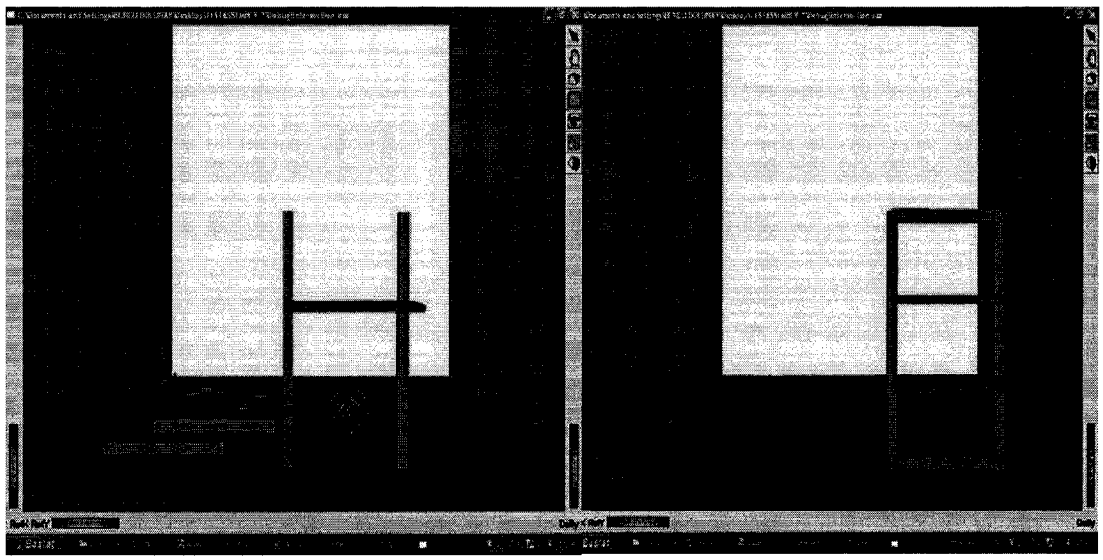

Figure 4.4: Picture on the left: half completed assembly, one of the screws is selected. On the right: Assembly completed

A log file containing user's operations with the virtual assembly environment was recorded automatically. Also previously determined performance measurements about the subject like completion times of individual steps were also measured and recorded in the log files during the experiment.

The log files consist of events $E_i = \{e_1, e_2, \cdots e_n\}$, actually characters that identify events and collected records $A_i : (a_1, a_2 \ldots\ldots\ldots a_m)$ for each subject. The only collected record in out experiment was completion time. For example the log file for User A is $R_j = \langle (i_j), \{e_1 = p_{s^1}, e_2 = m_{s^1}, e_3 = a_{s^1,c^2}\}, (a_1 = T_i) \rangle$ where

R_j = Event set of j_{th} subject

e_i = i_{th} event

p_x = Picking event : pick part x

m_x = Moving event : move part x

$a_{x,y}$ = Assembly event : Assembly of parts x and y

When the subject is working on the assembly, an event is triggered by using keyboard or mouse inputs and is encoded in the log file as an event. In other words the log files do not include user actions in terms of mouse or keyboard entries like. "Click {/ Object}" but instead the task that is executed by this entry, e.g. *Object Picked.* For this kind of data conversion, methods such as log-task table as mentioned in (Lecerof and Paternò, 1998) can be utilized. The data collected from each user was saved in a text file and directly inserted in the program for the first phase of the analysis.

4.2.2 Process Model

We constructed a process model for the assembly process adapting a task model preparation tool from software interface researches. A task model describes the set of activities required to reach the users' goals and how these activities are related to each other (Lecerof and Paterno, 1998). Any process model notation that provides investigators with the level of detail they find enough for their purpose can be used at this step. In our simulation we used the ConcurTaskTrees Notation (CTTN) that was previously used by Lecerof and Paterno (1998) in their software usability evaluation tool. In this notation we can see both the tasks and temporal relationships between the tasks. In CTTN notation there are originally four kinds of tasks which are:

i. User tasks: tasks that users perform alone without interaction with the system. Ex: reading the instruction manual.

ii. Application tasks: tasks that are completed by the system alone this time without user interaction. Ex: storing the time measurements for each task

iii. Interaction tasks: tasks that requires user – system interaction. Ex: picking a piece by clicking on mouse button

iv. Abstract tasks: tasks that are more complicated than a simple action and require completion of several actions to be completed. Ex: completion of overall assembly

Among those event types we use interaction tasks to notate user events with the system and abstract tasks for more abstract events. Abstract events are the events that are not a single interaction of the user with the system like picking object but rather a meaningful collection of simple actions such as completion of frame assembly. Hence the execution of an abstract event generally necessitates the completion of certain actions. Also, in order to form a model that describes the process, we need to know temporal relationships between the tasks in the model. In CTTN task tree notation (Lecerof and Paterno, 1998) temporal relationships are expressed using operators of LOTOS notation. We use two of these operators:

- T1 ||| T2: Interleaving, there is no prerequisite relationship between task 1 and task 2, they can be completed in desired order..
- T1 >> T2: Enabling, completion of task 1 is the prerequisite to start task T2.

These events and relationships between events are collected in a hierarchical tree structure, where higher level events are more abstract events and lower lever events (or leaves) are basic actions. We created the task model for our system using CTTN editor software that is used in (Lecerof and Paterno, 1998) and available on the web as an open source. Figure 4.5 illustrates the CTTN task model for the assembly simulation process on the left and a magnified small section on the right.

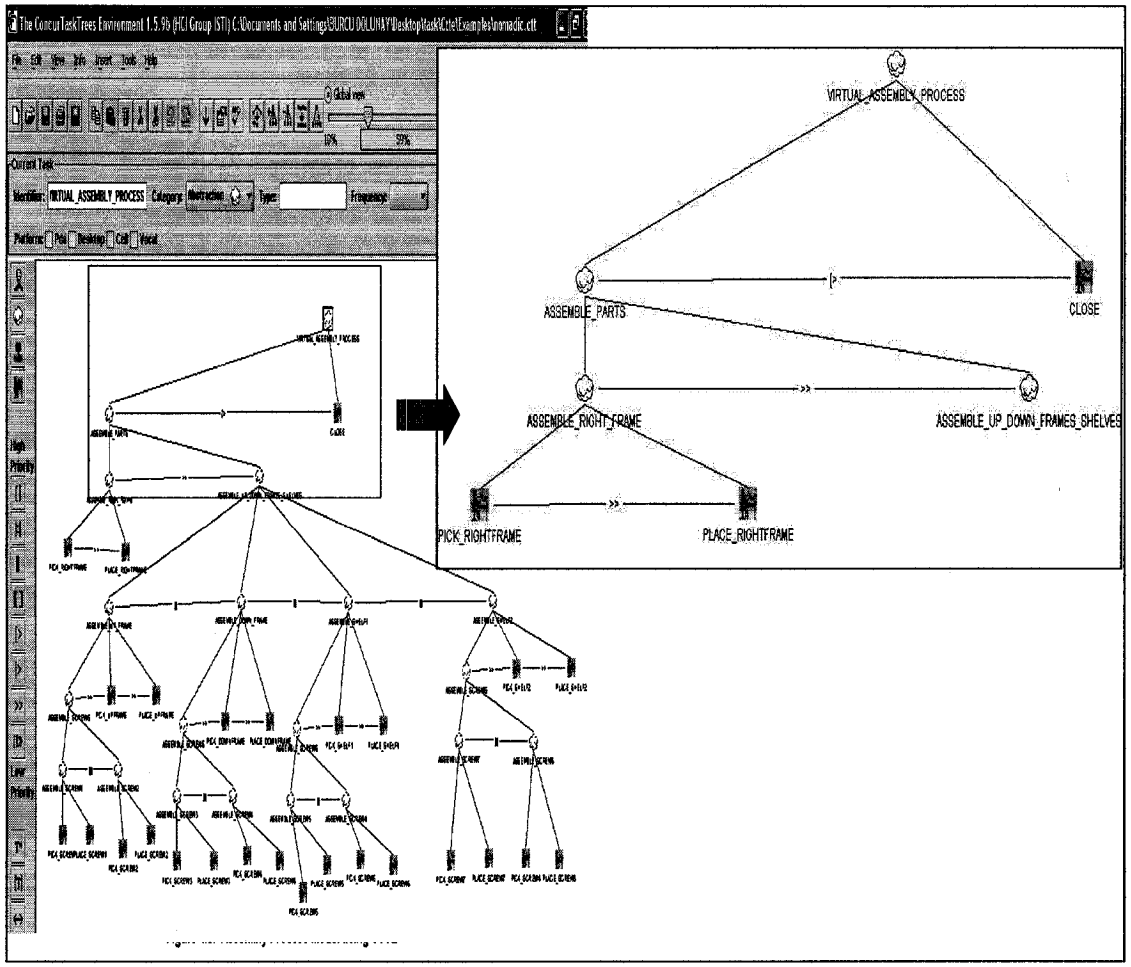

Figure 4.5 : Process Model for Assembly

4.3 APPLICATION OF FOUR-PHASE METHODOLOGY: DATA ANALYSIS

In this step we apply the proposed methodology to analyze the data collected during the experiments. Visual C++ and simple data management techniques are used to develop the test environment. As the model proposes, our test interface consists of four main modules:

- The first module takes user event streams. It calculates and stores previously determined performance parameters for each user. At this step parameters like user session completion time in our case are taken directly from collected records. Event streams collected from users and recorded performance indicators (completion times,

repetitions, cancellations) for experiment I are presented as an example in Appendix A.

- The second module clusters users according to their performance parameters.
- The third module:

  - Extracts paths from the logs of each user and collects the logs of same cluster users in a file.

  - Compares logs of each user with expected sub paths from the process models and collects the deviating paths of same cluster users in a file.

- Finally the last module scans the "path" files of each cluster and chooses frequent paths among them according to the determined threshold frequency limit.

The details of the steps listed above are given in sections 4.3.1, 4.3.2 and 4.3.3.

4.3.1 Performance Indicators: Completion Time, Cancellations and Repetitions

As we explained earlier in the model development, there are certain parameters that are good indicators of performance in any system. As examples to these parameters, some metrics were found to be useful in human computer interaction researches such as delays, ruptures, repetitions, help use patterns , cancel patterns, mouse travel, and mouse clicks per window or performance time, command frequency, command pair frequency, physical device swapping (Hilbert et al., 2000). In our study we determined number of repetitions and cancellations for a task and task completion time as performance parameters. Task completion time for a certain session is an important parameter to understand the ease of use for subjects. Only it must be made clear to the subjects that their task completion time is being measured so that they will not start free exploration in the program. For repetitions, in our case no task needs to be repeated more than once during the assembly so a repetition means that the user

reassembles a previous successfully assembled part. Frequent repetitions can be an indicator of a problematic situation or it can give the idea of putting a shortcut option for that frequently repeated task to the designer. The same idea also applies to the cancellations; cancellations are tasks that have been started but could not be finished and a cancellation is an indicator of a previous bad choice or confusion for the user.

So for our experiment, we calculated those three parameters for each user:

1. Number of repeated subtasks for subject *i* (S_i): K_i
2. Number of cancelled subtasks for subject *i* (S_i): C_i
3. Task completion time for subject *i* (S_i); T_i

Among those number of repeated and cancelled tasks are calculated from the event streams ($E_i : \{e_1, e_2, \cdots e_n\}$) and task completion time was collected at the end of session in the logs ($A_i : (a_1 = T_i)$). Table 4.1 illustrates the results for the first three subjects of test 1, as an example of the outcome obtained from this phase.

Table 4.1: Example from the list for three indicators calculated in this phase for each user

SUBJECTS	INDICATORS		
	K_i	C_i	T_i
S_1	10.125	8	11
S_2	8.128	9	23
S_3	13.43667	8	12

4.3.2 Clustering

As we explained in the model in chapter 3, the aim of clustering is partitioning the set of subjects into classes, such that members of each class shares similar performance results in terms of determined performance indicators. As the outcome of phase 2 we obtained an *mx3* matrix at the end of each session where *m* is the number of subjects (respectively, 16 and 12 for test 1 and test2) and 3 is the number of indicators. The indicators are task completion time, number of cancelled tasks and number of repeated tasks. Then we followed following steps:

1. We normalized the data using the following standardization method (Kaufman et al., 1990):

   $x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j^A}$ where:

   x_i : i^{th} subject
   x_{ij} : value of the j^{th} indicator of the i^{th} subject
   $x_{ij'}$: standardized value of indicator

   σ_j^A is the absolute deviation of indicator j and μ_j is the mean for indicator j and

   for any indicator j; $\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$ and $\sigma_j^A = \frac{1}{m}\sum_{i=1}^{m} |x_{ij} - \mu_j|$

   We preferred this standardization technique over other techniques in the literature since it is the most robust method to the distribution of data and outliers. (Kaufman et al., 1990)

2. We plot the data and made a correlation analysis on the data to see if there is any indicator that shows very little variation or that is highly correlated to one of the other indicators. This analysis enables us to discard dimensions that are unnecessary as a feature in clustering. Such an analysis is also called feature selection in clustering literature. So we performed a correlation analysis where we obtained following values for the correlation coefficient (R):

- between time and number of cancellation data sets R= 0.21;
- between time and repetitions is R=0.26;
- between repetitions and cancellations is R=0.1275.

These results are not significant to imply a correlation between any two data sets. Also figure 4.6 shows pair wise plots for indicators, we can say the points are randomly scattered and no pattern is observed.

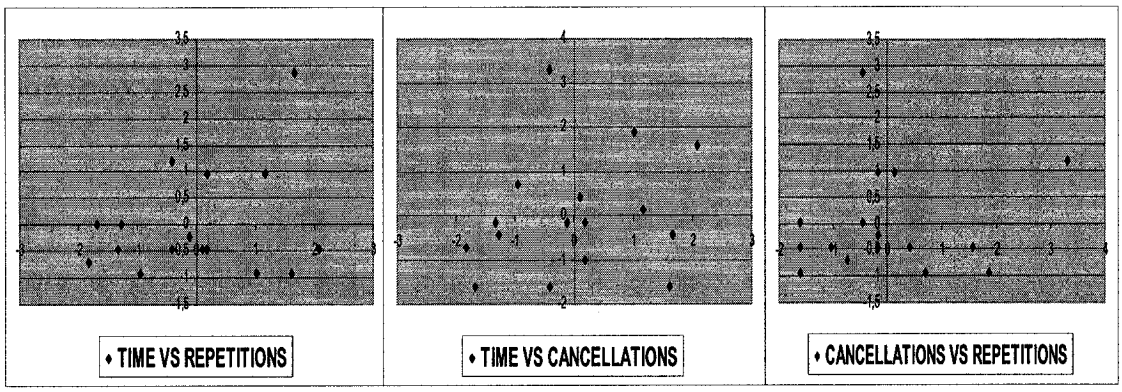

Figure 4.6: XY scatter plot for three indicators

3. Since all our performance indicators are numerical values, we use Euclidean distance between points as our similarity measure.
4. Finally we use k-means clustering algorithm to partition the set of n subjects into k clusters so that similarity between subjects in a cluster is high but similarity between different clusters is low. The algorithm to find the clusters proceeds as follows:
   i. Selects k initial cluster's centers
   ii. Computes similarity as Euclidean distance between a subject and each cluster center
   iii. Assigns a label to a subject based on the minimum similarity

iv. Repeats for all subjects

v. Re-computes the centers for clusters which is a mean of all subjects assigned to a given cluster

vi. Repeats from Step ii until objects do not change their labels.

The reason we decided to use k-means algorithm over the alternative data mining methods is because it is effective and simple to implement in practice. The most important disadvantages of k-means algorithm are its sensitivity to noise and outliers and the constraint that every point has to be in one cluster. In our data set we had 16 and 12 subjects respectively; from pre-evaluation of data before clustering we could detect the outliers in each category so we did not put them in the algorithm but assigned a cluster manually. So that the data set did not include outliers and noise, which are weaknesses of k-means algorithm. Furthermore we also want each subject to belong only one cluster which is also a weakness in the algorithm. We cluster the normalized data using the k-means algorithm and obtain labels of clusters for each user.

4.3.3 Path Extraction

For the path mining phase, our objective is to find out:

- whether subjects follow the expected sub paths
- are there any particular paths that are preferred by subjects among the paths that have alternatives
- are there any particular differences in terms of paths between clusters

We apply approach I and approach III as described in section 3.4.2 to reach our objective in this section. In approach I, we mine traversal paths for each user's sequence and then find the frequent sequences in each cluster. To mine the paths, we evaluate features and needs of our test case. In our assembly example, no events should be repeated. Moreover, subjects are expected to follow a certain path once they initiate by selecting the initial task in the path.

So for our purposes, what we need is to eliminate the effects of repetitions, cancellations and focus on the series of events that subjects prefer in completing. Since we include repetitions as one of the performance measures, we collect statistics about repetitions. As a result we apply "maximal forward sequences" algorithm from Chen et al. (1998). For the second part we devise a scanning algorithm to eliminate paths that are used less than a certain number of times. Finally, we also apply Approach III by implementing the "sequence comparison method" that was introduced in chapter 3. Results of these approaches for each test are presented in section 4.4.

4.4 RESULTS

The analysis approach we explained here can be used with different motives such as understanding of problems with a product to improve it, see points where users are failing: to provide better training at these points or to see different experiences by different groups of users etc. In the test study we asked subjects to assemble the parts using our virtual assembly, recorded their session and applied automated data analysis. It must be noted that the experiment conducted here was not a rigorous formal experiment but a small-sized case study where we aimed to apply automated data collection and analysis to detect problems about the assembly environment that would otherwise be detected by direct observation. In each test we

collected both quantitative statistical results and qualitative information that are the paths followed or violated. Statistical results include following:

- Task completion time for each subject
- Number of repeated tasks for each subject
- Number of cancelled tasks for each subject
- Number of repetitions/per task in the overall test results
- Number of cancellations/per task in the overall test results

4.4.1 Test 1

Statistical Records

Total 16 subjects were tested in the first scenario. Among the subjects, we found the mean task completion time is 9.522 minutes, the mean number of repetitions is 3.9 and the mean number of cancellations is 4.67. Standardized values for these indicators for each subject in the experiment 1 are plotted as shown in figure 4.7.

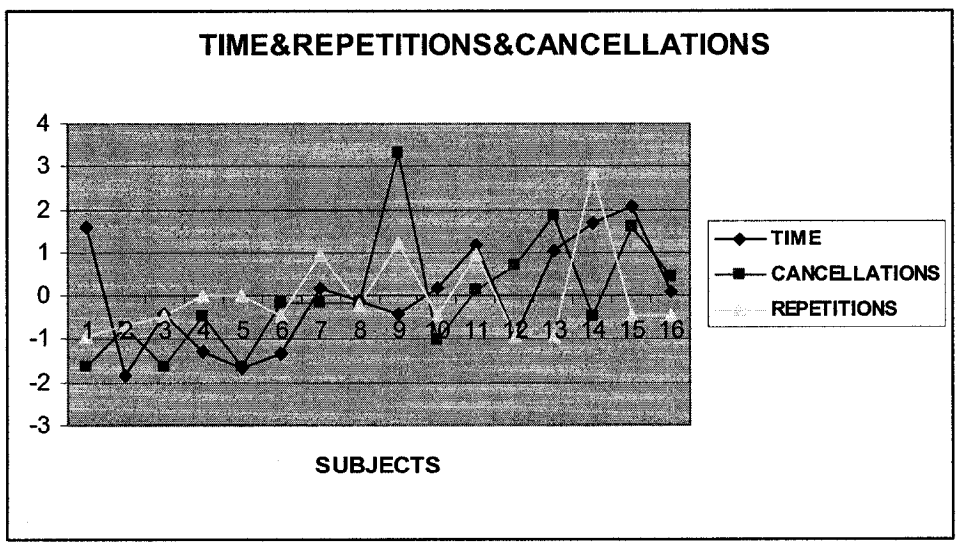

Figure 4.7: Standardized task completion time, number of repetitions and number of cancellations for 16 subjects in experiment 1

We also calculated number of repetitions and cancellations for each task. The most cancelled tasks were assembly of screw3, screw4 and screw5 which were cancelled in 1.375, 1.25 and 1.25 times/per user respectively. The least cancelled tasks were the assembly of shelf1, shelf2 and right frame which were cancelled 0.0625, 0.1875 and 0.5625 times/per user respectively. In case of repetitions, the most repeated task was assembly of right frame, assembly of shelf2 and shelf1 with 0.9, 0.8 and 0.75 times/per user respectively. Number of repetitions and cancellations per user for tasks is shown in figure 4.8 where DF&UF means down frame and up frame, SH1 and SH2 means shelf1 and shelf2, RF means right frame, S{1, ... 8} are the screws{1, ... 8}.

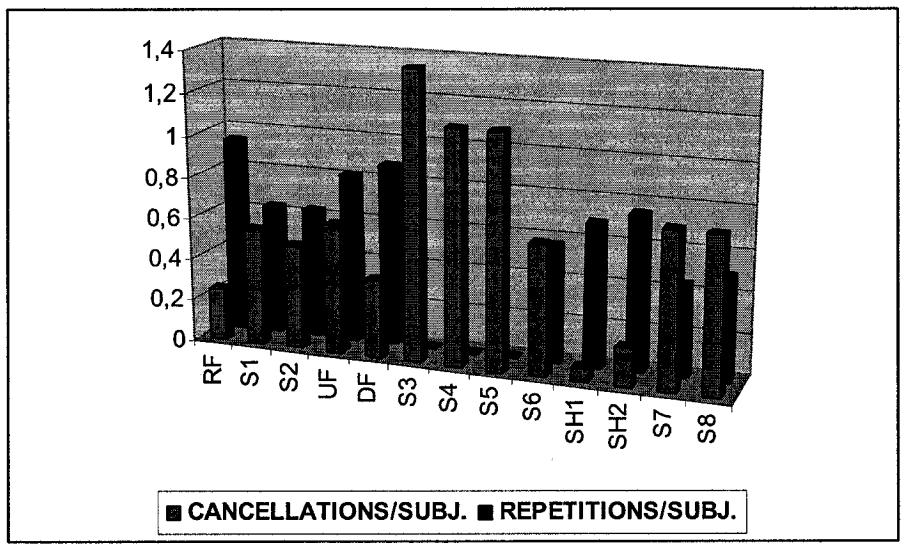

Figure 4.8: Number of repetitions and cancellations per piece user for test 1

Path Analysis

As a result of approach I we obtained paths traversed by subjects and their appearance frequency in the cluster. Below figure 4.9, figure 4.10 and figure 4.11 illustrate the paths whose frequencies are at least %15 in cluster one, two and three respectively, with their percentage in the overall set. In terms of performance indicators, cluster1 includes the subjects

that have the best overall performance whereas the cluster3 includes the subjects that demonstrated the worst overall performance.

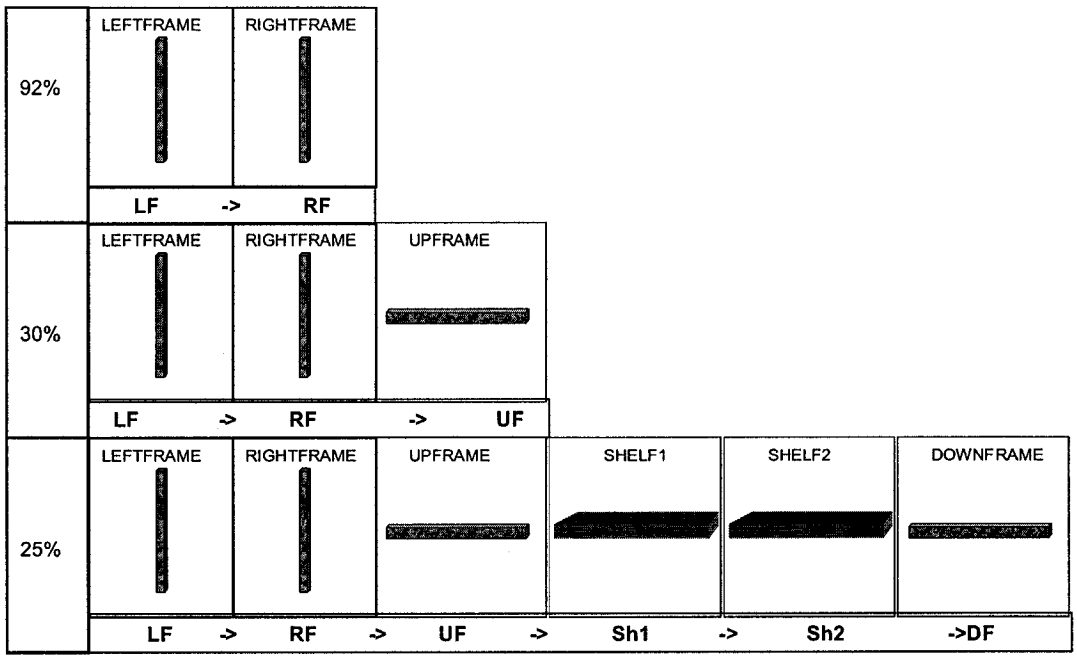

Figure 4.9: Paths from cluster 1

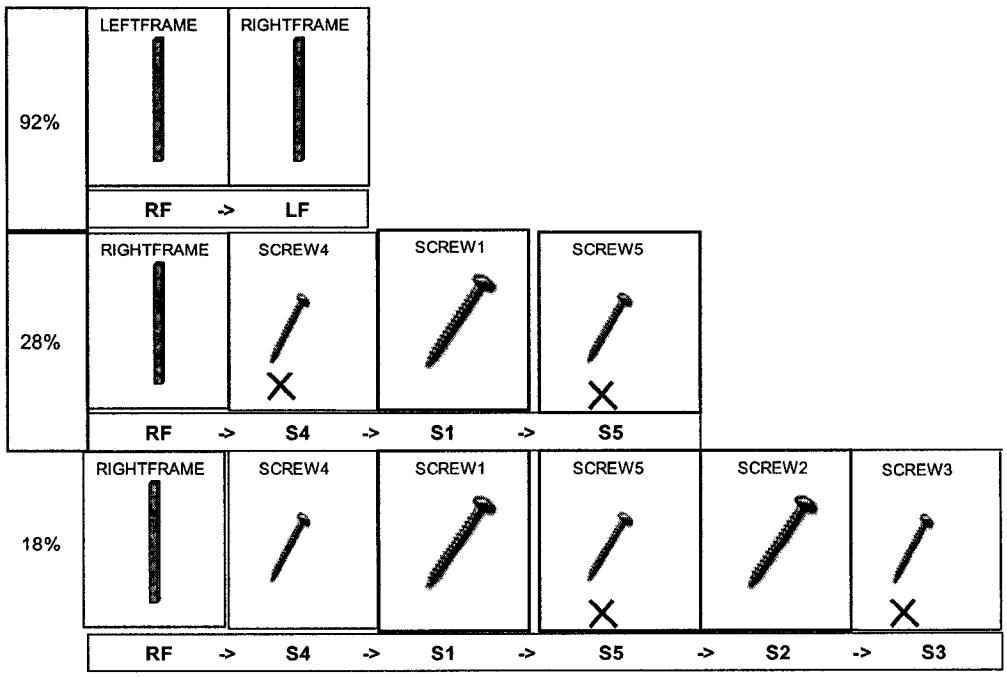

Figure 4.10: Paths from cluster 2

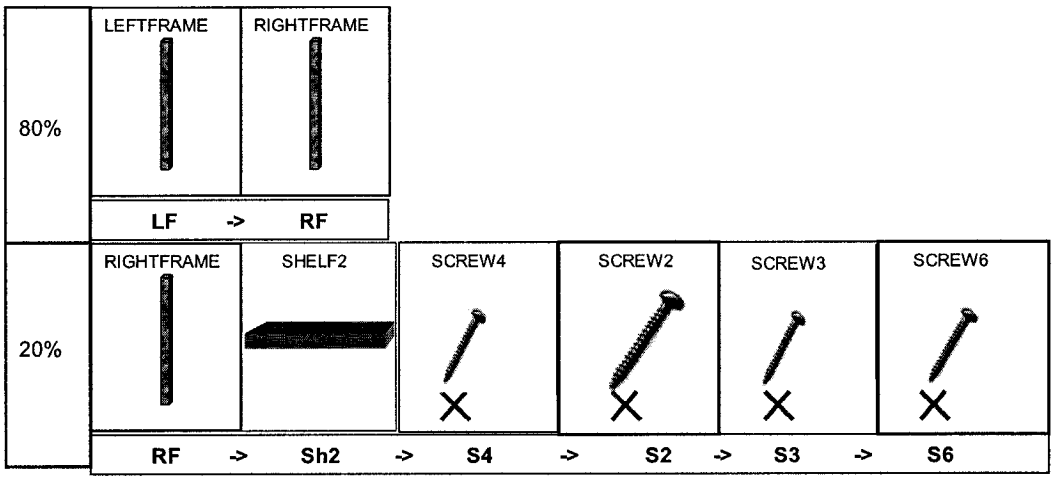

Figure 4.11: Paths from cluster 3

As result of approach III we found violating sub paths that repeated in each cluster. Number of violating paths is 57 in the experiment1, which means 4.75 tasks per user. While the shortest observed pattern consists of one event, the longest observed pattern consists of 11 events. Mean path length is 3.14. However none of the long patterns were frequent. It is observed that frequently violating sub paths consist of only one or two events as illustrated in figure 4.12.

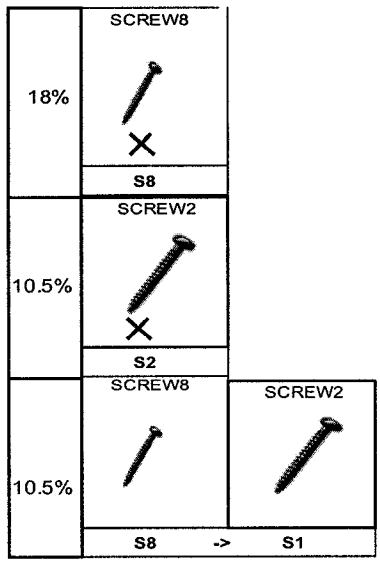

Figure 4.12: Violating sub paths: X indicates the failure of a task.

The second outcome of approach III includes two parts:

- V-Start: the set of tasks after which a violation of expected paths started
- V-End: the set of tasks with which the path turned to normal

In the set of last non-violating events after which an illegal path started (V-Start), the event "assembly of right frame" was the most frequent one such that %24.56 of all the violations started after this event. Furthermore, V-Start is caused %12.5 of the times after the "assembly of down frame". In the set of first back-to-normal path events (V-End), our results indicate that 14% of the violating paths turn back to the normal as a result of either "picking screw1" and "picking screw2" events.

Evaluation of Test 1 Results

Our analysis on the log files of 16 subjects revealed that the number of cancellations are noticeably higher for the events $a_{s^i,x}$, which are assembly of the i^{th} screw with part x. Cancellations per piece with respect to piece type can be seen in figure 4.13 (where DF and UF are down frame and up frame, SH1 and SH2 are shelf1 and shelf2 and RF is right frame). So analysis of canceled paths points to the problem of screws.

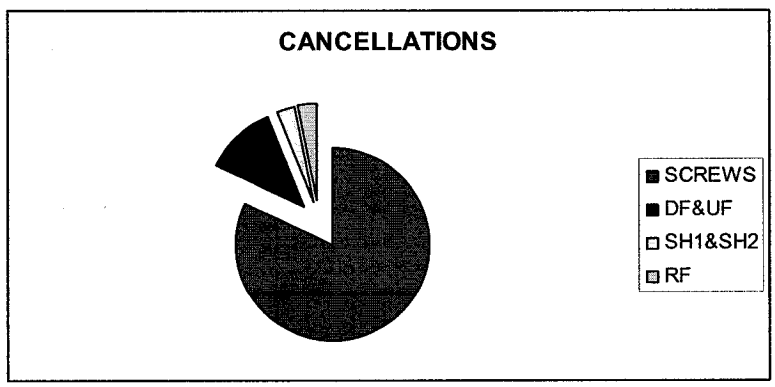

Figure 4.13: Pie chart for cancellations per piece for 4 different types of pieces in the assembly

However, we can not make the same conclusion for the results of task repetition analysis. Results revealed that most repeating tasks were assembly of frames or shelves. Nevertheless, we can explain these results from the qualitative data we collected during the test by observing subjects. Actually a repetition in our system means removing a previously successfully assembled part and reassembling it. Repetition process does not require a significant effort in our simulation since the simulation does not deal with physical tasks such as the use of hammers, attaching pieces together using force etc that would have been essential in a real life assembly process. Repeating a process simply requires placing a piece to the right place in a 3D space. Therefore every time users complete the assembly of large pieces, they play with other already assembled parts to obtain somewhat a *better adjustment.* This resulted in high repetition numbers for the large pieces which is in fact unlikely in real life. We learned two lessons from this experience:

i. The necessity and importance of path extraction phase: Although statistics tell a lot about a system, they may be misleading due to several factors For example without using the paths generated as result of our experiments statistics alone would not be sufficient to explain the unnecessary repetitions on large items. However, observed paths clearly demonstrate that repetitions on the large items were not due to user violations but due to the limitations of our virtual environment. Furthermore the statistical analysis could not interpret intentions of users once the experiments are completed. Most of the times we have to go through traditional observation stage to come up with correct comments about summary statistics. However in our methodology we are achieving this insight from the paths that are extracted from user logs.

ii. The importance of quality of virtual prototype: The quality of the virtual prototype is a crucial factor in the quality of the test results. In other words the weaker the model in imitating the real product or system, there is more possibility for the results to be affected from the differences between the prototype and real world. However employing several methods as proposed in this research work, improves the reliability of results.

In our experiment results from path extraction phase supported the expected outcome, since several frequent paths that emerged in the clusters had the structure in figure 4.14, where a user successfully assembles one of the larger pieces but makes some unsuccessful trials to find the right screw for the assembly of the next part.

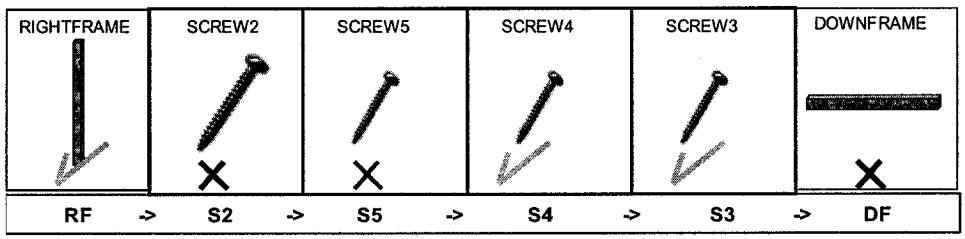

Figure 4.14 : An example path showing trial of two wrong screws to complete assembly of down frame

Analysis using approach III also supported these findings since the most frequent violating patterns were the paths that consisted of one event, which is generally an unsuccessful trial of a screw.

From the analysis, we can conclude that the algorithm correctly identified the problem with screws. For the second experiment the following improvement is implemented: better differentiation for screws and more instruction.

4.4.2 Test 2

The second round of experiment is run with the scenario where different types of screws have different colors and the instruction sheet is prepared using this difference. In the instructions, more emphasis is put on the places of the screws using color differentiation. Using different colors is a practical solution which is also realistic and can be achieved by putting different color stickers on screws in real life. Figure 4.15 is the assembly at the beginning with different coloured screws. Total 12 subjects were tested in the second scenario.

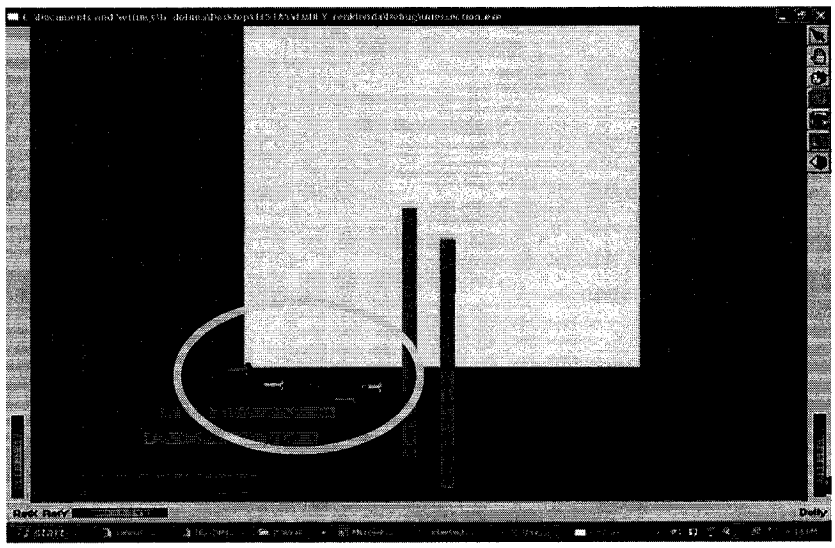

Figure 4.15 : Assembly simulation with different coloured screws

Statistical Records

Among the 12 subjects in the second test, we found mean task completion time is 2.397 minutes, mean number of repetitions is 2.25 and mean number of cancellations is 0.667. Standardized values for these indicators are plotted in figure 4.16.

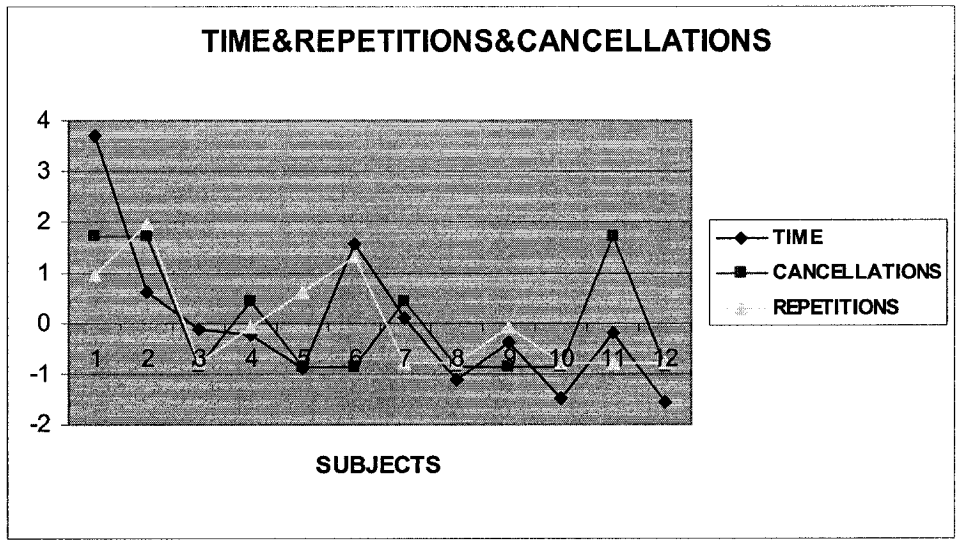

Figure 4.16: Standardized task completion time, number of repetitions and number of cancellations for 12 subjects in test 2

We also calculated number of repetitions and cancellations for each task. The number of cancellations decreased significantly from test 1 to test 2 .The most cancelled tasks were, assembly of screw 1, screw 5 and screw 6 which were cancelled 0.1667 times per user each.

In case of repetitions, the most repeated task was again assembly of right frame, assembly of shelf 2 and down frame with 1.3, 1.1 and 1.08 times per user respectively. Number of repetitions and cancellations per user for tasks is shown in figure 4.17 where DF and UF means down frame and up frame, SH1 and SH2 means shelf1 and shelf2, RF means right frame, $S_1$is S{1, ... 8} are the screws{1, ... 8}screw 1, S2 is screw2, S3 is screw 3, S4 is screw 4, S5 is screw 5, S6 is screw 6, S7 is screw 7 and S8 is screw 8.

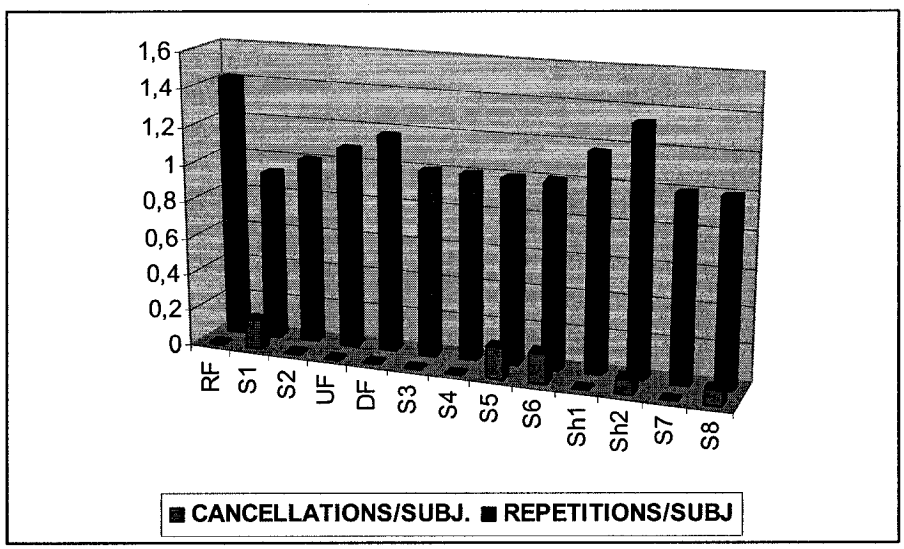

Figure 4.17: Number of repetitions and cancellations per piece user for test 2

Path Analysis

We applied the same path analysis method with test 1 to extract paths in test 2. The paths that emerged in test 2 in clusters 1 and 3 and their frequency percentages are given below in figure 4.18 and figure 4.19. In terms of performance indicators, cluster 1 includes the subjects that have the best overall performance whereas the cluster 3 includes the subjects that demonstrated the worst overall performance.

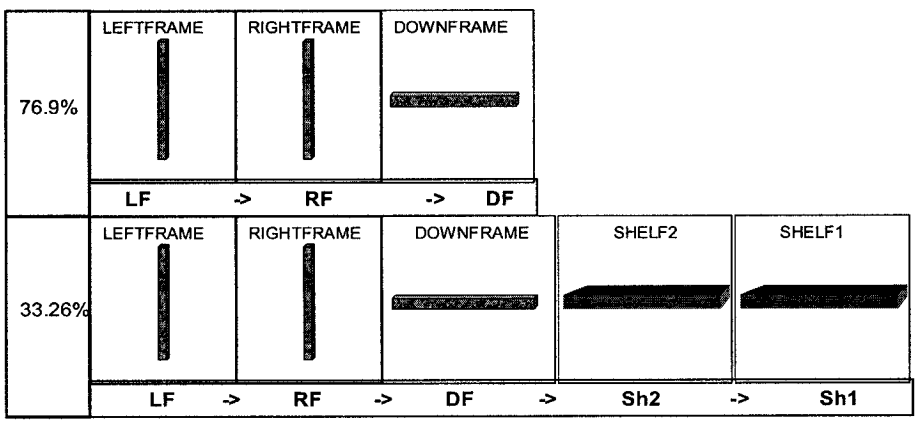

Figure 4.18: Paths from cluster 1

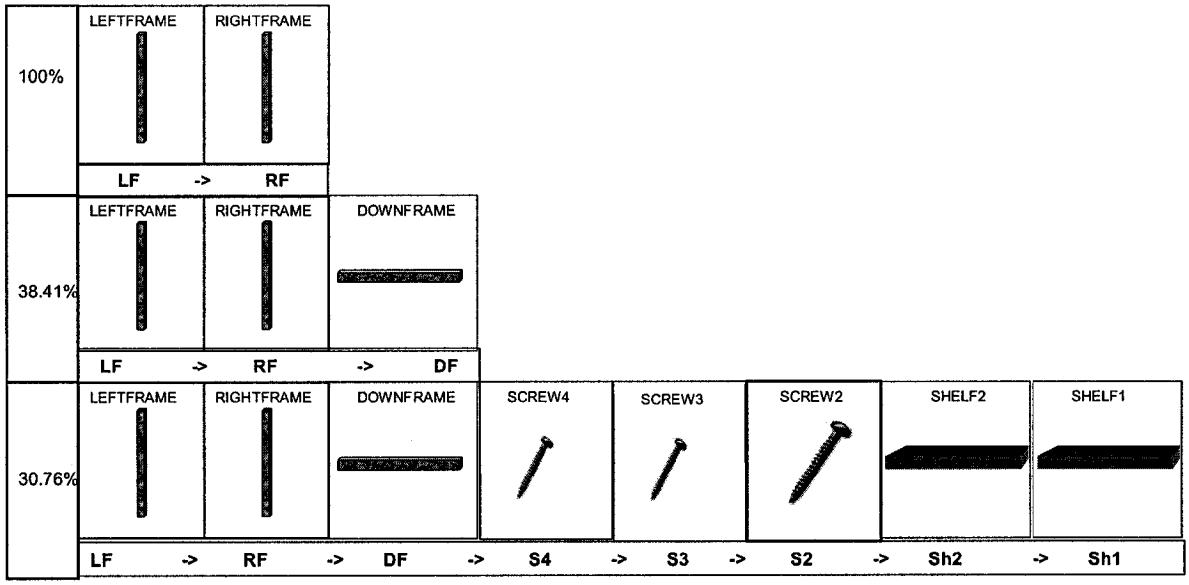

Figure 4.19 : Paths from cluster 3

We then found violating paths in test 2 by applying approach 3. In test 2 the number of violating paths was only 7. Only one of the paths consisted of 2 events and the remaining were all one-event violations, the mean path length was 1.14. One event showed up more than once in violating events, which is unsuccessful trial of assembly of screw 1.

In the second outcome of approach III (V-Start and V-End sets), in V-Start (the set of tasks after which a violation of expected paths started) there were not any tasks that appeared more than once. In V-End (the set of tasks after which the path returned to normal) "assembly of shelf1" and "assembly of screw4" were the only repeating tasks and they appeared only twice.

4.4.3 Comparison of Test I and Test II Results

To evaluate the results of test 1 and test 2 and to see if there is any difference in the results after the changes made in the assembly we performed following analyses:

1. Comparison of mean task completion time, mean number of repetitions and cancellations in both tests: To compare means we applied hypothesis testing on the means of two samples, with different sizes and unknown variances. Hypotheses and test parameters are summarized in Table 4.2.

Table 4.2: Hypothesis test concerning the difference between two means

H_0	Value of Statistic	H_1	Critical region
$\mu_1 - \mu_2 = d_0$	$t' = \frac{(\overline{x_1} - \overline{x_2}) - d_0}{\sqrt{(s_1^2/n_1) + (s_2^2/n_2)}}$ $v = \frac{(s_1^2/n_1 + (s_2^2/n_2))^2}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}}$ $\sigma_1 \neq \sigma_1 and\ unknown$	$\mu_1 - \mu_2 > d_0$	$t' > t_\alpha$

2. We compared the length and number of deviating paths in two tests.
3. We analyzed frequent traversal paths we obtained from two tests.

As a result of these analyses, the improvements can be summarized as below:

1. Mean assembly completion time decreased by 28.57% in the second set of subjects ($p<0.05$).
2. Mean number of cancellations decreased by 43.2% in the second experiment. ($p<0.025$). There was not a significant decrease in the number of repetitions. This is reasonable because we have concluded in the evaluation of experiment 1 that repetitions are not an indicator of problem with the screws. Details of the statistical results in terms of performance indicators are summarized in Table 4.3.

Table 4.3 : Statistical tests for comparison of two systems in terms of performance parameters

	Hypotheses	**p-value**	**t-value**	**Critical Region**
Completion times	$\mu_1 - \mu_2 > 0$	p<0.005	2.959282	$>t_{0.995}$
	$\mu_1 - \mu_2 > 5$	p<0.02	2.267037	$>t_{0.995}$
	$\mu_1 - \mu_2 > 8$	p<0.05	1.851691	$>t_{0.995}$
Repetitions	$\mu_1 - \mu_2 > 0$	p<0.15	1.3187	$>t_{0.85}$
Cancellations	$\mu_1 - \mu_2 > 0$	p<0.0025	3.754647	$>t_{0.9975}$
	$\mu_1 - \mu_2 > 5$	p<0.025	2.031673	$>t_{0.975}$
	$\mu_1 - \mu_2 > 6$	p<0.1	1.687078	$>t_{0.9}$

3. Besides these statistical analyses we also observe changing patterns in the pattern analyses. As illustrated in figure 4.20, the common frequent pattern that was observed in test 1 did not emerge in test 2. Frequent paths in test 2 generally included successful consecutive assemblies of parts. Also when we compare results of deviating paths analysis, we see the total number of deviating paths was 57 in the first test which means 3.5625 deviations per subject. This number decreased to 7 paths in the second test which means 0.58 deviations per subject. Also the mean length of deviation was 3.14 in the first test versus 1.14 in the second test. This implies that the users followed expected sub paths in the second test much better than the first test.

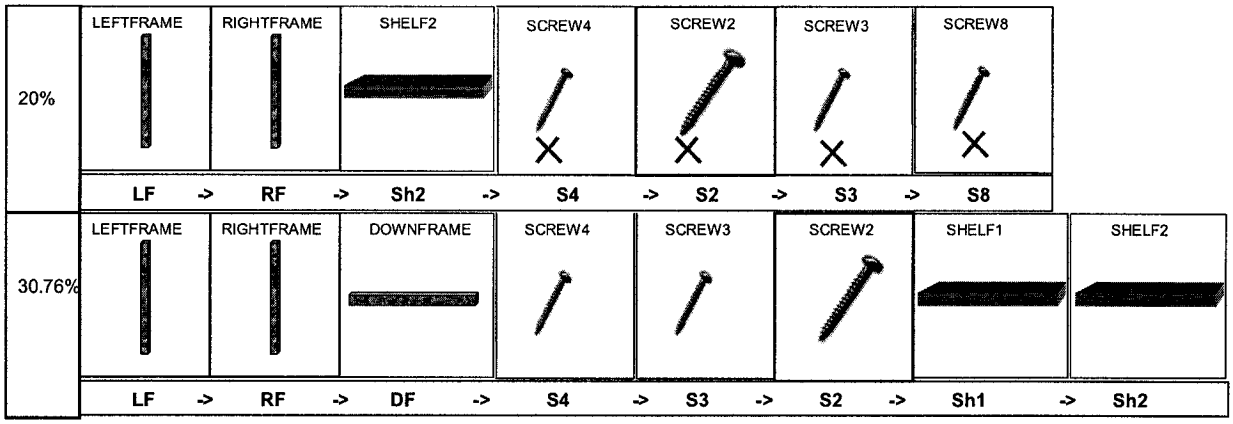

Figure 4.20 : Clusters with the lowest performances: The first path is taken from cluster 3 of test 1 and the second path is taken from cluster 3 of the test 2

Finally, another set of paths that emerged in both groups was attempting to find right places for large pieces before the screws as indicated in figure 4.21. Since before the experiment we knew that the screws were the confusing points in the experiment, we were expecting to find the paths about the screws like illustrated in figure 4.20. However the tendency to figure out the large pieces first and then search for right screws was not an anticipated structure. This means the analysis revealed a behavioural pattern that was unknown to experimenters before the experiment. This finding was also supported by direct observation of test subjects during the experiment.

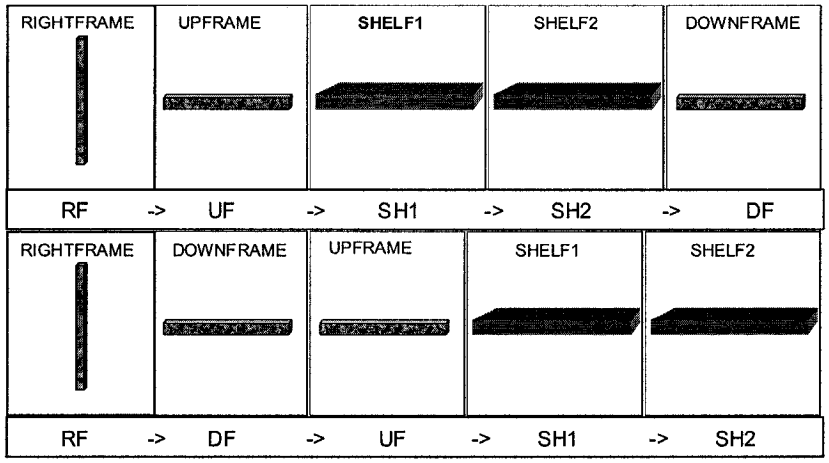

Figure 4.21: Path that emerged in both groups. Subjects first try to work out large pieces before screws.

4.5 DISCUSSION AND SUMMARY

To apply and test the analysis approach proposed in this research work, we constructed an experiment study, an assembly process for simple home furniture. As virtual prototype, we created a 3D simulation of the assembly process. Although our virtual prototype did not handle all the aspects of a physical assembly process like the labour required to work with hammers, physical forces etc., it could handle the problem we intended to simulate in the system, namely the confusion about the choice and assembly of small pieces like screws.

Our results show that the statistical analysis and paths extracted indicate the major problem of the system when interpreted together. Statistical results alone may not be enough to detect problems and can lead to erroneous results as observed in the repetitions case in our experiment. Thus a behaviour analysis approach like the pattern mining approach proposed n our methodology is essential to better understand user experience. Also in our study, since our sample size was not very large and the process was not very complicated, we did not observe distinct differences in terms of patterns followed among clusters. However we observed an

increase in the frequency of problematic paths in lowest performance cluster compared to highest performance cluster. This encourages us to think that with the tests of larger sample sizes the cluster information will provide much valuable insight. As a result, we can conclude that the methodology proposed here helps to find problems in systems without the time consuming necessity of observing users one by one.

CHAPTER 5

5 CONCLUSION

Today significant amount of effort and money is invested in designing and marketing of products or systems that serve to consumers' needs and expectations at the highest level. Understanding these needs and expectations is only possible by detailed analysis and interpretation of actual users' interaction with the product and systems. Today this is achieved through traditional usability tests that are time and resource consuming. In this thesis we have introduced a VR based automated end-user evaluation methodology, which is an automated usability testing, to provide automated support for decision makers. In broad terms, automation of usability testing requires two basic efforts, automated capture of user behaviours and automated analysis of this usage data. In this context, virtual reality technology is essential as a first step since it provides naturalistic real-time simulation environment where human and systems can exist together. Simulations together with appropriate equipment are used to capture behavioural data in physical systems. However the challenge is in automating the analysis of voluminous data without losing the hidden valuable information. In this work, we integrate results from different fields, and design a VR based user evaluation toolkit that enables automated analysis of behavioural data for all kinds of systems. We propose a four-phase data-analysis method:

i. using statistical methods to determine and calculate parameters that are good indicators of performance in system;
ii. clustering users according to the similarity of their experiences with the system in terms of these performance parameters;
iii. extracting paths that have been followed frequently in each cluster;
iv. comparing these paths with expected process model and with each other.

Comparison of actual user data and expected process models has been mostly carried out in three ways in literature:

i. by counting the number and kind of non-matching characters in the actual and target file and calculating numerical match percentages;
ii. by comparing the files with an abstract model and reporting behaviour that deviates from rules in the model;
iii. by aligning user actions with the target logs and providing visual representation of path comparison

However, in the usability, assembly and disassembly for maintenance tests/training, the interest may not be only the error identification but also the discovery of the paths to find out users' way to execute various tasks. Although applications with visual alignments are designed to serve this need, adapting them directly for user evaluation for product design is not convenient. Because log files generated as a result of automated data collection systems include significant noises. If we take the logs without filtering, excessive expert effort is required to extract meaningful information from comparison. In order to find meaningful behaviour patterns in voluminous data, noise in the log files should be cleared. In other words, "interesting" patterns must be mined from logs in an automated way. For this purpose, we have implemented and devised methods that can handle mining traversal patterns in sequential data. Once patterns are found we can compare the patterns with each other and process model. So the automated data collection and analysis toolkit proposed in this research provides a smart classification and filtering method before comparison. At the end of analysis, the results give insight about the users' way of dealing with certain tasks in the system or product. Furthermore we get insight about the relative performance of users of a path with respect to users of another path. In other words we not only see the followed paths but we

learn what paths are followed by low-performance users and what paths are followed by high-performance users as well.

To validate the effectiveness of the proposed method, we conducted an experimental study on 28 subjects at an assembly simulation. Results clearly demonstrate that the proposed methodology is capable of capturing user behaviour from the virtual interaction with a system.

The approach proposed in this work aims to fulfill the need of understanding the end-user without massive human guidance, in other words to automate the end-user product tests without losing information that an observer interprets. We do not claim that such an automated device can fully replace a human observer especially in the process of interpreting user intentions; however it can support and supplement evaluation process by increasing the effectiveness and applicability of automated end-user evaluation.

To conclude, the main contributions of this work are extending existing automated end-user assessment techniques for behavioural data to physical products and/or systems, and proposing a four-phase methodology to attain insights in user experiences and intentions.

5.1 POTENTIAL APPLICATIONS OF THIS WORK

In the area of a new product development, the proposed methodology may provide valuable insight in design-change process. For existing systems/products it can be used in after-sale support, namely in training to understand points that require more guidance or for maintenance of complex systems especially if they are used on a geographically-wide region. For training cases, knowing the weak point of the user in advance would increase the

efficiency of the training. More efficient training means decreased costs involved in the process and increased the user satisfaction with the product. For the maintenance support, VR based user assessment tool can assist the designers to identify problems faced during the maintenance of products. Similarly, user data may reveal better product design options which ease the maintenance of the products. Thus it assists in determining the points that require support without having to send support teams to the geographically dispersed locations for maintenance. Simply put, the developed methodology can be used whenever there is a need to understand if a product is functioning as expected on user's side. Its main advantage is that it provides automated data collection and analysis capability on large sample spaces, without being limited by location and time.

5.2 FUTURE WORK

Although we developed a methodology for automated usability evaluation support, the toolkit designed during this thesis has several limitations. Further improvements in both software side and data management part would increase the effectiveness of the method. Hence we propose the following future works:

- Eliminating all the non-automated analysis work, collecting all tools and proposed data analysis methodology in a user friendly interface
- Enabling different choices in the interface at each phase like different clustering algorithms for second phase and modules for the third and fourth phase. So that the designers can simply load their data and make their choices depending on the needs of their system and get the results.
- Today there are various techniques to provide visualization aid for data presentation after analysis. Such kind of a technique can be integrated to the system to provide better and easier understanding of end-results.

- Finally, in the proposed methodology we are dealing with sequential data analysis and adopted an event-based approach. We did not cover concurrent activities, activities that happen at the same time. So a definite future work can be to work on analysis of data that includes overlapping activities.

6 REFERENCES

Agrawal, R., Gunopulos, D., Leymann, F. (1997). Mining Process Models from Work Flow Logs. Research Report RJ 10100 (91916), IBM Almaden Research Center, San Jose California

Agrawal, R., Srikant, R. (1995). Mining Sequential Patterns. Proceedings of the International Conference on Data Engineering (ICDE), Taipei, Taiwan, March 1995.

Barbara, D. (2000). An Introduction to Cluster Analysis for Data Mining. Retrieved March 13 2006 from http://www-users.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf.

Borges, J., A. (2000). Data Mining Model to Capture User Web Navigation Patterns. Unpublished doctoral dissertation, University College London, London

Chen, M., Park, J. S., Yu, P. S. (1998). Efficient data mining for traversal patterns. IEEE Transactions on Knowledge and Data Engineering, 10(2), 209-221.

Cook, J. (1996). Process Discovery and Validation through Event-Data Analysis. Software Engineering Research Laboratory, Department of Computer Science, University of Colorado, Retrieved March 2, 2006, from http://citeseer.ist.psu.edu/87261.html

Dahan, E., Srinivasan, V. (1998). The Predictive power of Internet-Based Product Concept Testing Using Visual Depiction and Animation. Journal of Product Innovation Management 2000, 17: 99-109

Dick, S., Meeks, A., Last, M., Bunke, H., Kandel, A. (2004). Data mining in software metrics databases. Fuzzy Sets and Systems 145 81–110

Dunham, M. H., Xiao, Y. Efficient Mining of Traversal Patterns. (2001). In: Data & Knowledge Engineering 39 191-214

Fletcher, J. D. (1996). Does this stuff work? Some Findings from Applications of Technology to Education and Training. In: Proceedings of Conference on Teacher Education and the Use of Technology Based Learning Systems. Warrenton, VA: Society for Applied Learning Technology

Gamberini, L., Cottone, P., Spagnolli, A., Varotto, D., Mantovani, G. (2003). Responding to a fire emergency in a virtual environment: different patterns of action for different situations. Ergonomics, Volume 46, Number 8, pp. 842-858(17)

Ghosh, J. (2003). Scalable Clustering. The Handbook of Data Mining. edited by N. Ye, Arizona State University. Lawrence Erlbaum Associates Publishers, London, pp. 247-277

Hammontree, M., Weiller, P., Nayak, N. (1994). Remote Usability Testing. Interactions, July 1994, pp. 21-25

Hand, D, Mannila, H., Smyth, P. (2001). Principles of Data Mining. MIT Press, Cambridge, MA.

Hilbert, D. M., Redmiles, D. F. (2000). Extracting Usability Information from User Interface Events. In: ACM Computing Surveys, Vol. 32, No. 4, December 2000, pp. 384–421.

Hilbert, D.M., Redmiles, D.F., (1998). Agents for collecting application usage data over the Internet. In: Proceedings of Autonomous Agents'98.

Hurwicz, M. (2000).Web Virtual Reality and 3D – in VRML or XML?. Web Developer's Journal.

Ivory, Y. M., Hearst, M. A. (2001). The State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys, Vol. 33, No. 4

Jain A. K., Dubes R. C., (1988) Algorithms for Clustering Data, Prentice Hall

Karampatziakis, N., Paliouras, G., Pierrakos, D., and Stamatopoulos, P. (2004). Proceedings of the 7th ICGI 2004.

Kaufman, L., Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons.

Kempter, G., Dontschewa, M., Roux, P., Ritter, W. (2005). User Centered Virtual Prototyping: Do Users Feel Real with Virtual Prototypes? Paper presented at Human Computer Interaction International 2005, 22 – 27 July, Las Vegas, Nevada, USA.

Kempter, G., Weidmann, K.H., Roux, P. (2003). What are the benefits of analogous communication in human computer interaction? Universal access in HCI: inclusive design in the information society, edited by C. Stephanidis, Lawrence Erlbaum Associates Publishers, London, pp. 1427-1431.

Kuutti, K., Battarbee, K., Saade, S., Mattelmaki, T., Keinoen, T., S., Teirikko, T., Tornberg, A.(2001).Virtual prototypes in usability testing. Proceedings of the 34th Hawaii International Conference on System Sciences-2001

Lecerof, A., Paternò, F., (1998).Automatic Support for Usability Evaluation. In: IEEE Transactions on Software Engineering, Vol. 24, No. 10

Mannila, H., Toivonen H., Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. In: Data Mining and Knowledge Discovery, 1(3):259–289.

Milkova, E. D., Ikonomov, P. G. (2003). Using virtual reality simulation through product lifecycle. American Society of Mechanical Engineers, Manufacturing Engineering Division, MED, v 14, Proceedings of the ASME Manufacturing Engineering Division - 2003, p 761-767

Muehleisen, J. R. (1996). Mining Customer Support: A Progress Report on the Search for Usability Data in a Customer Support Database. Proceedings, 1996 STC Conference, pp. 302-305

Oh, H., Yoon, S. (2004). What Virtual Reality Can Offer To Furniture Industry? In: Journal of Textile and Apparel, Technology and Management. Vol. 4, Issue 1, Summer 2004

Sanderson, P.M., Scott, J.J.P., Johnston, T., Mainzer, J., Watanabe, L.M., James, J.M. (1994). MacSHAPA and the Enterprise of Exploratory Sequential Data Analysis (ESDA). In: International Journal of Human- Computer Studies, Vol. 41, 1994.

Siochi, A. C., Ehrich, R. W. (1990).Computer Analysis of User Interfaces Based on Repetition in Transcripts of User Sessions. ACM Transactions on Information Systems, 9(4), 309–35.

Siochi, A. C., Hix, D. (1990).A Study Of Computer-Supported User Interface Evaluation Using Maximal Repeating Pattern Analysis. In Proceedings of the Conference on Human Factors in Computing Systems (New Orleans, LA, April), pp. 301–305. New York, NY: ACM Press.

Stone, R. (2001). Virtual reality for interactive training: an industrial practitioner's viewpoint. In: Int. J. Human-Computer Studies (2001) 55, 699-711

Zettlemoyer, L.S., Amant R. S., Dulberg M. S. (1998).IBOTS: Agent Control through the User Interface. In: International Conference on Intelligent User Interfaces, Proceedings of the 4th International Conference on Intelligent User Interfaces, Los Angeles, California, pp: 31-37

Zhao, K., Liu, B., Tirpak, M. T., Schaller A. (2004). V-Miner: Using Enhanced Parallel Coordinates to Mine Product Design and Test Data. ACM 1-58113-888-1/04/0008

Zhou,C., Nelson, P.C., Tirpak, T.M., Xiao W. and Lane S.A.(2001). An Intelligent Data Mining System for Drop Test Analysis of Electronic Products Manufacturing. IEEE Transactions on Electronics Packaging Manufacturing, Vol. 24, No. 3

7 APPENDIX A: TEST I DATA

EVENT SEQUENCES RECORDED IN TEST I

CDIaGHOEMOEFIJKaMNOPRTaKLabQWZXRSTUVY
CDIEQEGaOPGHEFQIJMNKKLZXTURSabQWVY
CDIKEFaGMQGHCDGHKLIOPMNIJTQaQWabZXRSTUVY
CDEGMQOaRTEFMGHMIJMNOPKLQWabZXTURSVY
CDEGOQMaEFaRMOMGHIJMNOQOPKLQWabZXTURSVY
CDKLIJZXVYaOOEFGHMaZXQZXTVYRVYVYTTRVY
CICDIJIJKZZIJKLZXVYGHMEFMOORRabTUQW
CCCIZIQCDIJKLVZXVYMERGaRTCDCDIJMEKLEGaQZXEFGO
CIJCDEFOQGHKLQaMMZXZXQWabVYRTU
CDIKEaQTEGEFGHMOMMTQWabRTUIJKLZXVY
CDICDDIKLCDCDIKLZXVYQKLCDOERRGHGHMMabbTUV
CDEGMQOaRTEFaMGHMOMaQRTIJCDKLZXVY
CCDIKGEMOQRaTGHOEFMMOOIJKLQTURZXQWabVY
CDZOEFGHIJaOPMKLZabMNTQWZXRSTUVY
KCDIKZXVYEFCDZXOGMTaQWabTRCDCDOCDDOIJKL
CIJCDEFOQGHKLQaMMZXZXQWabVYRTU

COMPLETION TIME, NUMBER OF CANCELLATIONS AND REPETITIONS IN TEST I

Table A.1: Test I performance indicators

COMPLETION TIMES	REPETITIONS	CANCELLATIONS
14.92633333	0	6
3.396666667	1	9
8.130333333	2	6
5.255	4	10
3.874	4	6
5.059333333	2	11
10.12533333	8	11
9.135	3	11
8.127666667	9	23
10.118	2	8
13.43666667	8	12
6.325	0	14
12.94166667	0	18
15.13666667	16	10
16.51166667	2	17
9.853333333	2	13

8 APPENDIX B : CONSENT FORM FOR RESEARCH INVOLVING HUMAN SUBJECTS