Usability Analysis of Campus ONLINE, a Web-Based Course Registration System

KADİR BANİCAR

IŞIK UNIVERSITY
2008

Usability Analysis of Campus ONLINE, a Web-Based Course Registration System

KADİR BANİCAR

B.S., Computer Engineering, Işık University, 2006

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Computer Engineering

IŞIK UNIVERSITY
2008

IŞIK UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

USABILITY ANALYSIS OF CAMPUS ONLINE, A WEB-BASED COURSE REGISTRATION SYSTEM

KADİR BANİCAR

APPROVED BY:

Prof. Selahattin KURU (Işık University) _____
(Thesis Supervisor)

Assist. Prof. M. Taner ESKIL (Işık University) _____

Assoc. Prof. Seyhun ALTUNBAY (Işık University) _____

APPROVAL DATE : .. / .. /2008

USABILITY ANALYSIS OF CAMPUS ONLINE, A WEB-BASED COURSE REGISTRATION SYSTEM

Abstract

In this thesis we present our research in usability analysis of a web-based course registration system of our university which is Campus ONLINE, with using a proxy based user interaction and activity tracking system. The main goal of this study is to investigate the affect of usability with employing different tools, methods, and usability improvement techniques on Campus ONLINE, which is a web based tool developed as an online Student Information and Registration System of Işık University. The information which can be obtained using activity tracking is interesting for a number of scenarios. This includes a detailed tracking of all interaction with the displayed browser page, such as moving the mouse pointer around or scrolling the page. Additionally, the interaction should be tracked at the widget level, i.e. the mouse coordinates are mapped to elements like buttons, links etc. Finally, analyzing this valuable usability data acquired from system under investigation, we can reveal some design problems, usage bottlenecks, or usability issues that have never been discovered until today.

CAMPUS ONLINE WEB-TABANLI DERS KAYIT SISTEMI
KULLANILABILIRLIK ANALIZI

Özet

Bu tez çalışmasında web-tabanlı bir ders kayıt sistemi olan Campus ONLINE sisteminin kullanılabilirlik analizi proxy tabanlı kullanıcı etkileşim ve aktivite takip araçlarını kullanılarak gösterilmektedir. Bu çalışmanın asıl amacı değişik araçlar, metotlar, ve kulanılabilirlik açısından iyileştirmeye yönelik teknikleri kullanarak web tabanlı bir sistem olarak Işık Universitesi bünyesinde geliştirilmiş olan Ders Kayıt Sistemi Campus ONLINE, üzerindeki kullanılabilirlik faktöründen kaynaklanan etkileri incelemektir. Kullanım takip yöntemiyle elde edilebilecek bilgiler değişik analiz senaryoları için çok faydalı olabilir. Kullanıcının gösterilen tarayıcı sayfasındaki bütün etkileşiminin detaylı takibi ve ekranda fare imlecini nerelerde gezdirdiği gibi bilgileri içermektedir. Bunlara ek olarak, kullnıcının web sayfasıyla olan etkileşimi, örneğin mouse koordinatlarının sayfadaki butonlar, linkler vs. gibi elemanlarla eşleştirilp takip edilebilmelidir. Sonuda, bütün bu toplanan kullanılabilirlik açısından değerli veriler analiz edilerek, şu ana kadar gelen ve daha fark edilmemiş yada ortaya çıkarılmamış kullanılabilirlik problemleri, darboğazları, ya da tasarım problemleri ortaya çıkarılabilecektir.

Acknowledgements

I want to especially thank to Prof. Selahattin Kuru because it is very big chance and honor for me to work with him. I have learnt a lot of valuable things from him as a student, as a teaching assistant, and as a research assistant. Also, I feel indebted to Prof. Selahattin Kuru for his patience, support and advices as my thesis advisor. He always shares his experiences with me. It is very big pleasure to work with him.

I want to thank all my friends and colleagues for their close friendship and encourage. Also, I want to thank to Assist. Prof. M. Taner Eskil and Assoc. Prof Seyhun Altunbay, whose are in my jury, for their valuable advises on my thesis.

Finally, I want to give my special thanks to my mother Şehnaz Banicar, my father Faruk Banicar and my sister Merve Banicar. Thanks a lot for their unconditional love and support.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API	Application programming interface
CHI	Computer-Human Interaction
DET	Detection error-trade off
DOM	Document Object Model
EER	Equal Error Rate
FAR	False Accept Rate
GEMs	Game-like Elicitation Methods
HCI	Human-Computer Interaction
HF	Human factors
HMI	Human-machine interface
JVM	Java Virtual Machine
MMI	Man-machine interface
OMI	Operator machine interface
QUIS	Questionnaire for User Interaction Satisfaction
SUMI	Software Usability Measurement Inventory
SUS	System Usability Scale
UCD	User-centered design
UE	Usability Evaluation
UI	User Interface
UID	User Interface Design
UX	User Experience
WAMMI	Website Analysis and Measurement Inventory

Chapter 1
Introduction

Research in HCI has recently started by developing a computational science of usability. The main goal of this research is to develop a systematic approach to usability studies by adopting the strict experimental method of computational science. The first step is to develop instrumentation methods which collect data about software usability, in a manner related to instrumentation that has been done for analyzing performance. This data is then used to build computer aided models and simulations which explain the data in depth. Finally, different data visualization techniques can be applied onto collected data to help analyst. Same approach can also be used to model the browsing behavior of users on the web in our case [1].

Data collection from web pages is not as easy as any other instrumented methods. Rapidly developing nature of web creates problems when it comes to obtaining feedback about the usage of web applications. The data left by many interactive applications in the server's log file is not sufficient for extracting detailed information about the real usage of the application. For instance, it is not possible to say in which order the fields of a form were filled in.

In this work, we investigate means for obtaining more information about the usage of Campus ONLINE which is a web based tool developed as an online Student Information and Registration System. This includes a detailed tracking of all interaction with the displayed browser page, such as moving the mouse pointer around or scrolling the page.

Also, the interaction of user with Campus ONLINE should be tracked at the element level. For example, the mouse coordinates should be easily mapped to elements like buttons, links, and any other visual elements on the page. Later, this collected data can be combined with knowledge about the layout of the HTML pages in order to complete tracking of all user activity.

The information can be collected by using activity tracking systems are interesting for a number of scenarios. Even if the main purpose in our case is to conduct usability tests of a web based application, tracking approach is very flexible enough. Also, it can be used in order to track interactions during web application development, beta testing or continuous evaluation of live websites. On a more abstract level, it can be employed for profiling users and for implicit interaction with websites.

In the later chapters of this thesis details of how to obtain and collect valuable data will be explained based on some special tool usage. But at the beginning extensive set of tools research will be done. Later each tool will be tested or evaluated in terms of easy usage with our online systems. Easy integration is a key role for our systems because we will begin testing or analysis of a system already in use which is called Campus ONLINE. Because of that special case, we need to find transparent tools which can be easily integrated into or in front of the working web server. This kind of approach can only be achieved by using specialized proxy servers for usability data logging in order not to disturb users and collect their actual usage and actions related to the system.

The main goal of this thesis is to investigate suitable tools for making a usability analysis of an actively used system called Campus ONLINE and giving some insights about usage insufficiencies about system. Finally, we can reveal some design problems, or usability issues that has never been discovered until today, and propose a solution for fixing those issues. Also the most valuable deliverable of this thesis for the Campus ONLINE system will be an integrated usability analysis tool that can be used repeatedly for analyzing actual usage of the system.

Chapter 2
Background of Usability

The field itself is known under names like Computer-Human Interaction, Human-Computer Interaction, which is preferred by some who like "putting the human first" even if only done symbolically, User-Centered Design, Man-Machine Interface, Human-Machine Interface, Operator Machine Interface, User Interface Design, Human Factors, ergonomics, etc. Human factors and ergonomics have a broader scope than just human computer interaction. In fact, many usability methods apply equally well to the design of other complex systems, and even to simple ones that are not simple enough.

2.1 What is Usability?

Usability is a narrow concern compared to the larger issue of system acceptability. Basically, acceptability is the question of whether the system is good enough to satisfy all the needs and requirements of the users as well as other potential stakeholders. The overall acceptability of a computer system is again a combination of its social acceptability and its practical acceptability [2].

When a system is socially acceptable, we can further analyze its practical acceptability within various categories such as cost, support, reliability, compatibility with existing systems, etc., as well as the category of usefulness. Usefulness is the issue of whether the system can be used to achieve some desired goal.

It can be broken down into the two categories of utility and usability [3], where utility is the question of whether the functionality of the system can do what is needed, and usability is the question of how well users can use that functionality.

Note that the concept of "utility" does not necessarily have to be restricted to the domain of excessive usage. Educational software ("courseware") has high utility if students learn from using it, and an entertainment product has high utility if it is fun to use.

Usability is the extent to which a computer system enables users, in a given context of use, to achieve specified goals effectively and efficiently while promoting feelings of satisfaction [4]. Usability applies to all aspects of a system according to Nielsen [2], which a human might interact, including installation and maintenance procedures. It is very rare to find computer equipment that truly has no user interface components. More user-friendly (efficient, flexible, powerful) systems or websites are therefore easy to learn and easy to use.

2.2 What is Measured as Usability?

According to Nielsen, which is important to realize that, usability is not a single or one-dimensional property of a user interface. Usability has multiple components and it is traditionally associated with these five usability attributes [2]:

- **Learnability:** The system should be easy to learn so that the user can rapidly start getting some work done with the system.
- **Efficiency:** The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.
- **Memorability:** The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- **Errors:** The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.

- **Satisfaction:** The system should be pleasant to use, so that users are subjectively satisfied when using it; they like it.

From Nielsen's point of view, defining the abstract concept of "usability" in terms of these more precise and measurable components can we arrive at an engineering discipline where usability is not just argued about but is systematically improved, and evaluated even possibly measured [2]. Even if you do not intend to run formal measurement studies of the usability attributes of your system, it is an illuminating exercise to consider how its usability could be made measurable. Clarifying the measurable aspects of usability is much better than aiming at a fuzzy feeling of "user friendliness".

Usability is typically measured by having a number of test users, who is selected to be as representative as possible of the intended users, use the system to perform a pre-specified set of tasks. Also, it can be measured by having real users in the field to perform whatever tasks they are doing anyway. In either case, an important point is that usability is measured relative to certain users and certain tasks. It could well be the case that the same system would be measured as having different usability characteristics if used by different users for different tasks. For example, a user wishing to write a letter may prefer a different word processor than a user wishing to maintain several hundred thousands of pages of technical documentation. Therefore, usability measurement starts with the definition of a representative set of test tasks, relative to which the different usability attributes can be measured.

2.3 What are the Methods to Measure Usability?

Usability evaluation as defined by Mack and Nielsen consists of methodologies for measuring the usability aspects of a system's user interface and identifying specific problems [5]. Most usability techniques are quantitative and measure the performance, retrieval times, with success or failure rates. Qualitative assessments usually consist of observations and interviews.

There are different types of evaluation methods used to examine usability-related aspects of a system. According to Mack and Nielsen, the evaluation methods can be classified into four categories [6]:

- **Automated:** means hundreds or thousands of participants report their own behavior through a browser window that has an open text field and survey questions.
- **Empirical:** Usability is assessed by testing the interface with real users.
- **Formal:** Using exact models and formulas to calculate usability measures.
- **Informal:** Based on rules of thumb and the general skill, knowledge and experience of the evaluators.

On the other hand, Benbunan-Fich categorized the usability evaluation or inspection methods into four categories [7]:

- **Objective Performance**: Measures the capability of the visitors using the website in terms of time taken to complete specific tasks through the system.
- **Subjective User Preferences:** Measures the users' preferences to the system by asking them to elicit their opinions or use a questionnaire for rating the system being tested.
- **Experimental:** Based on controlled experiments to test hypotheses about design and their impact on user performance and preferences.
- **Direct Observation:** Inspect and monitor the users' behavior while they are interacting with the system to detect usability problems.

Most widely accepted name of all these methodologies described above is called "Usability Inspection". We can briefly explain main idea of the usability inspection as having evaluators that inspect user interface of given system. Basically, it is aimed to find fundamental usability problems. Also, some of the methods deal with usability problems that can affect severely the overall usability of the system under inspection. Many inspection methods can be applied at any step of the usability engineering lifecycle.

Usability Evaluation and Engineering perspectives and their respective set of methods can be seen from Figure 2.1 within much more integrated view. Development life-cycle in Figure 2.1 which gives us the big picture of development environment integrated with usability engineering methods.

If you have adequate resources such as time and money with direct access to the users, and skilled or experienced usability engineers or developers, you can use the whole set of methods given in the figure below. But in real life designers or developers need to select suitable set of evaluation methods based on certain factors. The factors include stage of design, novelty of project, number of expected users, criticality of the interface, cost of product and finances allocated for testing, time available, and experience of the design and evaluation team [9,10].

The following part is intended to overview the available methods used for inspection, evaluation and analysis of Figure 2.1.

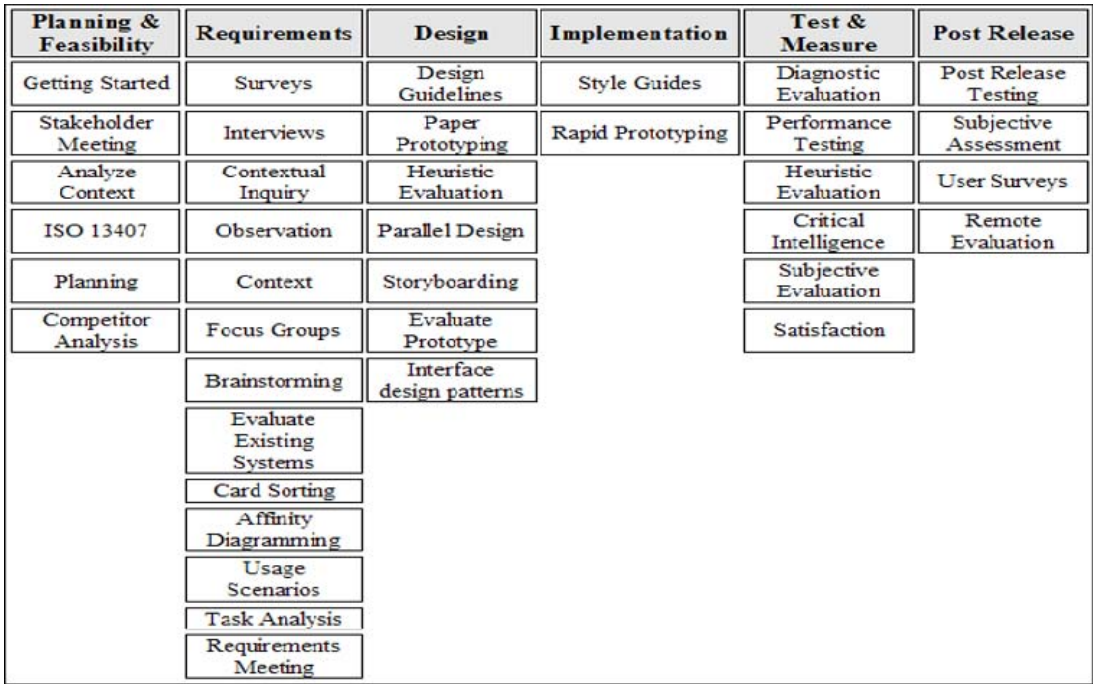


Figure 2.1 User centered design with usability methods integrated [11, 12].

- **Interviews and Observations:** are based on face to face and personal inquiry. A set of questions about what they do can be asked to users after test completed in interviews. On the other hand, observation is based on watching what they really do at the end of the session. Conducting both types of inspections can be done at the same time.
- **Surveys:** are other types of interview mainly but its not done face to face with users. A list of questions is asked and responses of the users are recorded interactively. Surveys are different from questionnaires also because it's more interactive, although it's not structured nor organized formally in focus groups.
- **Questionnaires:** are basically written set of questions. It is different that surveys because written set of questions given to the users in order to answer when they are ready to answer. Also, it is not interactive like any other type of interviews, so collecting answered questionnaires from users need more effort.
- **Card Sorting:** is mainly used to categorize pre-defined set of words into several categories. Categorization is done by users where they can sort or collect related words with each other. This type of inspection method is generally applied at the early development stage.
- **Heuristic Evaluation:** is the most informal method in others. It consists of pre-defined list of questions. Measurement is done according to those predefined metrics that help the expert to analyze the systems usability.
- **Heuristic Estimation:** is another form of "Heuristic Evaluation". This type is based on estimating the usability of different designs relative to each other. Estimation is done by experts depending on quantitative metrics such as expected user performance metrics.
- **Cognitive Walkthrough:** uses a detailed procedure in order to understand problem solving process of users. In other words, analysts are trying to get in the memory patterns of users in order to simulate their goals to achieve correct actions.
- **Pluralistic Walkthrough:** uses group meetings where users and developers met at an early design stage in order to discuss design factors of a sample scenario.

- **Feature Inspection:** is based on determining a list of feature sequences which are followed in order to accomplish a task. This list of sequences can be consist of key steps, tricky points, checkpoints for long tasks, and steps that require extensive experience on a proposed features.
- **Consistency Inspection:** is based on inspecting whether designed interface is working as it should be by designers.
- **Standards Inspection:** is based on inspecting an interface against defined standards or requirements for compliance.
- **Formal Usability Inspection:** is a type of inspection with combining different type of inspections into more procedural way. This type of inspection uses pre-defined strict user roles associated with both elements of heuristic evaluation and cognitive walkthroughs.
- **Usability Testing:** is based on conducting experiments in order to discover design specific problems related with usability. Basics of testing have a common background related with experimental psychology and data collected from experiments are interpreted in terms of heavy statistical analysis. Although, heavy statistical analysis can be done, it is not preferred as the only interpretation method. Because, through the test not only statistically meaningful data is being generated, other type of important data such as verbal and visual data is also generated.
- **Remote Evaluation:** is based on collecting user's feelings about the software being tested in a subjective way. This subjective assessment helps the evaluator to understand how users feel about the software being tested. The general method adopted for this inspection is to use standardized questionnaires in order to preserve subjectivity.

All usability methods described above can easily be categorized into two bold subjects such as group or single evaluator inspection by their application. Also, many evaluators can be attracted by their easiness of application because most of the usability inspection methods described above can be done by regular developers. On the other hand, this trick situation is not acceptable and should be avoided. Only for some of the methods such as heuristic evaluation or estimation can be applicable by developers in real life but even this is hardly acceptable for usability engineers.

2.4 How Usability Analysis is Done?

A usability analysis can help you to understand which parts of your web page are viewed by users and their respective navigational behaviors. This approach is useful for extracting analytical data from each and every one of visitors used your web application. Also, usage data collected can be useful in order to improve design of your web application and flow of interaction in between your users and your system. Usability is not something magical though it is simply about analyzing how a user interacts with a web page, more generally a system under inspection, in order to make it more user friendly than before.

Different type of analysis could be done on different type of data acquired from several types of methods. If we are doing a Heuristic Evaluation our analysis is based on pre-determined set of standard check list. So, we do not have to apply statistical methods on that data, we just conduct tests based on that check list with different tasks on target system. But if we are conducting some kind of survey or questionnaire specially designed in terms of usability such as WAMMI, SUMI, MUMMS, and QUIS, we can apply statistical methods on the results of specially crafted survey or questionnaires.

Same thing applies to the lab based usability testing or remote usability testing. In this case, we can play with raw data collected from test results, and apply statistical methods on that raw test data to create meaningful reports about usability of our system. In this usability analysis, we can analyze the quantifiable data such as:

- The time users take to complete a specific task.
- The number of tasks (or the proportion of a larger task) of various kinds that can be completed within a given time limit.
- The ratio between successful interactions and errors.
- The time spent recovering from errors.
- The number of user errors.
- The number of immediately subsequent erroneous actions.

- The number of commands or other features that were utilized by the user (either the absolute number of commands issued or the number of different commands and features used).
- The number of commands or other features that were never used by the user.
- The number of system features the user can remember during a debriefing after the test.
- The frequency of use of the manuals and/or the help system, and the time spent using these system elements.
- How frequently the manual and/or help system solved the user's problem.
- The proportion of user statements during the test that were positive versus critical toward the system.
- The number of times the user expresses their clear frustration or clear satisfaction.
- The proportion of users who say that they would prefer using the system over some specified competitor.
- The number of times the user had to work around an unsolvable problem.
- The proportion of users using efficient working strategies compared to the users who use inefficient strategies (in case there are multiple ways of performing the tasks).
- The amount of "dead" time when the user is not interacting with the system. The system can be instrumented to distinguish between two kinds of dead time: response-time delays where the user is waiting for the system and thinking-time delays where the system is waiting for the user. These two kinds of dead time should obviously be approached in different ways.
- The number of times the user is sidetracked from focusing on the real task.
- Questionnaires and interviews are also useful methods for studying how users use systems and what features them particularly like or dislike.

To determine a system's overall usability on the basis of a set of usability measures described above, one normally applies the statistical methods on the data collected from usability tests to discover success rates. According to the applied tasks the mean value of each of the attributes that have been measured is taken and whether these means are better than some previously specified minimum is to be determined.

Since users are known to be very different, it is probably better to consider the entire distribution of usability measures and not just the mean value [8].

This type of lab based usability test not only makes it hard to cover all possible tasks on a site due to the restricted number of participants but also rarely leads to representative results. The rather qualitative than quantitative data accumulation makes it difficult to find the majority of usability problems and the lab environment forces the user to act under observation.

However tests with large numbers of participants are very time consuming and expensive, but this side effect can be diminished by using remote usability testing tools. Proxy based usability analysis tools can be used to collect raw data as in the usual lab based usability tests. Furthermore, this kind of test tools give you the ability to conduct usability tests without interfering the subject and collect raw data as much possible as well as more information about user behavior while they are doing real tasks related to the system under test.

Chapter 3
Tools for Usability Analysis

Successful user interfaces, even for simple systems, are difficult to create. Particularly, designing interfaces for the Web is challenging, as completing common tasks online often causes inconvenience for users. If a website is difficult to use and the website's information is hard to read or doesn't answer users' key questions and if users get lost on a website they simply leave [3]. Unfortunately, Website usability continues to be a problem. Although many companies now have dedicated usability specialists on staff, and even more companies say that they want to improve the usability of their websites and intranets only an estimated 66% percent of corporate sites provide adequate usability [9]. Therefore, usability testing is necessary to identify and eliminate insufficiencies.

3.1 Categorization of Usability Analysis Tools

In general usability testing, a group of users that each representative of the target audience, are brought to a usability lab and asked to complete several predefined tasks. Their actions are monitored by a usability engineer or supported by advanced monitoring with special eye-tracking tools. Afterwards, the users are interviewed to gain a more detailed insight into the issue. However, traditional user testing techniques are inadequate and various Automated Testing Tools have been developed in order to make usability testing easier and as accurate as it is in traditional ones. They are used to evaluate usability from different perspectives.

In addition, various companies and organizations offer specialized services to conduct usability testing with the use of their testing labs. Even large organizations like Microsoft and IBM have their own usability testing labs and tools.

The major Usability Engineering processes can be categorized based on the involvement of users and systems in the method, as illustrated in Table 3.1. So, we can easily categorize the developed and dedicated tools to the usability testing and analysis with the related topics and dimension illustrated in the Table below because they are closely related.

Table 3.1 Dimensions of major usability methods supported [10].

	User-based Methods (with direct user involvement)	**Developer-based Methods** (without direct user involvement)
Methods Directly Involving a System	focusing on User-System Interactions • Prototyping • User testing • Observing users • Measuring performance	focusing on Developer-System Interactions • Expert evaluations • Certifications • Automated evaluations
Other Usability Methods	focusing on User-Developer Interactions • Interviews • Questionnaires • Collaborative methods	focusing on Developer-Documentation Interactions • Verification • Document examination • Model-based evaluation

From web based software point of view, we can also introduce another category for usability analysis tools such that Remote Usability Testing and Evaluation. Basically, this category is a variation of usability lab testing and user testing area but specifically tailored to overcome basic problems of traditional usability lab testing; problems mainly caused by increased costs, expensive testing equipments, and time consuming scheduling events with traveling costs of test subjects.

3.2 Remote Usability Testing and Evaluation

In the past years, there has been an increasing interest in remote usability evaluation which can be defined as usability evaluation but evaluators are separated in space and time from users. For clarifying the idea, the term *remote* used in this context of remote usability evaluation, is relative to the developers and refers to users not at the location of developers. On the other hand, the term *local* refers to the location of the developers [11].

In traditional laboratory-based usability evaluation, users are observed directly by evaluators. However, remote or distributed location of users disrupts the opportunity for direct observation in usability evaluation. Therefore, with remote evaluation, the network or in our case internet serves as a bridge between users and evaluators, taking interface evaluation to a broad range of networked users (e.g., representative group of users) in their actual work conditions.

Basically, there are two kinds of remote usability methods such as moderated and automated as seen below [12].

- **Moderated:** type is based on watching another person by another one, usage of computer, viewing their screen movements with a screen-sharing tool, and talking with them over the phone. The moderator watches and listens to where the participant runs into difficulty while interacting with the application or site, and records both the conversation and the participant's screen. It's pretty much the same as in-person testing, but without the user's facial expressions. Also known as facilitated remote research.
- **Automated:** means hundreds or thousands of participants report their own behavior through a browser window that has an open text field and survey questions. As the user navigates through a web site, they enter their feedback and answer page-specific questions in the browser frame. It's more than a survey, because it's still behavioral, but there is nobody watching and talking with the users. Some people call this unattended remote usability. The data is usually automatically distilled into a report with collection of given answers, click data, or exercise results.

As seen from Figure 3.1 the difference in between moderated and automated remote usability testing can also be differentiated as qualitative and quantitative measured values. Also, horizontal and vertical axis of Figure 3.1 give us the ability to understand what is being tested with in conjunction with what tool is being used. The Figure 3.1 below gives us some information about leading usability tools, not only remote usability tools but also survey tools, remote sorting tools, and other kind of web analytics tools with their respective place on the landscape. In the next part, detailed explanation of those tools included in landscape will be covered but basically moderated tools like Morae, UserVue or Ethnio can be incorporated into design anywhere. Other tools like wireframes or conceptual artifacts are only available at the pre-design stage where sketches or simple design decision are going to be taken. Tools like MindCanvas, WebSort and SurveyMonkey can be used to gather requirements for design at the pre-design or post-design stage. Any other tool in the landscape such as web analytics, Offermatica or UserZoom is useful for automated tests after design while collecting data from huge number of participant.

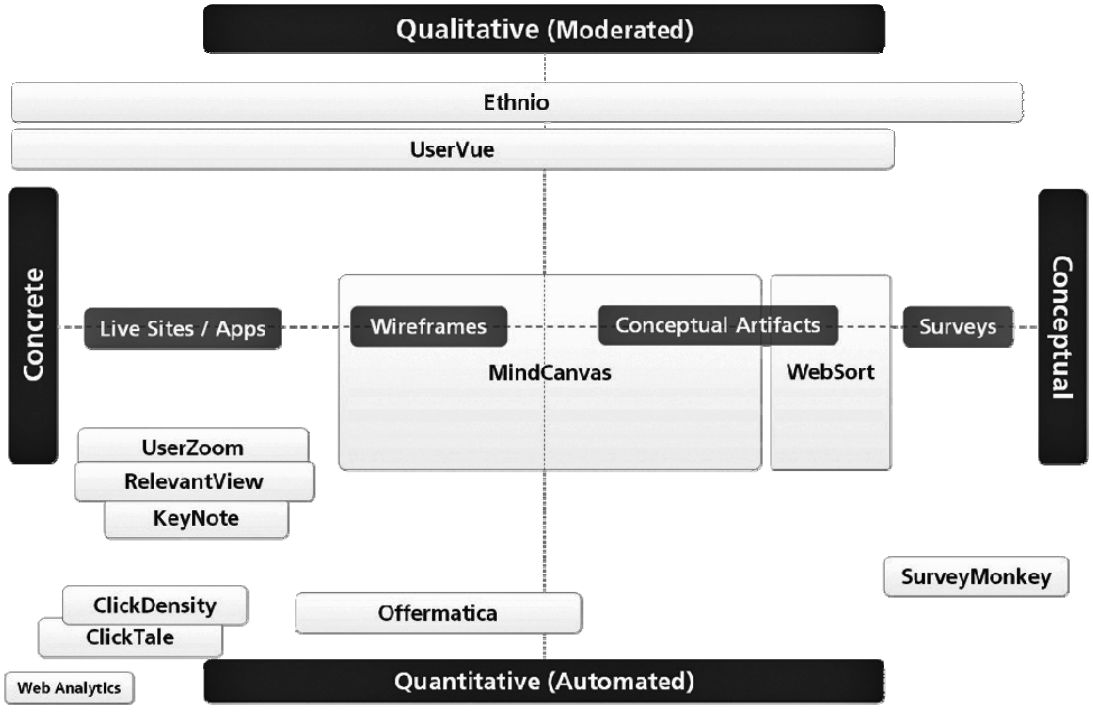


Figure 3.1 Dimensions of Remote Usability Landscape [13]

3.3 Leading Usability Tools

SUMI

SUMI [14] is widely used and accepted questionnaire whose standardization base and manual have been regularly updated. It is good for desktop products, but has also been used to evaluate command-and-control applications. It is a commercial product which comes complete with scoring and report generation software. It is designed and sold by the Human Factors Research Group at University College Cork.

SUS

SUS **[**14] is widely used and accepted questionnaire, developed by J Brooke in 1986 and not published until years later. It is very robust and has been extensively used and adapted. It is public domain and nobody has published any standardization base about it. As all the public domain questionnaires, this is the most strongly recommended.

QUIS

QUIS [14] is a developing questionnaire that has been modified many times to keep it current since its first appearance developed by Kent Norman. It is commercially available and is supported by Ben Shneiderman in his book. [15] Despite the lack of standardization and validation base, it has many supporters.

CSUQ

CSUQ [14] is a well-designed questionnaire developed by Jim Lewis and it is public domain. It has excellent psychometric reliability properties but no standardization base. This usability test looks at the interface of Web sites to see if the Web site does what it is supposed to do

IsoNORM

IsoNORM (Germany Only) [14] is a questionnaire designed to test the usability quality of software following the ISO 9241 part 10 principles. It is created by a team led by Jochim Puemper. Strong reliabilities are claimed for the sub-scales, although it appears there maybe a strong inter-correlation between them as well. Downloads and an online version are available, as well as articles about it (all in German.)

IsoMetrics

IsoMetrics [14] is a questionnaire produced by Guenter Gediga and his team. It is another attempt to produce a way of measuring ISO 9241 part 10, with reference to specific software features that may give rise to low usability data. It is therefore good both for summative and formative assessments. The questionnaire is well researched and detailed statistical information is given. Downloads of English and German versions are available online but there is lack of standardization base for it.

WAMMI

WAMMI [14] is a new questionnaire, designed to evaluate the quality of use of web sites. It is backed up by an extensive standardization database, and it is purchased on a per report basis. It is the result of a joint development project by Jurek Kirakowski and Nigel Claridge.

WAMMI was developed by Human Factors Research Group (HFRG) in 1999. WAMMI is an evaluation tool for websites. It is based on a questionnaire filled by visitors of a website, and gives a measure of how useful and easy it is to use the visitors found about the site. The WAMMI report provides the following information:

- Overall usability score and the general rating of a website.
- Detailed usability profile in terms of five usability scales: attractiveness, control, efficiency, helpfulness, and learnability.
- Detailed listings of those aspects of the website that visitors have found to be especially good or especially problematic.

NIST Web Metrics

The objective of the National Institute of Standards and Technology Web Metrics [16] is to explore the feasibility of a range of tools and techniques that support rapid, remote, and automated testing and evaluation of website usability. Web Metrics tool set consists of the following:

- Web Static Analyzer Tool (WebSAT) that checks the HTML code of web pages against usability guidelines, either its own set or a set of IEEE Standard 2001-1999 guidelines. It can check individual pages or an entire website.
- Web Category Analysis Tool (WebCAT) that lets the usability engineer to quickly construct and conduct a simple category analysis across the Web.
- Web Variable Instrumental Program (WebVIP) that lets the usability engineer to rapidly instrument a website so as to capture a log of user behavior on the site.
- Framework for Logging Usability Data (FLUD) which checks behavior of website users by capturing user interaction log.
- FLUDViz tool that lets the usability engineer to visualize and analyze a single usability session.

The methods measure either objective usability of a website or user's subjective perception about the website [16]. Objective measurements are based on evaluating a web based application with scanning its HTML code as stated in depth by NIST team. On the other hand, NIST team stated that subjective measurements are based on evaluating overall impression of the website on the users. The advantage is that the resulting measurement can easily be quantifiable.

However NIST Web Metrics team stated that the external factors such as connection speed, and cultural issues or other human factors are not considered. NIST Web Metrics places the users at the centre of usability evaluation and it is suitable for evaluating websites since a website is normally visited by many users from different backgrounds and different places [16].

Protocol Analysis

The protocol analysis or "think aloud" method was introduced by Benbunan-Fich from Seton Hall University. It is based on direct observation of a real interaction between the user and the system [7].

During the evaluation, the user is asked to carry out a predefined task using the system. At the same time, the user is asked to explain step by step his thoughts by "thinking aloud".

The user needs to explain his thinking process and reasons for his actions. The way of the user approaches a task and the reasons why problems occur during the user interact with the system is captured by using a concurrent protocol. Video or audio tapes are required for this evaluation process.

ClickDensity

ClickDensity [17] usability analysis tool has been unveiled by Box UK, a web development company based in the UK, enabling users to gain a visual insight into the behavior of website visitors.

According to Box UK, the tool is accessible and user-friendly, simple to setup and install. It records the position of every click made by visitors on a webpage. Data can be displayed in various ways including the use of heat maps, which provide a "virtual thermal image" of the density of clicks on a page section.

The ClickDensity tool offers full visitor behavior analysis, enabling website owners to see where visitors are clicking, identify "non-links" and measure link popularity. It also enables them to identify usability issues including determining where visitors are clicking that they should not be.

Morae

Morae [18] give you the power to reveal web based software usability issues without the expense and complexity of a usability lab. Morae is a software developed by TechSmith, whom is also developed online version as remote usability testing tool called UserVue, for usability testing and user experience research that helps you to identify website and application design problems and share them with stakeholders. With the insight Morae provided, you can make critical design changes that will improve usability with increased return of investment in customer satisfaction.

Ethnio

Ethnio [12] is a remote usability testing product developed by Bolt | Peters User Experience. Bolt | Peters is a research firm dedicated to studying how technology can be made easier to use for people. After leading the industry in conducting this research through Remote Moderated Usability Testing, the firm developed Ethnio.

Ethnio is developed as partially open source which makes it easy for other researchers, developers, and web marketers to quickly and easily conduct their own Remote Moderated Usability test sessions. Ethnio mainly developed for recruiting online users for remote usability tests and further it gives ability to moderate users while they are doing daily tasks.

MindCanvas

MindCanvas **[**12] is a remote research tool developed in order to gather insights about your customer's thoughts and feelings. Online surveys require users to complete boring html forms. MindCanvas uses Game-like Elicitation Methods (GEMs) to let online users participate in answering the complex questions that you face in designing a product or service. Visual Analysis Engine of MindCanvas lets us rapidly mine the data and create rich visualizations for you to explore. It has following benefits:

- Everybody hates surveys but loves games. With game-like elicitation methods users enjoy giving you design feedback according to developers of MindCanvas.
- You can easily interact with research results and also you can explore, embed results into presentations, or email them to colleagues.
- Also, they provide online collaborative research service that makes research step easy. Further, users do not need to struggle with statistics.
- You can either gather qualitative feedback with remote moderated studies or quantitative data with remote automated studies.

UserVue

UserVue [18] is an online service that lets you remotely observe and record user's desktops as they navigate applications and sites. Whether you're perform user testing, conducting user research, collecting design feedback, or collaborating on projects, UserVue gives you accurately and repeatedly user experience data.

ClickTale

ClickTale [12] records visitor's every action as they browse your website. You can generate and watch movies in order to understand their behavior and you can gain valuable insights and improve your website's usability.

ClickTale provides a new approach to us for website analysis in order to make optimizations and inspect usability related issues. Traditional web analytics services provide generally aggregated usage data for web pages. On the other hand, ClickTale provides information about insights of what users have been doing while they are interacting with your website. It generates movies of user browsing sessions with meaningful reports as well as aggregated web analytics. This way, we can expand our point of view in order to understand behaviors of users, which consequently results in improved website usability.

ClickTale is an online service hosted on their servers, so software installation on the server or client side is not needed. Also, users can browse your website without noticing any difference. The only thing you have to do is creating an account, which takes only a few minutes. Like any other web analytics tools ClickTale also uses small piece of JavaScript code in order to collect data from website under inspection. The JavaScript collects data and sends it to the ClickTale servers for further processing.

ClickTale can create movies of browsing sessions as well as "ClickTale Heatmaps™". Heatmaps can show you where users clicked on a web page or where users clicked but normally they should not be. According to ClickTale website they have been running for over a year on thousands of production websites [12]. ClickTale enabled web pages have been visited over 100 million times and over one million pages have been recorded [12].

ClickTale is not like other solutions according to its developers because it can accurately generate movies because they record the HTML of tracked web pages. Also, they are offering the only solution that works with session based and intranet websites. They have an advanced API to ease the integration of complicated scenarios. They also use a third party Content Delivery Network to deliver tracking related files to your visitors. So even in the rare event that tracking servers are down, your website will remain unaffected and continue to be protected by a large distributed server network.

VULab

VULab [19] is tool designed for remote usability research. Basically, it is designed for conducting usability analysis of online games but it is also suitable tool for conducting usability analysis of web based applications. Also, it can record audio and video of user's interactions without installing any application on the client side.

Researcher can design a series of pre-session and post-session questions with VULab in order to get feedback from users. After, users log into the system VULab displays pre-session questions if there any, then users are redirected to the website or application under inspection. Then VULab starts recording and every move of users are recorded into a database. At the end of the session the recording is stopped and users are prompted to answer post-experience questions set up by the researcher. All answers of users are recorder into database.

WebEffective

Keynote's WebEffective [12] is an easy-to-use online, self-service user experience research tool for conducting in depth customer experience and market research studies. WebEffective is the leading online research platform according to Keynote's but there are many other tools of the same type that will help you get insights into what your customers and prospects are doing and thinking.

WebEffective is driven by Keynote's extensive infrastructure which includes one of the most representative panels of over 160,000 real online users from a cross-section of demographics, languages, and broadband and dial-up connections [12].

WebEffective also have the flexibility to use other user panels of your choosing or to recruit visitors to your site. If you've been using focus groups, usability labs, and opinion survey, you should take a look at WebEffective because you can simplify your efforts with conducting user experience research from one central tool.

CrazyEgg

CrazyEgg [20] is a website tracking system that lets you track visitors' every click, then see the report as a visual heat map overlaid on top of your web site. The hotter an area visualized in heat map, the more clicks it's received.

The service will also give you an overlay containing metrics like number of clicks and percentage of total clicks, so you can see just where users are going. Developer Hiten Shah has previously said: “We capture the exact position of each click, as well as clicks on form fields, buttons and images. This way we can provide a very accurate heat map of the click stream”.

TapeFailure

TapeFailure [21] is designed as simple as possible to use tool. You simply sign up for an account, which takes only a couple minutes, and then create a unique identifier for each website you wish to track. After that, you just drop in a small piece of JavaScript into each page you would like to track, and from there TapeFailure handles everything else. TapeFailure tracks your users and records both tapes of your users' browsing and statistics about the users' browsing session. All you have to do is log into your account and from there you have the ability to view the recorded tapes and statistics.

RobotReplay

RobotReplay [22] is a Rails application written by Nitobi that allows you to record how your users use your site, and play it back later. You just have to include a piece of JavaScript on your site, and then user events are chunked and sent up to RobotReplay every now and then.

Once the user is done with their session you can see it, and play it back as it happened. The playback happens by using the browser itself. This means that at

anytime you can stop the playback and actually interact with the site. There is obviously much use from a usability standpoint of this tool.

WebQuilt

WebQuilt [23] is a web usability logging and visualization system that helps web design teams record and analyze usability tests. The logging portion of WebQuilt transparently gathers click stream data as users complete specified tasks. This data is then aggregated and presented in an interactive graph, where nodes of the graph are images of the web pages visited, and arrows are the transitions between pages.

To aid analysis of the gathered usability test data, the WebQuilt visualization provides filtering capabilities and semantic zooming, allowing the designer to understand the test results at the gestalt view of the entire graph, and then drill down to sub-paths and single pages.

The visualization highlights important usability issues, such as pages where users spent a log of time, pages where the users get off during the task, navigation patterns, and exit pages, all within the context of a specific task.

WebQuilt is designed to conduct remote usability testing on a variety of Internet-enabled devices and provide a way to identify potential usability problems when the tester cannot be present to observe and record user actions.

UsaProxy

UsaProxy [24] tool make collection of data possible to track user's every move to obtain highly detailed and useful information about the actual usage of interactive websites and web applications. Actions such as moving the mouse pointer around, scrolling a page, or fill out a form in a specific order are highly valuable Usability indicators.

In the scope of the publication "Knowing the User's Every Move - User Activity Tracking for Website Usability Evaluation and Implicit Interaction" for the WWW 2006 conference the Media Informatics Group of the University of Munich presented an implementation for detailed tracking of user actions on web pages.

The tool is based upon an HTTP proxy approach working with existing server and browser setups. In addition, the user's experience isn't altered due to the JavaScript technology that stands behind the client-side monitoring. The tracked interactions are logged along with the ordinary HTTP communication directly on the computer where UsaProxy has been installed.

Chapter 4
Usability Analysis of Campus ONLINE

In recent years, the web has increasingly become an important platform for applications that can be available from everywhere to end-users by internet. In contrast with the earlier web applications, newly developing technologies were not as simple as old ones. So, web applications or web sites are now using more sophisticated user interfaces than straightforward web page layouts, which make everything more complex to analyze in terms of user's point of view.

There is a noticeable tendency to move more of the application to the client [24]. Earlier web applications followed the standard HTTP request and response protocol, but many newer applications are Ajax based and only contact the server in order to fetch partial data [24]. Rapidly developing nature of web creates problems when it comes to obtaining feedback about the usage of web applications. The data left by many interactive applications in the server's log file is not sufficient for extracting detailed information about the real usage of the application [24].

4.1 What Usability Data is Measured

In order to increase usability of a web site or web based software, we should track and analyze performance of the respective systems under usage in terms of user's successfulness on doing their tasks. The key to the successful analysis is dependent on revealing usability problems caused by the design of a web site. This is basically done by collecting specific and related data about actual usage of the web site while users implicitly or explicitly interacting with the user interface.

This data logging enable us to understand user behaviors and usage patterns which help us to improve web site design, by changing the overall structure of web site or the navigation and flow between pages or the layout of web pages in order to make it much more usable by end-users. This type of data cannot be extracted from server-side logs because they are insufficient when trying to obtain logging data for user interaction. Following samples can show us the usability point of view:

- Conducting usability tests on web based applications, not all data can be obtained such as accurately collecting timestamps of all clicks
- Creation of general usage statistics, for example to obtain data about where users are clicking most on the web page
- Analysis of individual usage patterns, e.g. for self-adapting websites

In Chapter 2, the idea behind usability analysis and background of usability is explained but I give definitions about summary of what is measured in computational science of usability in Table 4.1.

Table 4.1 Summary of major usability characteristics

Usability Characteristic	Definition
Memorability	Ability for users to come back to the system and remember how to use it once they've been away from it for some time.
Learnability	Ability for users to learn the system easily.
Efficiency of use once the system has been learned	Ability for users to save time in their work once they've learned the system.
Error recovery and prevention	When the system presents an error message to users, it gives enough information for them to be able to continue with their work. Better yet, the system helps to prevent errors.
Subjective user satisfaction	Users' overall feelings about the system. Is it pleasant to use?

As seen from Table 4.1, in this chapter we can concentrate on main characteristics described as Memorability, Learnability, Efficiency, Error Recovery, and User Satisfaction. In order to measure data related to usability, we can use some tools specially developed for usability studies as described in Chapter 3. There are many tools for this area of study but only few of them are academically suitable to use in a research.

From this point of view we can easily concentrate our work of analysis on using open source or academically developed tools such as UsaProxy, WebQuilt, and NIST Web Metrics tool set. Besides of those tools mentioned, we can benefit some other analysis ideas from third party solutions such as heat map generation from click streams, mouse trails and click density. Also, other visualization techniques can be applied in order to make analysis step easier than heavy statistical analysis.

A usability test is a formal evaluation process that has its goal improvement of the usability of the product being tested. The formal Usability tests are generally covered by using specially equipped usability testing facilities or usability labs. In the lab test, real users are invited to the lab to use and test the product while every move of participants are observed and recorded. But in our case, we do not have any special usability test equipment or usability test lab in our university, so that means we will collect usability data by using remote tools such as proxy tools. In the next section basics of those tools will be explained and then we will discover how to analyze and what is going to be visualized from collected data.

4.2 How Usability Data is Measured

In order to address the issues that are mentioned in the previous section we need to concentrate on collecting further logging data in addition to the server-side logs. After taking a look at related efforts from Chapter 1 through Chapter 3, it was decided that the best way to log user interaction for web applications is to use Ajax technology for the logging task. However, developing an Ajax based logging technology from ground zero is tedious and more complex than one would think at first because of various criteria such as the fact that the logging should not be invasive.

The requirements above maybe confusing a little but tracking must take place either at the client or server side in order to make analysis possible, which implies that significant changes in the setup should be made. Luckily, there are already developed various non-invasive and flexible data logging systems. So, we can use these tools without developing our own solution for tracking and without making any changes in the server-side pages. In the next sections of this chapter those tools will be compared and explained in details. For now, we are trying to explain their working structures.

This ability gives us a great flexibility and possibility for making a usability analysis of our production website. Furthermore, we do not have to install any additional software at the client side with using HTTP Proxy approach, which is inserted between the client and server as seen in Figure 4.1. [24].

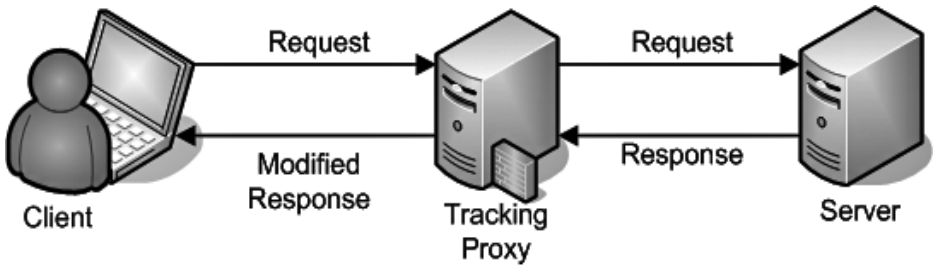


Figure 4.1 The tracking HTTP proxy [24].

The main idea behind the HTTP proxy for activity tracking is injection of JavaScript into all HTML to monitor mouse movements and other types of interaction. Basically, all HTTP traffic such as requests and responses are intercepted by proxy. Then, proxy outputs a log file with details about any requests sent to servers as well as the responses that a server sends back to the client.

If we will be able to log only the requests, then the data collected by the HTTP proxy would not be different from the standard web server logs. Only things like the URLs of requested pages and input data typed into forms by the user would be available. This is not sufficient for the usability analysis purposes of logging the user actions in detail such as tracking mouse movement.

This type of transparent data collection ability is developed by two tools in our list. First one is UsaProxy, with many other useful interaction tracking capabilities. The other one is WebQuilt. Sadly, WebQuilt fails to fulfill detailed user activity tracking requirement because it only serves as a proxy supported user testing tool not user tracking or action tracking tool. So, we choose UsaProxy tool even if they are designed in same way their setup and deployment greatly differs in usage.

On the other hand, WebQuilt is really a good piece of remote usability testing software especially for instrumented environments, but when we tried to deploy it on our servers we faced huge problems on the client side tests such as improper page layouts generated by WebQuilt proxy. This affect is not acceptable for our case, because our online system is under heavy load by students, instructors, and staff of the university. We made some other test deployments without deploying WebQuilt on real server but we could not manage it to work properly.

The only tool left in our hand is the UsaProxy. Even if WebQuilt has the same idea behind its roots, UsaProxy is worked like a charm for our environment. Because it is entirely written with java, besides it is open source. Also, UsaProxy has transparent server side deployment option, indeed this makes it suitable tool for our case. WebQuilt has some incapability for the server side usage because it came with integrated Apache TomCat web server which acts unexpectedly as a proxy for our case.

It should be noted that in contrast to “normal” HTTP proxies, UsaProxy does not perform any caching of content; it only acts as a filter between the client and the server as seen in the Figure 4.2. UsaProxy passes on images and other data unmodified at top while HTML content is modified at bottom by adding special JavaScript. Also, server response is recorded and identified via logging ID [24].

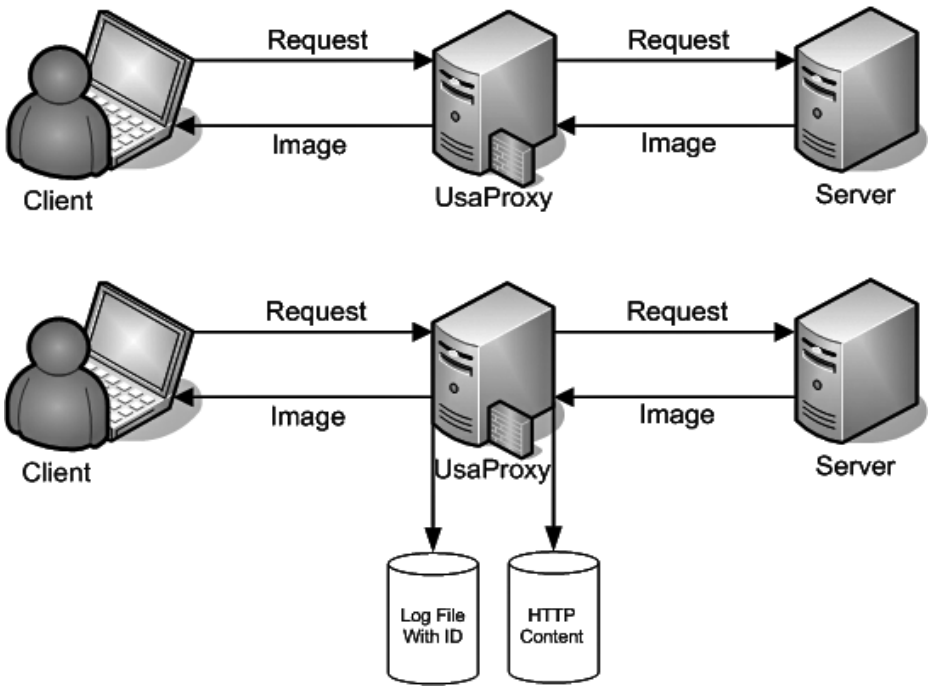


Figure 4.2 UsaProxy behind the scenes [24].

According to Atterer the monitoring and logging of user actions must proceed accurately, in real time and in a reproducible way for a usability engineer. Also, a set of features that are common in ground to standard usability tests should be provided with an accurate listing of what the user actually did while performing the predefined tasks. In order to be able to reconstruct website usage with a measurable and reproducible way, UsaProxy give us the possibility to track the following user actions in general context [24].

- Navigation behavior metrics across multiple websites, such as the most frequently used path through a website, key entry and exit pages, page views, page requests, and erroneous path selections with recovery times.
- Time-based metrics such as the average time spent on a page, total amount of time the mouse pointer hovers over elements to detect whether the user hesitated over other interesting links/text areas before clicking on it, and time taken for form fields, click frequency, general typing speed.
- Actions of a user within a web page, related to mouse input. This includes the absolute or relative mouse position according to the windows size. Identification of the elements under the mouse pointer. Both clicks and

hovering over elements need to be logged. If the user is familiar with the task, the recorded data must show the intentional and straight mouse movements.

The UsaProxy allow us to record as many user actions as possible if it is deployed properly on the server side. This is achieved by executing JavaScript code inside the user's browser which is automatically injected in to web pages by UsaProxy. For the current version of UsaProxy, the logged data includes the following information:

- Loading and resizing of HTML pages. Resizing means that the window for the current page is closed or a different web page will be loaded.
- Giving the input focus to the window or blurring with taking focus away
- Mouse clicks on the page. UsaProxy records the page coordinates from the top left corner of the document
- Also, UsaProxy records the DOM location of the HTML element that the user clicking on it or hovering over it.
- Mouse movements, in page coordinates
- Scrolling inside the browser window
- Key presses done by user to achieve a task
- Selections such as text in the screen or selection of HTML elements
- Changes in the context of HTML elements

Logging idea behind the HTTP proxy with filtering responses and requests of client and server in order to track what is done in HTML pages was demonstrated in Figure 4.1. UsaProxy improves this technique because it logs more detailed information in the log files and automates the process of adding JavaScript to the pages.

In addition to the client-side actions outlined above, UsaProxy also records the browser's HTTP requests and corresponding server HTTP response headers. In those cases where the response has a Content-Type of text/html, the entire response body is also recorded. All log data is written to the file system of the computer that the proxy runs on. [24]

4.3 Data Visualization

According to Atterer and UsaProxy visualization advises, it is possible to visualize usage data collected with UsaProxy range from ordinary listings of website statistics to complex screenshot images. The design of the UsaProxy gives us the ability to imagine the following type of visualizations can be possible [24]:

- According to Atterer [24] accurate traffic reports that include listings of all visitors, their page visits such as hits and views, visit entry points, the most common navigational paths through a website, window width, most common browsers and platforms with additional usability-related data such as the average time spent on the page before a link was clicked, scrolling activities, window resizing and average window size, popular mouse positions, the most frequently hovered-over or clicked links/buttons, and so on can be visualized.
- Also, we can visualize the visited web pages and paths through a site using special diagrams, similar to the visualization module of the WebQuilt application according to Atterer. WebQuilt team is developed a visualization tool that can generate a graph diagram with thumbnail pictures of the pages connected to each other with weighted directed arrows [23]. For instance, emphasized arrows indicate more heavily traversed paths in ample visualization. This visualization maybe combined with a number of semantic or statistically meaningful metrics such as the time spent on the respective page, or buttons that were pushed.
- According to Atterer [24] mouse movements can be visualized by lines ranging from simple, edged constructions to fluid, dynamic paintings. The representation can vary depending on speed or direction of movement. For example, the line width can be changed according to the velocity.
- Overlay of the usage data and the web metrics on the currently evaluated web page is imaginable according to Atterer either using a screenshot or directly within the browser [24].

  For example, click density of links and buttons maybe displayed in a dedicated part of the screen together with other statistical information. Mouse movements could also be overlaid on the web page screenshot.

- As noted from log data samples, a replay of the actions of a certain user on a web page appears feasible as stated by Atterer [24], either a live replay for customer support or a replay which happens later, for example while interpreting the results of a user test.

4.4 Analysis of the Measurements

So far, we made a research about suitable tools that have flexible and transparent but powerful logging capabilities as a solution for usability analysis of Campus ONLINE. We have introduced a sophisticated, yet flexible and transparent solution for user activity tracking called UsaProxy which is the most suitable tool for our online tracking purposes.

Compared to other available solutions, UsaProxy allows user tracking for web applications without manual preparation of web pages for tracking. It also does not require control over client or server machines. Also, UsaProxy system is an open source project, which leverages the power of free development and modification of the proxy server to respond our needs. Those unique features of UsaProxy also help us to reduce the cost of usability evaluation.

As explained before in the preceding chapters and in the data visualization section of this chapter there are some other type of visualizations for making analysis easier and log data much more human readable. But neither WebQuilt visualization tool nor UsaProxy is capable of generating reports or suitable visualizations from the log files collected with UsaProxy.

This situation makes analysis step harder for us to conclude right now and directs our research into another point, which is the integration of UsaProxy with Campus ONLINE. Even if we have been accomplished deployment of UsaProxy manually, collected data will be useless unless some sort of analysis tool is going to be used on it. So, development of an analysis tool is inevitable for our research right now.

The development of an analysis and management web front-end for UsaProxy and adding integrated reporting and analysis features into it will give us the ability to

control UsaProxy at any time from anywhere. Furthermore, we should also have to design an integrated data visualization solution for Campus ONLINE with using collected data from UsaProxy.

After that, UsaProxy integration with Campus ONLINE will be completed and applicable for continuous, simple and repeatable usability analysis. In the next chapter we will explain details of how integration with UsaProxy is done, what and how was developed as an extended web based usability analysis tool for Campus ONLINE.

Chapter 5
Implementation of Integrated Usability Analysis Tool

The main goal of this thesis is to investigate suitable tools for making a usability analysis of an actively used system called Campus ONLINE and giving some insights about usage insufficiencies about system. Even though this goal seems to be achievable easily at the beginning of this thesis research, some surprises occurred throughout testing, analysis and deployment try outs. Our research revealed that there are many tools in use but neither good and known tools are not free nor they totally fulfill our requirements for doing usability analysis of Campus ONLINE.

The most promising tool is UsaProxy as we said before in the preceding chapters, but even UsaProxy is not totally covers all requirements alone. The most wanted part for an analysis tool is visualization but UsaProxy does not even bundled with any visualization tool. This makes things harder for who is going to make analysis.

The next surprise for us appeared when we decided to use UsaProxy as an analysis tool for doing analysis of Campus ONLINE with it. UsaProxy logs the collected data from user events as a pure text file which needs special care for extracting useful data from that log file in order to exploit it for usability analysis.

Another difficulty appeared is caused by the deployment procedures of UsaProxy. It can be easily deployable tool for various types of configurations, but in our case it should be deployed on the server-side in order to work correctly. This creates a limitation in terms of flexible configuration because whenever usability analysis has to be done it should be configured at the server side manually.

So, starting and stopping the UsaProxy needs physical access to the server which is impractical for analysis of a web based software. Also it means that special configuration should be made every time to manage UsaProxy.

All described difficulties such as management, deployment, and visualization features lacked of standard UsaProxy should be added into the design with a web front-end which makes controlling all of the features from one centralized management point.

The next sections of this chapter will extensively focus on the details of how web front-end is implemented in order to centralize configuration, management and deployment of UsaProxy with newly added features such as data import, export, extraction and visualization of data from UsaProxy logs.

5.1 Web Front-End Design

The basic idea behind the web front-end is to collect each and every one of developed solutions into one central management panel. This way all usability analysis process can be controlled from one point also analysis results can be acquired from one place which would be an integrated usability analysis tool of Campus ONLINE.

Designing a web interface seems to be easy but without functional parts, which makes the interface meaningful, it's not much different than just a sketch on a canvas. So, initially design of web front-end is just used for clarifying what should be in the design. This means which functionalities are going to be displayed on the screen and where they are going to be placed to make management easy.

Another idea behind the scene is that web front-end should be designed in a simple yet powerful way. Throughout out my thesis research, I researched what usability is and how it can be applied on user interfaces of any HCI model. One thing superimposed on every HCI related paper is simplicity of design which makes your interface usable.

As you can see from Figure 5.1 below design of the web front-end simplicity rule is applied. It is designed as a separate interface in order to manage usability analysis process. But it has the flexibility of integration into any other web based system.

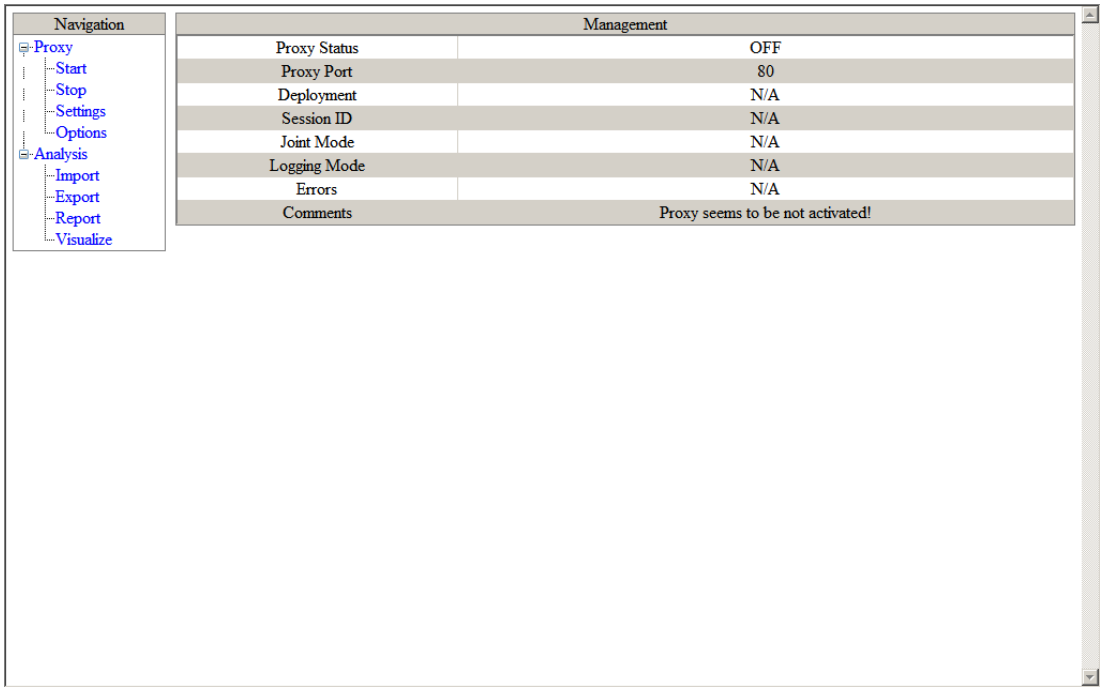


Figure 5.1 Initial screen of web front-end.

Navigation and management parts are divided into two separate windows in browser as seen from Figure 5.1. Management pane shows the current status of the proxy with deployment details. These details are required because you have to check the status of the UsaProxy before initiating any activity related to proxy. Navigation pane is designed as a tree structure which helps us to categorize related function under each other.

As you can see there are two main categories under navigation pane which is “Proxy” and “Analysis”. Under “Proxy” category we can control UsaProxy easily by clicking just related links. For example, apparently seen from link names you can start, stop and configure UsaProxy deployment. In order to work this interface correctly with UsaProxy, it should be deployed on the server where UsaProxy is installed

Under another part “Analysis”, we can handle usability related useful functions such as import, export, report, and visualize. Actually this part is useful after UsaProxy is started and some usability data collected with it and then we can extract required usability data from UsaProxy logs. For example, import feature under web front-end is used to initiate parsing process of UsaProxy logs. After parsing process is completed, we can export related data as an excel file for further processing with other tools or we can use integrated reporting capabilities of web front-end.

Further, we can visualize user sessions in order to understand their behaviors of while accomplishing their tasks. Also, we can analyze how they accomplished their tasks with analyzing what pages they viewed, what paths are used by them, and how they used their keyboard or mouse successfully manage their tasks. All this type of actions can be done online by the usage of web front-end but as we stated before those are just sketches of what can be done not how they can be done.

In order to accomplish and make it working real front-end, we need to implement some other background solutions. Firstly we need to design a wrapper interface for UsaProxy in order to make it controllable by web front-end, and then we need to implement a log parser for processing UsaProxy logs to be able to extract useful data. Log parser stand alone is not useful for us because we want to use extracted data from logs later on by querying them for doing analysis, so we need to design a database in order to save extracted information. Finally, we can query on database to make visualization about paths, pages, mouse trails, and mouse clicks. The details of how these features are implemented will be explained in the next sections of this chapter.

5.2 Proxy Wrapper API Design

One of the main problems that affect the whole analysis process is UsaProxy is not accessible from web interface. Also, this means that it has not any proper API to allow control from outside of UsaProxy. UsaProxy mainly designed as a stand alone proxy, but we need to design a proper API for exposing it as a web based tool.

UsaProxy is written by using Java programming language. This can be seen as an advantage at first but when internet and www comes into the arena it became a disadvantage. Even if java can be accessible from web pages as an applet, UsaProxy does not have any implementation for applet interface. Also, applets are too restricted java programs in order to authorize access from web to system resources. This creates a problem for us because we want to get full access to JVM in order to start and stop UsaProxy as a background process.

The only solution left in our hand is to design a wrapper API for UsaProxy in order to control it from web. In our case Campus ONLINE runs on IIS with ASP and supports ASP.NET. I decided to design wrapper API in ASP.NET with using C# language because this choice gives us a great control over system resources and helps us to access everything on the server.

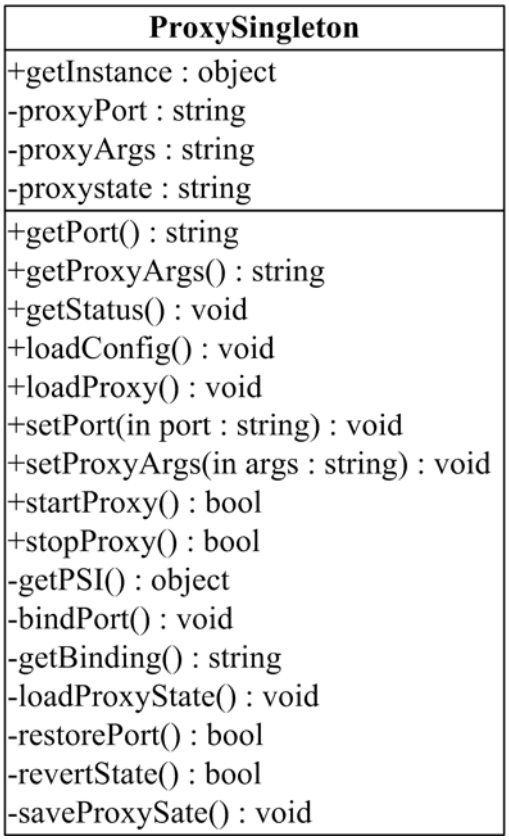


Figure 5.2 Proxy Wrapper Class

As seen from Figure 5.2 proxy wrapper class is designed as a singleton which its name implies. This means singleton design pattern, which guarantees that only one instance of the same class can be created, is applied in the proxy wrapper API design. This is an explicit requirement for proxy wrapper API because when there are multiple instances of the same class in web front-end this makes the system unstable and makes UsaProxy not controllable properly.

Also, Figure 5.2 reveals the inner methods of proxy wrapper class which gives a total control of UsaProxy to us. For example, we can start and stop UsaProxy. We can configure its deployment settings such as server port, remote host name and remote port. We can load different configurations from database or any other file and pass various arguments to the UsaProxy. Further, we can save its current state information in order to make persistent access to object state. Finally, when it has been started or stopped by using web front-end it can configure IIS server metabase in order to bind UsaProxy to specified ports.

5.3 Database Design

As mentioned in introductory part of this chapter and explained in web front-end section database design is inevitable requirement for analysis purposes. UsaProxy only gives us a text based log files, which is impractical to use for data query without processing. Database need is emerged in here to store extracted information from log files. Database also gives us the flexibility to query usability data later on UsaProxy has been stopped or while it has been working.

At first I want to exploit open source advantage of UsaProxy in order to integrate an embedded database into it, but when I deeply examined the code of UsaProxy this idea became an extraordinary effort for me. Because its structure is so dependent on text files and text based processing, which makes impossible to embed or integrate a database into it without completely re-engineering. So in short, I decided to design an external database solution not an embedded one into UsaProxy.

Campus ONLINE is a web based system that is working with a database already in use, but we prefer not to integrate UsaProxy database into it. Because it will make

database design dependent on the Campus ONLINE and also will create some portability issues about usability data. When portability comes one step forward, embedded and small databases are the best because their memory and storage requirements are lesser than production type of database servers.

In this case, ADO.NET supported embedded database solution will save us, which means MS Access database is suitable for our purposes. It also has ADO.NET supported drivers and also it is supported by ASP.NET technology. MS Access database also will lessen our burden of portability issues because it is a stand alone file that can be copied anywhere without problem.

Another challenge for us is about how we have to design the database in order to effectively and efficiently use it with UsaProxy logs and also with the developed Log Parser. In order to understand design issues with those dependencies, we have to examine UsaProxy logs deeply because our data will be extracted from UsaProxy logs. When we examined UsaProxy log files, we can easily categorize UsaProxy log entries into two types: first one is “HTTP Traffic” and second one is “Event Traffic”.

We can explain “HTTP Traffic” as requested pages by user in their browsing sessions. Also, “Event Traffic” can be explained as user’s interaction with requested page. Furthermore, event traffic can contain event types, session id, mouse movements, clicks with id, name, value and text of DOM elements and many more trackable values by UsaProxy in requested page. On the other hand “HTTP Traffic” contains IP, Date, Time, url, and server data id.

As you can see from Figure 5.3 that categorization explained above and related attributes with it is used in database design. “UPL_HTTPTraffic” table is designed for category one in order to store ip, url, date, time and sd. But category two entry is much more complex to store it in one table than category one. This complex logging structure of UsaProxy needs to be separated into more tables in order to store related and meaningful data with each other.

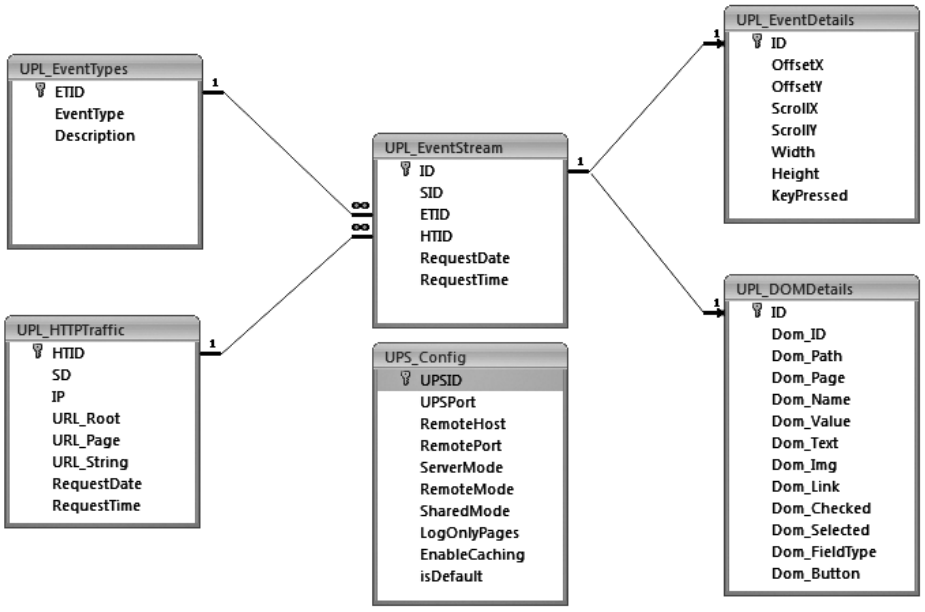


Figure 5.3 Relational Database Design

When we examined category two log entry in depth we can easily see that can be decompose into event types and details of events. At first this seems to be enough for storing category two entries but actually its not. Details of events can be split into two separate tables because it consists of DOM related details and event dependent data.

As shown in Figure 5.3 “Event Traffic” entries separated into four tables. Main table is “UPL_EventStream” table. This table relates all tables with each other by holding respective ID’s of each table. In more details, “HTID” is a foreign key in this table which is used to relate events with page requests. “ETID” is the foreign key of “UPL_EventTypes” table to relate type of events tracked. “SID” is an attribute of “UPL_EventStream” table store session id of users which can be extracted from logs like “RequestDate” and “RequestTime”. “ID” of “UPL_EventStream” table is used as a foreign key in two separate tables called “UPL_EventDetails” and “UPL_DOMDetails”. These two tables are designed to store all details related with “UPLEventStream” table.

As you can understand from relations between them, one “UPL_HTTPTraffic” record could have many records in “UPL_EventStream”. Also one “UPL_EvenTypes” record could be related with many “UPL_EventStream” record. But one “UPL_EventStream” record could only have one record in “UPL_EventDetails” or “UPL_DOMDetails” tables or in both tables. These tables are joined with “UPL_EventStream” as LEFT INNER JOIN by design because not any one of these tables includes the whole ID’s of “UPL_EventStream” table.

Before these tables separated into two tables, they were one huge table which includes all attributes of those two tables with one to one relation with “UPL_EventStream” table. But when all attributes examined we can decided to split related attributes into two separate tables for performance reasons. “UPL_EventDetails” table holds the details of mouse, screen, and keyboard related events where as “UPL_DOMDetails” table holds only the details related with DOM elements such as links, texts, values of forms, and field types vs. The only unrelated table with each other tables is added for managing configurations and deployment details of UsaProxy.

5.4 Log Parser Design

Throughout this chapter and preceding chapters we have explained what is needed for integrated usability analysis tool. We described the main functionality of web front-end and so far we explained the database design but this part of implementation is the key part for extracting data from log files. Log parser should be developed on top of our database design because results of parsing process will be stored into the database.

Also, database design will guide us while designing log parser because we designed database according to the log file structure of UsaProxy. Even database structure will help us to detect attributes in UsaProxy logs logically; we should deal with text files and read it to extract attributes in real. Log parser itself should be memory efficient, fast and very flexible for future developments in order to adapt new attributes into its parsing capabilities.

For preserving memory efficient parsing operations we need to use associative array like dynamic data structures. This type of data structures can change from language to language but their basics are common Hash Tables. Also, C# has got “Hashtable” class for storing (key, value) pairs into it. Hash tables of C# is extended class implements the hashing functionality but its base class is called “Dictionary”, which is a generic type of data structure. We will concentrate our focus onto “Dictionary” data structure and we will extend it like Hash tables in order to define our special data type for using with log parser.

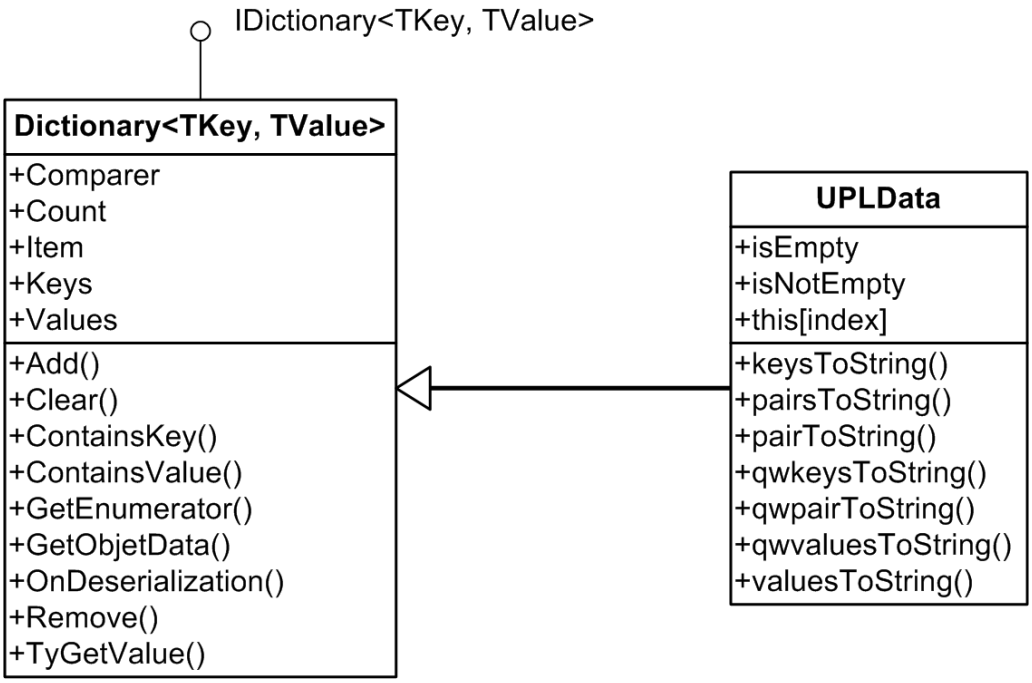


Figure 5.4 Special data type of log parser for storing extracted attributes

As you can see from Figure 5.4 class diagram, UPLData class is an extended class from Dictionary<TKey, TValue> generic type base class. Our parser data and extracted attributes are mainly is type of string. So, UPLData class extends Dictionary<String, String> base class and adds some extra methods into it according to our needs.

First, UPLData class is extended to change indexing property of generic Dictionary class. We need to insert and delete (key, value) pairs dynamically but access its pairs like using an array. So, we override its indexing structure which can be extended in C#. After indexing structure is overridden, we can insert (key, value) pairs just like using arrays. For example, UPLData[“any key”] = “any value”; this will give us really great flexibility while working with strings in log parser.

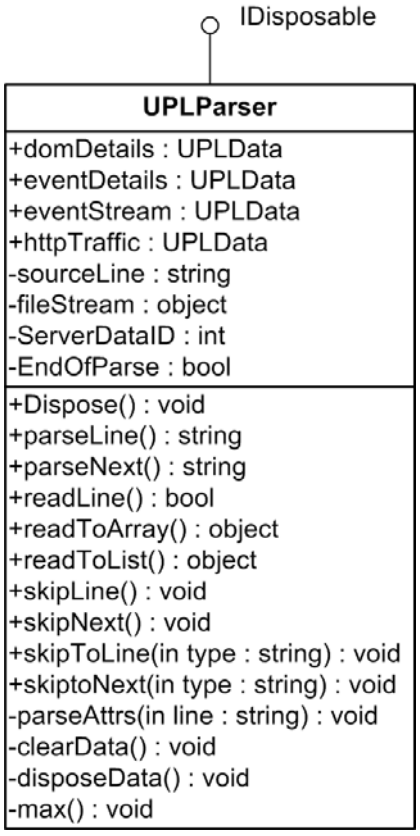


Figure 5.5 Class of log parser

As seen from Figure 5.5 “UPLParser” class implement “IDisposable” interface, which automates garbage collection in C# programming language. This interface should be implemented in a special way and there is a MS best practice design pattern called “Disposable Pattern”. With applying disposable pattern, we enhanced memory usage of log parser. Also, we designed dynamic data structure called “UPLData” class which will coordinate well with this design pattern.

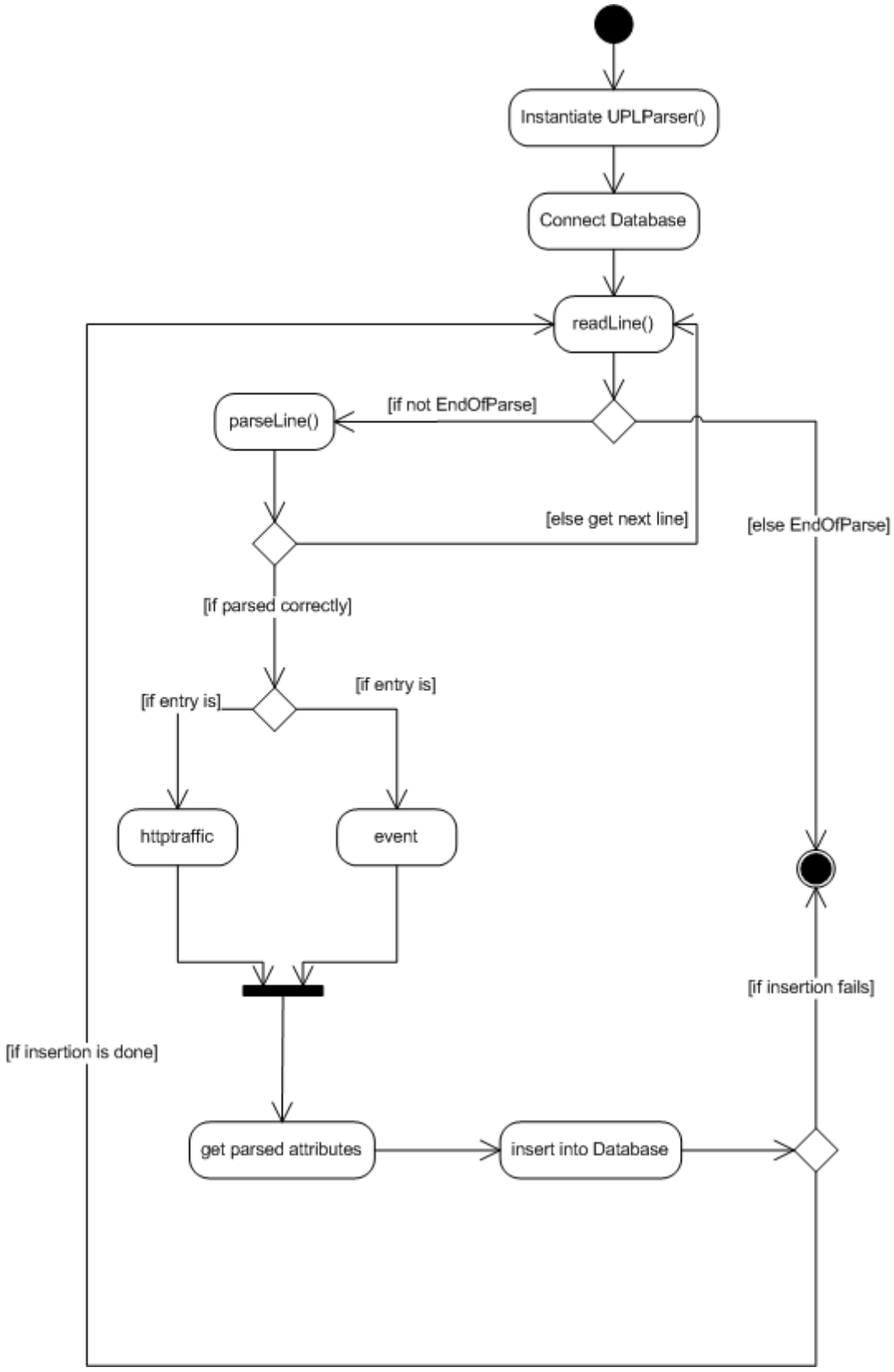


Figure 5.6 Activity diagram of parsing algorithm

As we said before log parser should have to deal with text files. In order to extract one or more (attribute, value) pairs from log files, we have to open log file and sequentially process it line by line as seen from Figure 5.6. While processing each line “parseLine” method of “UPLParser” class as seen from Figure 5.5 and Figure 5.6 will be applied on them. “parseLine” method categorizes each read line from sequential log file and prepares read line of string for further processing.

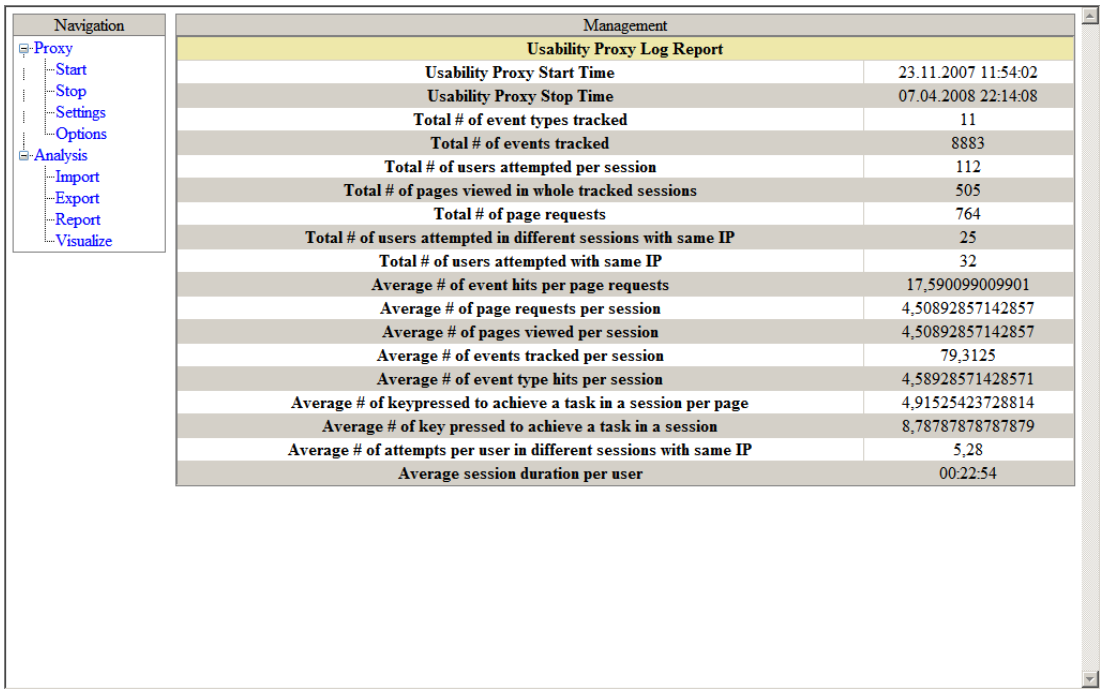
Each line read from log file by “readLine” method is an entry as explained before in database design section. So, each line is categorized first as “Event” or “HTTP” traffic. After, type of the entry has been detected further processing is applied on that line with “parseAttr” method as shown in Figure 5.5. “parseAttr” method parses all attributes of tables described in database design and tries to fill corresponding value of each attribute.

After “parseAttr” method extracted related data from log file it tries to match related data with corresponding attribute. As we described before we designed a special data type for easily managing this process and we used that data structure in “UPLParser” class. We can see from Figure 5.5 that “UPLParser” class includes some special properties called “domDetails”, “eventDetails”, “eventStream”, and “httpTraffic”. These properties are type of “UPLData” and as you can see they are named with related table names of designed database. “parseAttr” method inserts each extracted (attribute, value) pair into one of these related properties described as if its inserting attributes into database.

Finally, this parsing process is completed when each line entries of log file is processed by log parser as seen from Figure 5.6. The “UPLParser” class is a core parser processor, which means that it is not responsible for managing the whole parsing process. It just extracts (attribute, value) pairs from read line by “readLine” method. Management of parsing and insertion into database processes are completely handled under “Import” feature of web front-end as seen from Figure 5.6.

5.5 Reporting and Visualization Features

Reporting and visualization features of web front-end are designed as a final touch of integrated usability analysis tool. After, database and parser designs are completed we can start to concentrate on extracting meaningful information from database. Reporting part consists of a list of extracted total and average values from database. On the other hand visualization is composed of much more complex structure in order to make analysis process simple.



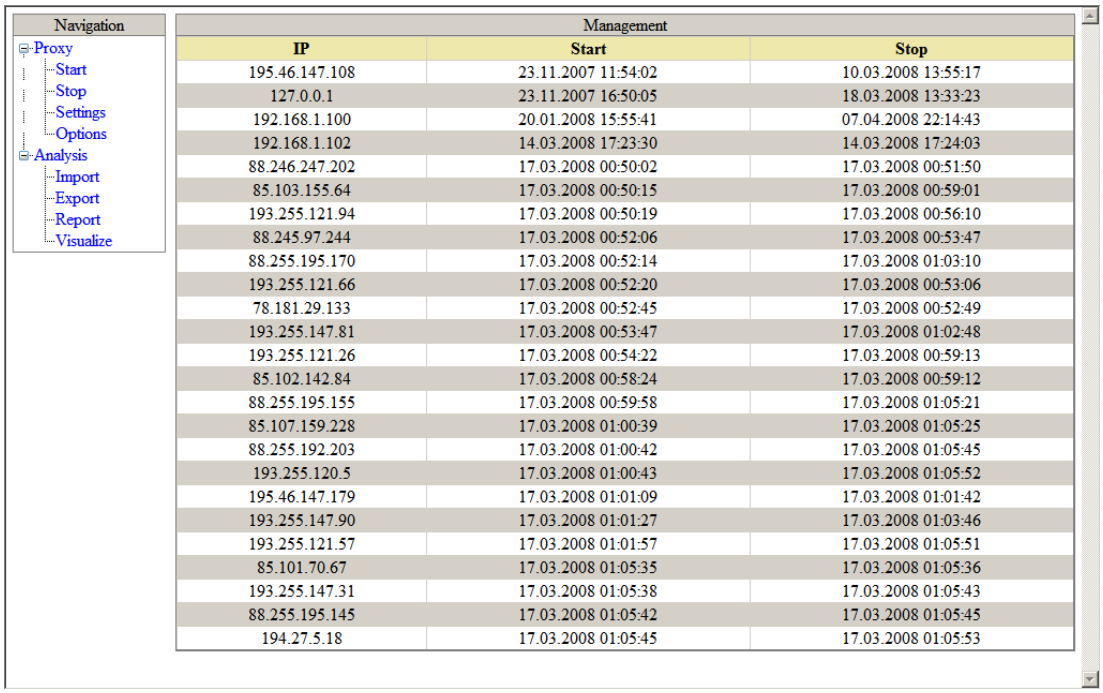
Usability Proxy Log Report	
Usability Proxy Start Time	23.11.2007 11:54:02
Usability Proxy Stop Time	07.04.2008 22:14:08
Total # of event types tracked	11
Total # of events tracked	8883
Total # of users attempted per session	112
Total # of pages viewed in whole tracked sessions	505
Total # of page requests	764
Total # of users attempted in different sessions with same IP	25
Total # of users attempted with same IP	32
Average # of event hits per page requests	17,590099009901
Average # of page requests per session	4,50892857142857
Average # of pages viewed per session	4,50892857142857
Average # of events tracked per session	79,3125
Average # of event type hits per session	4,58928571428571
Average # of keypressed to achieve a task in a session per page	4,91525423728814
Average # of key pressed to achieve a task in a session	8,78787878787879
Average # of attempts per user in different sessions with same IP	5,28
Average session duration per user	00:22:54

Figure 5.7 Usability Proxy Log Report

As you can see from Figure 5.6 reporting feature contains starting and stopping time of Usability Proxy with much more statistical data. For example, we can see total number of event types with total number of events collected or inserted into the database. Total number of users attempted in the analysis process with different measuring points such as per session, per different ip, or per session.

Also, page related statistics can be viewed through this report. For example, total number of page views or total number of page requests throughout the analysis. Another interesting statistics are average of session durations per user and average number of attempts of user in different sessions with same ip.

All those statistical results seen in Figure 5.7 are extracted from querying resulting database. Each line of statistics is generated with separate line of SQL queries. So, this report can easily be customized or extended. The listed ones in Figure 5.6 above are just designed as proof of concept but still meaningful and powerful result in terms of usability analysis.



IP	Start	Stop
195.46.147.108	23.11.2007 11:54:02	10.03.2008 13:55:17
127.0.0.1	23.11.2007 16:50:05	18.03.2008 13:33:23
192.168.1.100	20.01.2008 15:55:41	07.04.2008 22:14:43
192.168.1.102	14.03.2008 17:23:30	14.03.2008 17:24:03
88.246.247.202	17.03.2008 00:50:02	17.03.2008 00:51:50
85.103.155.64	17.03.2008 00:50:15	17.03.2008 00:59:01
193.255.121.94	17.03.2008 00:50:19	17.03.2008 00:56:10
88.245.97.244	17.03.2008 00:52:06	17.03.2008 00:53:47
88.255.195.170	17.03.2008 00:52:14	17.03.2008 01:03:10
193.255.121.66	17.03.2008 00:52:20	17.03.2008 00:53:06
78.181.29.133	17.03.2008 00:52:45	17.03.2008 00:52:49
193.255.147.81	17.03.2008 00:53:47	17.03.2008 01:02:48
193.255.121.26	17.03.2008 00:54:22	17.03.2008 00:59:13
85.102.142.84	17.03.2008 00:58:24	17.03.2008 00:59:12
88.255.195.155	17.03.2008 00:59:58	17.03.2008 01:05:21
85.107.159.228	17.03.2008 01:00:39	17.03.2008 01:05:25
88.255.192.203	17.03.2008 01:00:42	17.03.2008 01:05:45
193.255.120.5	17.03.2008 01:00:43	17.03.2008 01:05:52
195.46.147.179	17.03.2008 01:01:09	17.03.2008 01:01:42
193.255.147.90	17.03.2008 01:01:27	17.03.2008 01:03:46
193.255.121.57	17.03.2008 01:01:57	17.03.2008 01:05:51
85.101.70.67	17.03.2008 01:05:35	17.03.2008 01:05:36
193.255.147.31	17.03.2008 01:05:38	17.03.2008 01:05:43
88.255.195.145	17.03.2008 01:05:42	17.03.2008 01:05:45
194.27.5.18	17.03.2008 01:05:45	17.03.2008 01:05:53

Figure 5.8 Initial screen of visualization

As we said before visualizations are much more complex than simple reports. So, they need to be handled differently that statistical data extraction. The flow of visualization screen is dynamically changing based on the selections of analyzer. This means you have to make some choices before starting to analyze real visualization data of users.

As seen from Figure 5.8 its flow starts with revealing the users attempted into the analysis with their ip, start and stop time of respective sessions. Each row in this list of users is clickable onto it. Clicking on a row you can select that user.

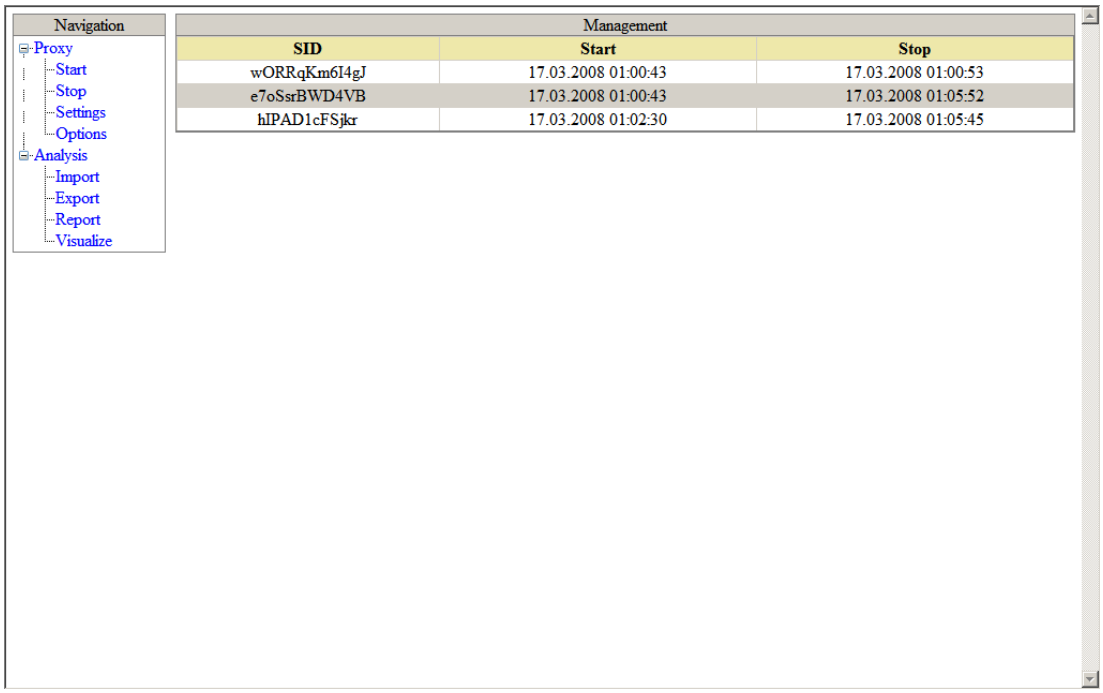
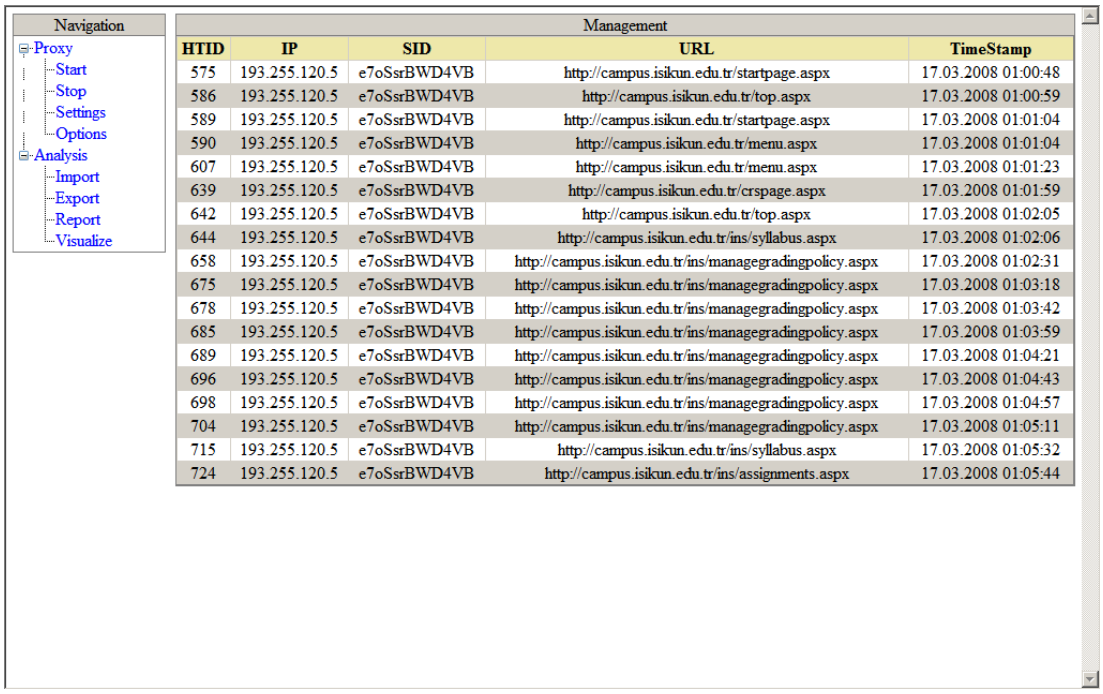


Figure 5.9 Second screen of visualization

After selection of user is done, the list is dynamically changed based on your selection as seen from Figure 5.9. Now, you can select the session you want to see its detail. As you can see there can be multiple sessions for same user but you can only view one session detail at a time. So, when you made another selection by clicking on a row the list will be dynamically change based on your selection.



HTID	IP	SID	URL	TimeStamp
575	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/startpage.aspx	17.03.2008 01:00:48
586	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/top.aspx	17.03.2008 01:00:59
589	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/startpage.aspx	17.03.2008 01:01:04
590	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/menu.aspx	17.03.2008 01:01:04
607	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/menu.aspx	17.03.2008 01:01:23
639	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/crspage.aspx	17.03.2008 01:01:59
642	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/top.aspx	17.03.2008 01:02:05
644	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/syllabus.aspx	17.03.2008 01:02:06
658	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:02:31
675	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:03:18
678	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:03:42
685	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:03:59
689	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:04:21
696	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:04:43
698	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:04:57
704	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/managegradingpolicy.aspx	17.03.2008 01:05:11
715	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/syllabus.aspx	17.03.2008 01:05:32
724	193.255.120.5	e7oSsrBWD4VB	http://campus.isikun.edu.tr/ins/assignments.aspx	17.03.2008 01:05:44

Figure 5.10 Third screen of visualization

As you can see from the Figure 5.10 above now you can view the path tracked by selected user in selected session. All the entries displayed in Figure 5.10 are visited pages by selected user in ascending time order.

This reveals the starting and ending time of user session with how user is interacted with system. Also, we can reveal how long it takes to jump another page or we can calculate session duration of user.

If this is not enough for you to analyze what is going on within the user session you can click one more time on a row to select a page. After, you clicked a row and selected a page with in this path; you will see that a download is initiated by browser.

As you can see from Figure 5.11 you can save or open automatically generated image. This image is visualized with querying database to extract related mouse movements and mouse clicks based on your selection. You can visualize each and every row in the list show in Figure 5.10 if you want.

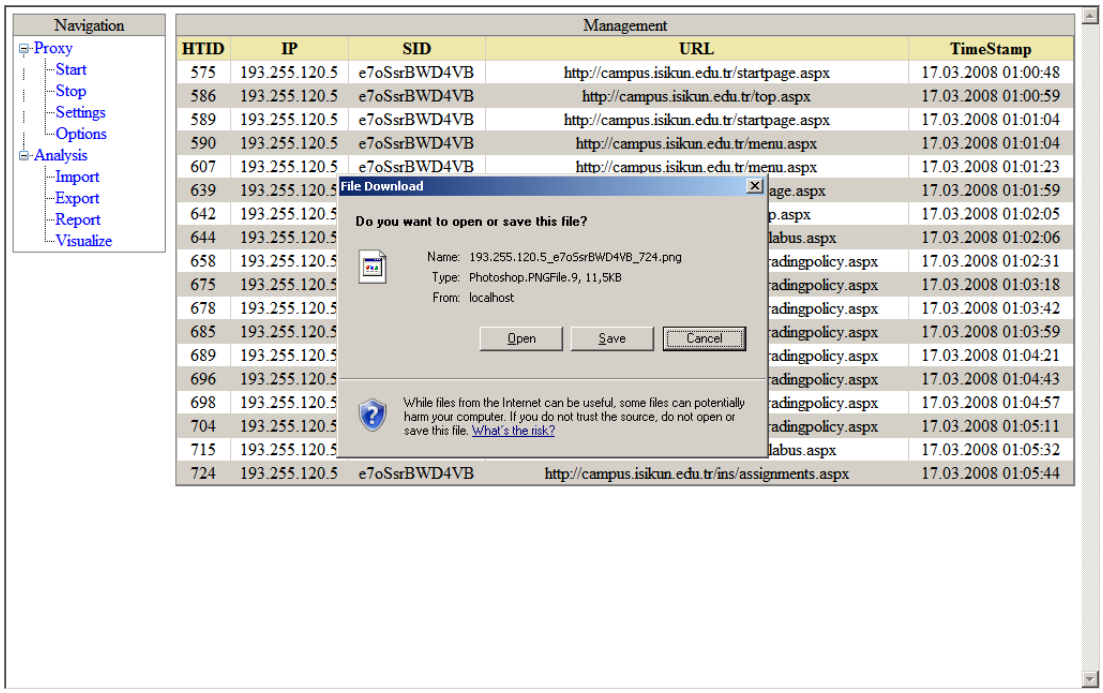


Figure 5.11 Final step of visualization

Figures 5.11 and 5.12 represent the final step of the visualization. If you want to analyze or visualize different user you have to click “Visualize” from the navigation tree on the left.

After you decided to view visualized image of user’s mouse movements with mouse clicks, you will see an image like in Figure 5.12. This image is generated automatically by passing extracted mouse coordinates from database into .NET framework drawing methods in order to visualize user’s interaction with selected page. The image in Figure 5.12 represents sample visualization and it contains some marks about how user is interacted with page.

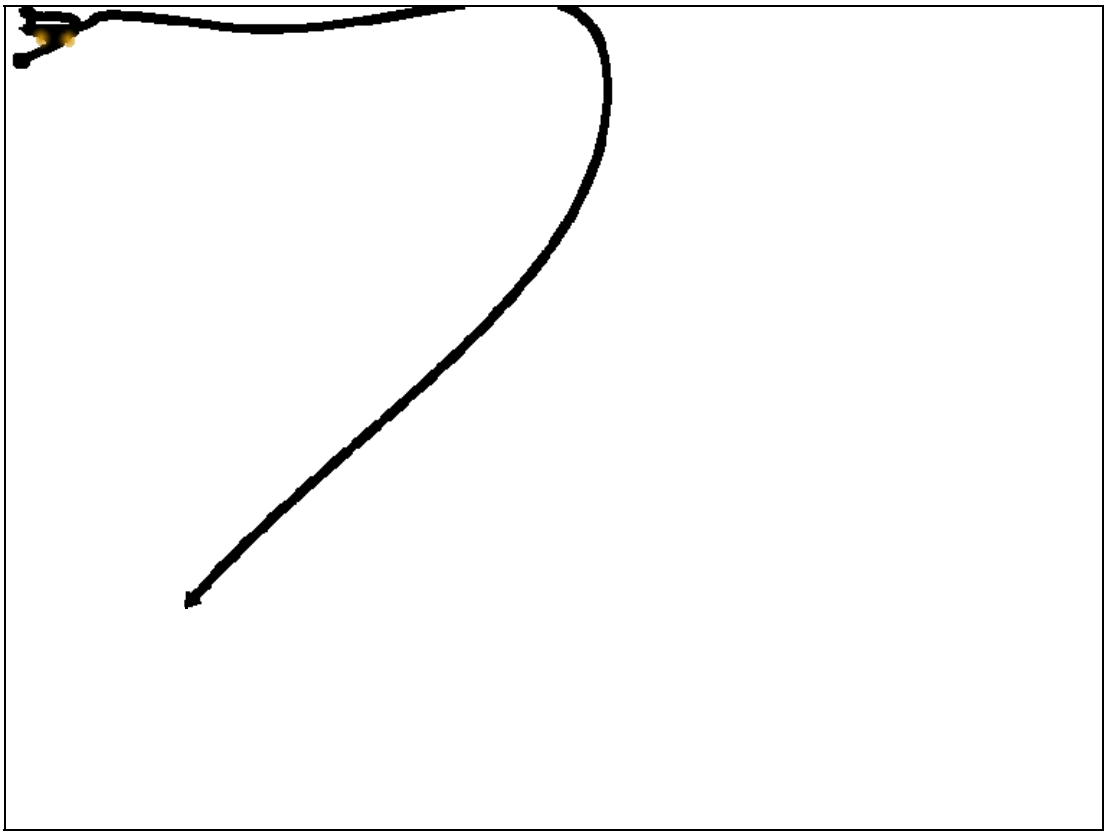


Figure 5.12 Visualization of mouse trails

For example, user is started to move mouse from the round black point at the upper left of the screen, then made some circular movements around there while user clicking on the spots visualized as orange semi-transparent points. Finally, user moved pointer from upper left corner to the middle of the screen and we can understand where user stopped by looking arrow head math at the end of the line.

Chapter 6
Conclusion and Recommendations for Future Work

The main goal of this thesis is to investigate suitable tools for making a usability analysis of an actively used system called Campus ONLINE and giving some insights about usage insufficiencies about system. So, we can reveal some design problems, or usability issues that has never been discovered until today, and propose a solution for fixing those issues.

Although we accomplished the tasks to reach our goal, we faced a number of difficulties. All difficulties tried to explain in details throughout thesis. As a result, when we asked ourselves the question why we cannot conduct the resulting usability analysis of Campus ONLINE, we can easily conclude that most of the effort is concentrated on developing integrated usability analysis tool. This fact prevented us to make real usability test and harvest resulting data in order to complete usability analysis of Campus ONLINE. For future work, my recommendation is to conduct usability tests with using this integrated analysis tool.

These usability tests can be designed by firstly determining focus groups and then selecting which part or functionality of the Campus ONLINE will be taken under usability testing. After all tasks are designed and focus groups are formed, which will be formed by selecting novice or expert users of the system, we can notify selected users about designated tasks before conducting test. Also before conducting test we can apply some of the surveys described throughout this thesis with collecting ip of participant computers. This way we can easily find and extract participant's usage statistics and we can also create user profiles easily from their data.

As summary of our research, we presented a set of suitable tools for our goal but finally decided to choose UsaProxy, which is an advanced, non-intrusive tracking solution for our system which does not require special setup at the client or server side. It constructs the backbone of our solution for usability analysis but it has lacked of some features as we described throughout thesis.

We aimed at those features lacked of UsaProxy and as a result we developed an integrated usability analysis tool that can be used repeatedly for analyzing actual usage of the Campus ONLINE. Even further development on this tool can be done. For example, visualization part of generating images from mouse activity can be enhanced by adding page statistics with semantic information into the resulting image. Maybe with further development replaying of user activities can be done available because UsaProxy has a suitable fundamental capability of replicating user movements when it is deployed in shared-browsing mode.

The UsaProxy performs detailed user activity tracking while modifying web pages “on the fly” before passing them on to the client browser. This ability of UsaProxy gives us a great flexibility to setup a test environment without disturbing users of the system. On the other hand, when we have been analyzed collected data after test deployments in development stage about actual usage of Campus ONLINE, in short test periods like, we revealed that SSL protection of Campus ONLINE affected the data collection of UsaProxy.

Besides all the other problems described above, we have faced another situation caused by our web server and hosting structure. In our web server, we have been hosting more than one intranet web sites for the public usage of University. This creates a point of confusion while deploying and interpreting collected data from UsaProxy log files. This confusion is partially overcome with developing domain independent log parser but when it combined with SSL problem, data collection is again under threat. In order to make usability testing safe and data collection flawless, UsaProxy should be deployed in front of the http server with disabling SSL protection or without disabling SSL we can only inform test participants about not to use SSL feature of Campus ONLINE.

References

[1] M. Guzdial, P. Santos, A. Badre, S. Hudson, M. Gray. "Analyzing and Visualizing Log Files: A Computational Science of Usability." s.l. : *HCI Consortium Workshop*, 1994.

[2] Nielsen, J. "Usability Engineering." San Francisco : Morgan Kaufmann, 1994. ISBN 0-12-518406-9.

[3] Nielsen, J. "Introduction to Usability." Alertbox. [Online] [Cited: 10 Jan 2008.] http://useit.com/alertbox.

[4] ISO9241. "Part 11: Guidance on Usability." s.l. : *International Standards Organization* , 1999. ISO-9241.

[5] Dix, A., Finlay, J., Abowd, G., & Beale, R. "Human-Computer Interaction." London : Prentice Hall Europe, 1998. 2nd Ed.

[6] Mack, R. L. and Nielsen, J. "Executive Summary, in Usability Inspection Methods." New York : John Wiley & Sons, 1994.

[7] Benbunan-Fich, R. "Using Protocol Analysis to Evaluate the Usability of a Commercial Website." *Information & Management*. 2, 2001, Vol. 39.

[8] Nielsen, J. "Usability AlertBox." [Online] [Cited: 10 Jan 2008.] http://www.useit.com.

[9] Nielsen, J. "Ten Years." *Alertbox.* [Online] [Cited: 10 Jan 2008.]

[10] Carter, J. "Usability Engineering Methodologies Research." *USER Lab.* [Online] [Cited: 08 Jan 2008.] http://www.cs.usask.ca/research/research_groups/userlab/res-UEM.html.

[11] Hartson, H.R., Castillo, J.c., Kelso J., Neale, W. C. "Remote Evaluation: The Network as an Extension of the Usability Laboratory." s.l. : *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1996.

[12] Bolt, N. "Guide to Remote Usability Testing." *ok-cancel*. [Online] 18th July 2006. http://www.ok-cancel.com/archives/article/2006/07/guide-to-remote-usability-testing.html.

[13] Bolt, N. "Types of Remote Usability Testing." Remote Online Usability Testing Wiki. [Online] 2006. http://remoteusability.com/.

[14] Garry, P. "Web-Based User Interface Evaluation with Questionnaires." [Online] [Cited: 10 Jan 2008.] http://oldwww.acm.org/perlman/question.html.

[15] Schneiderman, B. "Designing the User Interface". MA Reading : Addison-Wesley Publishing Co., 1998.

[16] NIST. "Technical Overview." *Web Metrics Testbed.* [Online] [Cited: 10 Jan 2008.] http://zing.ncsl.nist.gov/WebTools/tech.html.

[17] ClickDensity. [Online] http://www.clickdensity.com.

[18] TechSmith. "Screen Capture, Recording, and Presentation". [Online] [Cited: 10 Jan 2008.] http://www.techsmith.com.

[19] Kushniruk, A. W. and Ho. F. "The Virtual Usability Laboratory: Evaluating Web-based Health Systems." : Victoria B.C., AMIA Annu Symp Proc. 2006, 2004.

[20] Crazyegg. [Online] http://www.crazyegg.com.

[21] TapeFailure. [Online] http://www.tapefailure.com.

[22] RobotReplay. [Online] http://www.robotreplay.com.

[23] Jason I. Hong and Jeffrey Heer and Sarah Waterson and James A. Landay "WebQuilt: A Proxy-based Approach to Remote Web Usability Testing.", s.l. : *ACM Transactions on Information Systems*, July 2001, Vol. 19. Pages 263–285.

[24] Atterer, R., Wnuk, M., and Schmidt, A. "Knowing the User's Every Move: User Activity Tracking for Website Usability Evaluation and Implicit Interaction". Edinburgh Scotland : *WWW2006 15th*, 2006.

Appendix A: Sample Log Data Collected

193.255.147.86 2007-12-03,03:12:58 httptraffic
url=http://campus.isikun.edu.tr/assignments.aspx?crscode=MAN211 section=01
sd=0
193.255.147.86 2007-12-03,03:13:01 httptraffic
url=http://campus.isikun.edu.tr/crspage.aspx?sgrp=mycrs crscode=MAN211
section=01 sd=1
193.255.147.86 2008-00-03,03:12:58 sd=0 sid=8pU19m6VJjhB event=load
size=1021x356
193.255.147.86 2007-12-03,03:13:05 httptraffic
url=http://campus.isikun.edu.tr/crspage.aspx?sgrp=mycrs crscode=MAN211
section=01 sd=2
88.255.192.194 2007-12-03,03:13:06 httptraffic url=http://campus.isikun.edu.tr/
sd=3
193.255.147.86 2007-12-03,03:13:10 httptraffic
url=http://campus.isikun.edu.tr/crspage.aspx?sgrp=mycrs crscode=MAN211
section=01 sd=4
88.255.192.194 2007-12-03,03:13:12 httptraffic
url=http://campus.isikun.edu.tr/top.aspx sd=5
88.255.192.194 2007-12-03,03:13:12 httptraffic
url=http://campus.isikun.edu.tr/menu.aspx sd=6
193.255.147.86 2007-12-03,03:13:14 httptraffic
url=http://campus.isikun.edu.tr/webresources.aspx?crscode=MAN211 section=01
sd=7
193.255.147.86 2007-12-03,03:13:14 httptraffic
url=http://campus.isikun.edu.tr/top.aspx?crscode=MAN211 section=01 sd=8
193.255.147.86 2007-12-03,03:13:14 httptraffic
url=http://campus.isikun.edu.tr/syllabus.aspx?crscode=MAN211 section=01 sd=9
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=load
size=180x356
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mousemove
offset=28,1 dom=abafaaa
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mouseover
id=Form1 dom=aba
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mouseover
id=Table4 dom=abad
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mouseover
dom=abadadaa link=http://course.isikun.edu.tr/crspage.aspx?sgrp=opencrs
crscode=MAN211 section=01 text=Open%20Courses

193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mouseover id=Table4 dom=abad
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mouseover dom=abadacaa link=http://course.isikun.edu.tr/ExternalTools.aspx text=External%20Tools
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mousemove offset=60,0 dom=abadacaa link=http://course.isikun.edu.tr/ExternalTools.aspx text=External%20Tools
193.255.147.86 2008-00-03,03:13:10 sd=4 sid=8pU19m6VJjhB event=mouseover id=Table4 dom=abad
193.255.147.86 2008-00-03,03:13:11 sd=4 sid=8pU19m6VJjhB event=mousemove offset=65,114 dom=abadaba
193.255.147.86 2008-00-03,03:13:11 sd=4 sid=8pU19m6VJjhB event=mouseover dom=abadabah link=http://course.isikun.edu.tr/crspage.aspx?crscode=MATH231 section=02 text=%A0%A0%A0%A0%A0%20%20MATH231%2002
193.255.147.86 2008-00-03,03:13:11 sd=4 sid=8pU19m6VJjhB event=mouseover dom=abadabag link=http://course.isikun.edu.tr/crspage.aspx?crscode=MATH103 section=01 text=%A0%A0%A0%A0%A0%20%20MATH103%2001
193.255.147.86 2008-00-03,03:13:11 sd=4 sid=8pU19m6VJjhB event=mouseover dom=abadabae link=http://course.isikun.edu.tr/crspage.aspx?crscode=MAN211 section=01 text=%A0%A0%A0%A0%A0%20%20MAN211%2001
193.255.147.86 2008-00-03,03:13:11 sd=4 sid=8pU19m6VJjhB event=mouseover dom=abadabad link=http://course.isikun.edu.tr/crspage.aspx?crscode=HSS101 section=01 text=%A0%A0%A0%A0%A0%20%20HSS101%2001
193.255.147.86 2008-00-03,03:13:11 sd=4 sid=8pU19m6VJjhB event=mousemove offset=71,6 dom=abadabad
link=http://course.isikun.edu.tr/crspage.aspx?crscode=HSS101 section=01 text=%A0%A0%A0%A0%A0%20%20HSS101%2001
193.255.147.86 2008-00-03,03:13:11 sd=4 sid=8pU19m6VJjhB event=mousedown offset=69,3 dom=abadabad
link=http://course.isikun.edu.tr/crspage.aspx?crscode=HSS101 section=01 text=%A0%A0%A0%A0%A0%20%20HSS101%2001
88.255.192.194 2007-12-03,03:13:16 httptraffic url=http://campus.isikun.edu.tr/startpage.aspx sd=10
88.255.192.194 2007-12-03,03:13:16 httptraffic url=http://campus.isikun.edu.tr/bottom.htm sd=11
193.255.147.86 2008-00-03,03:13:14 sd=8 sid=8pU19m6VJjhB event=load size=1201x127
193.255.147.86 2008-00-03,03:13:14 sd=9 sid=8pU19m6VJjhB event=load size=1021x356
193.255.147.86 2007-12-03,03:13:19 httptraffic url=http://campus.isikun.edu.tr/crspage.aspx?crscode=HSS101 section=01 sd=12

Appendix B: UsaProxy Log File Analysis

UsaProxy is the fundamental tool selected in this thesis in order to make usability analysis. It basically captures user interaction patterns from a web page and the resulting interaction data is written to a log file. This section gives a more detailed overview of the different types of log entries explained by Atterer and developer team. It is based on the list of logged information that was presented in the previous section.

Transmission of text/html content to the client

Whenever a web page with text/html content passes through the proxy, the proxy records the entire HTTP request sent by the browser as well as the server response as stated by Atterer [24]. This is useful for later analysis of the content that was sent, especially in those cases where HTML was generated dynamically by the server. In addition to the standard log entry fields, ServerData ID is added to each http traffic recorded which uniquely identifies the files related to the request and response processed by UsaProxy.

Example: sd=15, repeatedly inserted after each HTTP Traffic entry in log

Loading and unloading of HTML pages

The client side JavaScript signals to the proxy when a page has been loaded completely and when a page is left for another page or the browser window is closed [24].

Example: load width=1394; height=777 unload

Resizing of the browser window

Whenever the user or JavaScript on a page resizes the page window, an appropriate log entry is generated [24].

Example: resize width=1050; height=555

Focus and Blur

Changes of the input focus the proxy records whenever the browser window gains focus (i.e. the user clicked inside it, used Alt-Tab to switch to it, or similar) or loses focus (e.g. because the user clicked inside another application's window, switched to another browser tab etc.): focus / blur [24].

Mouse Clicks

For mouse clicks, the tracking code takes note of the coordinates of the click, which is measured in pixels relative to the upper left corner of the HTML page [24]. If the target element can be identified via its name or id, that information is also written to the log according to Atterer [24]. If element is not identified with neither id nor name, the HTML tag of the element is recorded. Also, if the element is an anchor or surrounded by anchor tag, the target URL as well as the link text is logged:

Example:

click x=57;y=230 target=id:ds all
click x=144;y=210 target=name:x
click x=512;y=305 target=unknown:DIV
click x=1260;y=3420 target=unknown:scrollbar
click x=292;y=853 target=link:http://example.org+link text:Click here

Mouse Movements

All of the mouse movements generated by mouse pointer are written to the UsaProxy log file according to Atterer. However, if every JavaScript "onmousemove" event

were transmitted back to the proxy, this would take up too much bandwidth, so the coordinates are only logged with a rate of about one sample every 300 msec [24].

Example: mousemove x=913;y=674

Furthermore, mouseover entries provide additional information on the HTML element which the pointer currently hovers [24].

Example: mouseover x=57;y=230 target=id:ds all

Scrolling

Similar to mouse movements, scrolling coordinates are only recorded with reduced temporal resolution according to Atterer. The logged is based on the current vertical offset of the browser from the top of the document according to Atterer. For example, a y value of 42 means that the topmost 42 pixels of the page are not currently visible, and 0 means that the page is scrolled to the top [24].

Example: scrolledTo y=42

Key presses

According to Atterer if the user presses a key, the pressed key is logged in the form shown below. Key presses are usually also noticed even if the user invoked a keyboard shortcut of the browser, such as pressing Ctrl-F to start searching for text inside the page [24].

Example: keypress key = u

Appendix C: CD Containing Measurement Data and Results

All collected data and developed tools with deployment instructions are included in CD

Curriculum Vitae