A Generic Usability Test Environment for Web-based Services

J. Lindenberg & M.A. Neerincx
TNO Human Factors Research Institute
P.O. Box 23, 3769 ZG Soesterberg
lindenberg@tm.tno.nl, neerincx@tm.tno.nl

Summary

The success of web-based services depends critically on the users' satisfaction and on the effectiveness and efficiency of their computer use. The development of a tool that supports systematic user tests with end-users interacting with (a prototype of) a future interface of a web-based service will be discussed. The full scope of relevant user and task characteristics can be addressed and fast results can be provided with this generic usability test environment called TIATO ("TNO Internet Applicatie Test Omgeving"). It can be used to set up usability tests, to conduct local and remote tests and to pre-process the data generated during the usability test. TIATO is currently being used for the evaluation of web-based services. The tool will be demonstrated and the experiences will be discussed at the conference.

1. Introduction

The number and diversity of web-based services as well as the number of users are growing enormously. Due to usability problems however, the utility of these services is far from optimal. The cost of laborious interaction may for example exceed the profit of the service. Therefore, the U-WISH (Usability of Web-based Information Services for Hypermedia) project aims at a user-centred design method for web-based services that guides the iterative process of software development. Several partners of the Dutch Telematics Institute participate in this project: the research institutes TNO-HFRI, CTIT and CWI, and the companies KPN, Océ, Cap Gemini and Rabofacet. In the current project phase, guidelines, techniques and tools for the specification and assessment of web-based services are being developed. In the next phase, these usability-engineering elements will be integrated into a coherent design method for web-interfaces and subsequently applied.

In the field of human-computer interaction, several techniques and methods have been developed for usability assessments that take place at different stages of software development (see for example Karat (1997) for a brief general overview, Nielsen & Mack (1994) for an overview of inspection methods and Gray & Salzman (1998) for a review of experiments that compare such methods). Systematic tests with end-users who perform actions with (a prototype of) the future interface provide the most complete results. Such user tests should address the full scope of relevant user and task characteristics, and should provide fast results that guide the user interface design process (Neerincx, 1998). This paper presents a software tool that enables effective and efficient user tests: TIATO, a generic usability test environment for web-based services. TIATO is developed in co-operation with General Design, a Dutch company that is specialised in the design and implementation of web-based services.

2. Usability Testing

The Final Draft International Standard (FDIS) of the International Organisation for Standardisation (ISO), ISO/FDIS 13407 "Human-centred design processes for interactive systems", provides a relative high level of guidance of the design and evaluation of human-machine systems. This standard refers to ISO 9241-11 "Guidance on usability specification and measures" that defines usability as the effectiveness, efficiency and satisfaction with which specific users achieve specified goals in particular environments. Effectiveness is the accuracy and completeness with which specific users can achieve specified goals in particular environments. Efficiency refers to the resources expended in relation to the accuracy and completeness of the goals achieved. Satisfaction comprises the comfort and acceptability of the work system (e.g. application) to its users and other people affected by its use. The actual outcome of the three types of usability measures can change over time due to increasing experience of the user. A usability assessment should address these changes, for example with performance and knowledge tests that measure learning effects (Neerincx & de Greef, 1993).

Table I shows the three types of measures that should be captured by a usability test for web-based services. The measurements that indicate effectiveness are accuracy (e.g. errors indicate inaccuracies) and completeness (e.g. the percentage of successful task performances of the total set of tasks). The measurements used for efficiency are effort (e.g. perceived mental load, eye movements) and duration. The satisfaction of the user is measured by the experienced ease-of-use, the attractiveness of the service and the feeling of trust that a user experiences while using the service.

Table I: Three types of usability measures.

Effectiveness	Efficiency	Satisfaction
accuracy	effort	ease-of-use
completeness	duration	attractiveness
		trust

3. TIATO Requirements

The requirements for TIATO are based on the current usability testing practice but anticipates possible future needs (Nielsen, 1997). The requirements for TIATO are to:

- Provide introduction, instruction and questionnaires to the users.
- Provide tasks to the users and detect task completion.
- Provide means to construct test material.
- Provide several users local and remote access to web-based services.
- Log user behaviour and responses.
- Pre-process the gathered raw data (prior to further data analysis with a statistical software package)

In the following sections, the various requirements for the test environment will be elaborated.

General set-up. A usability engineer should be able to set up an evaluation using the test environment. That is, he or she should be able to define tasks including a start state and an end condition. Questions that are presented after task completion should also be defined at this stage. The development of a new test or the adaptation of existing ones should be feasible for someone without a specific technical background.

Task presentation. A usability test consists of a number of tasks which users have to perform. Web-based tasks have to be carried out with a conventional browser (i.e. Netscape Communicator or Microsoft Internet Explorer). The test environment should start with a presentation of the first task and subsequently present the service that is needed to perform the task in a specific predefined starting state (e.g. a specific web-page). The user carries out the task and the test environment monitors whether or not the predefined stop condition has been reached. The stop condition can be a target page, a maximum time or a user-indicated stop. When the stop condition is detected, one or more questions can be presented. The test environment presents the next task when all questions are answered by the user. After completion of all tasks and questions the test is finished.

Logging. During task performance, the test environment logs the behaviour of the users and their answers to the questions presented after each task. The measures of Table I can be used to deduce the effectiveness, efficiency and satisfaction of the use of a web-based service. The measures can be derived from the indicators given in Table II. It is important to use multiple indicators: one data type complements and may explain results of the other data type (e.g. requested pages are required for the interpretation of eye-track data, and eyetrack data can be used to determine which link was used to request a certain page). Whereas most indicators of Table II have been used in previous evaluations eye-tracking is a rather new technique for usability evaluation. A number of interesting eye-track measures (e.g. scanpath length, number and duration of fixations, convex hull area and spatial density) are suggested in the literature based on readability (Kolers et al., 1981) and usability (Kotval & Goldberg, 1998; Yamamoto and Kuto, 1992) research.

Table II: Indicators of the usability measures of Table I

	requested pages	**mouse clicks**	**keyboard input**	**time**	**eyetrack data**	**answers**
accuracy	◆					◆
completeness	◆					◆
effort	◆	◆	◆	◆	◆	◆
duration				◆		◆
ease-of-use						◆
attractiveness						◆
trust						◆

Performance constraints. Of course, the test environment has to fulfil certain performance constraints. The test environment should not noticeably delay the web-service access speed or change the behaviour of the web-service in any other way. It should also be possible to conduct usability tests with several users at the same time.

Processing data. It is to be expected that the usability tests will generate an enormous amount of data. To manage the amount of data, software should be developed to pre-process the data. For further analysis with statistical software. It should however also be possible to use the raw data.

Remote evaluation. Evaluations can take place in a usability lab with the web-based services installed locally. The test environment should however also be able to deal with evaluations of remote sites, possibly with remote users.

Other requirements. The tool should be compatible with contemporary and future browsers of various manufacturers, the web-service that will be tested should remain untouched and it should be easy to add future extensions to the test environment.

4. TIATO Design

The first objective was to use TIATO in a usability lab. The usability lab consists of an observation room with observation equipment and a test room with test-pc's. During the design stage a number of technical options were considered (adapting the web-server, adapting the web-based service, adapting a web-browser or creating a proxy server). The proxy server[1] proved to be the best solution because all requirements could be fulfilled without introducing significant restrictions.

The users' browsing behaviour (e.g. *requested pages*) is logged by the proxy by monitoring the generated web-traffic between the pc of the user and the web-based service. The *eye-track data* can be read from the communication port of the test pc. The *mouse clicks* can be obtained from the operating system as well as the *keyboard input*. The eyetrack, mouse and keyboard data are all integrated with the browsing data by the proxy and logged into a database. In addition, the proxy can provide the test environment itself as a web-based service. This service is able to present tasks, provide access to web-services, check for a stop condition and show questions. The test environment service also provides the usability engineer the means for setting-up a usability test. The task and question definitions for the test are maintained in the database. The proxy retrieves this information when an experiment begins. Pre-

[1] A proxy server is an intermediary between a web-server and a client pc.

processing the logged data can also be performed with TIATO. The design supports the testing of remotely installed web-based services and the use of remote participants. Figure 1 shows the architecture of TIATO.

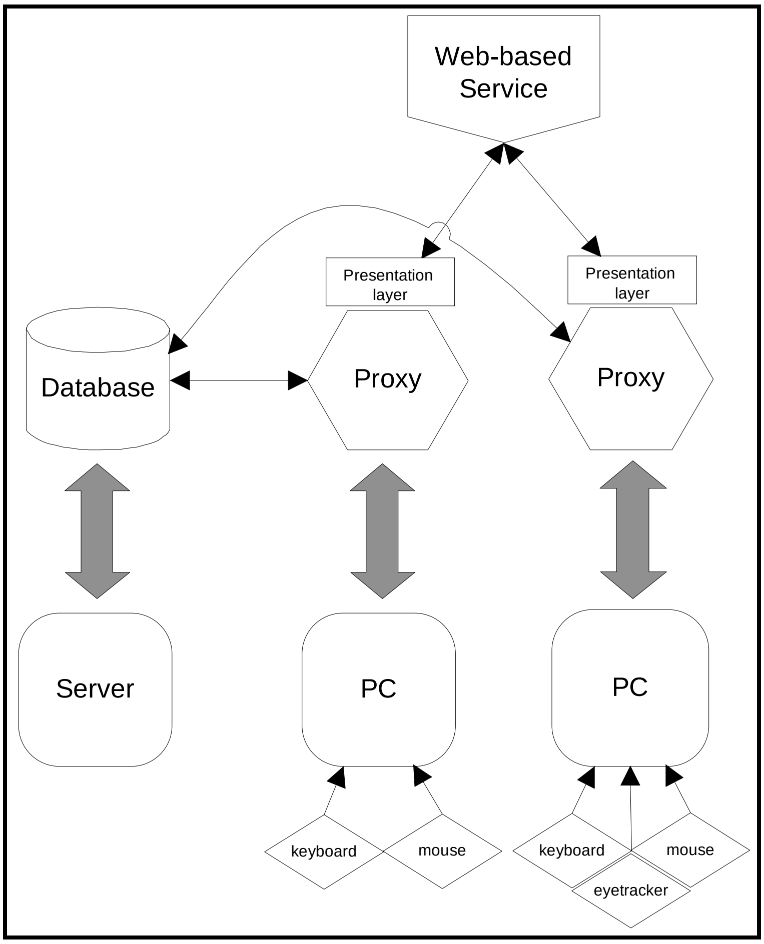

Figure 1: The TIATO architecture in the case of two test pc's

5. TIATO Implementation

The test environment was implemented using various web-techniques (such as JavaScript, HTML, Server Side Scripting). The Java proxy server "Jetty" from Mort Bay Consulting was adapted for TIATO. The part of the program that reads the eyetrack data from the COM port is written in C++. The program that obtains the mouse clicks from the operating systems is also written in C++. The database is an Oracle 8 Enterprise Edition Database. The proxies uses the Oracle JDBC driver[2] to obtain the test definition from the database. The Oracle JDBC driver is also used to log all the data generated by the users into the database.

In the current implementation, the experimenter can set up an evaluation using the web-based service interface provided by the environment. First, a new test must be set up. Second, the tasks, questions and settings have to be defined for that experiment using the web-based forms provided by TIATO. When the definition is finalised, the experiment can begin. The user is shown a maximised browser window with a *Start Test* button. When this button is pressed the first task is presented along with a *start task* button. When the user presses the button, the web-based service is presented in the pre-defined start state. The user must carry out the task within the web-based service. The test environment will detect the predefined end-state that marks the end of the task. The questions accompanying that task are presented next and when these are answered by the user, the next task is presented. The cycle is repeated

[2] A database access tool.

until the final task is carried out and the final questions are answered. The browser will then show a page containing the words *End of Test*.

When all users are finished the data from the database can be pre-processed using our dedicated stored procedures from the test environment itself. The raw data can also be read directly into statistics software using ODBC.

6. Conclusions and Discussion

Whether TIATO can be used for effective, efficient and complete usability testing of local and remote web-based services with local and remote users will be tested in the oncoming experiments. These experiments will comprise of a variation of tasks that have to be performed by the users with the three available web-based services. The first service is a TNO-HFRI web-site as a standard example of a company web-site. The second service is a theatre-booking environment, which uses virtual reality and natural language techniques. A web-based public counter named OLE2000 is the third service. The experiments will evaluate general interface adaptations that are developed to increase the effectiveness, efficiency, and satisfaction of the use of web-based services, i.e. adaptations to user and task characteristics that are crucial for the usability of web-based services (Chen & Rada, 1996; Czaja, 1997; Höök et al. 1996). For the user characteristics the focus is on spatial and learning ability, for the task characteristics the focus is on the various types of tasks (open, closed, goal driven and situation driven).

It is difficult to evaluate web-based services containing Java applets with TIATO. TIATO provides a solution to annotate the Java code but this requires direct access to the code and a considerable programming effort. Another solution is the logging of user behaviour by a human observer using video analysis software (Observer of Noldus).

Systematic tests with end-users who perform actions with (a prototype of) the future interface provide the most complete and clear results. When TIATO is able to cope with the set of task characteristics of the proposed experiments, it proves to be a valuable tool for systematic usability testing. Our experiences with TIATO in practice will be reported at the conference.

Acknowledgement

General Design, a software company in Amsterdam The Netherlands, designed and implemented TIATO and we would like to thank the main contributors Dirk Soede and Peter Boersma.

References

Chen, C. & Rada, R. (1996). Interacting with hypertext: a meta-analysis of experimental studies. *Human-Computer Interaction*, 11, 125-156.

Czaja, S.J. (1997). Computer technology and the older adult. In: Helander, M.G., Landauer, T.K. & Prabhu, P.V. (eds.), *Handbook of Human-Computer Interaction*, 2nd edition, Chapter 34. Amsterdam: Elsevier North-Holland.

Höök, K., Sjölinder, M. & Dahlbäck, N. (1996). Individual Differences and Navigation in Hypermedia. *SICS Research Report*, R96:01, SICS, Sweden 1996.

Gray, W.D. & Salzman, M.C. (1998). Damaged Merchandise? A Review of Experiments That Compare Usability Evaluation Methods. *Human Computer Interaction*, 13, 203-261

Karat, J. (1997). User-Centered Software Evaluation Methodologies. In Helander, M.G., Landauer, T.K. & Prabhu, P.V. (Eds.) *Handbook of Human-Computer Interaction*, 689-704. Amsterdam: Elsevier Science.

Kolers, P.A., Duchnicky, R.L. & Ferguson, D.C. (1981), Eye movement measurement of readability of CRT displays. *Human Factors*, 23(5): 517-527

Kotval, X.P. & Goldberg, J.H. (1998). Eye movements and interface component grouping: an evaluation method. *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting 1998.*

Nielsen, J. & Mack, R.L. (1994). Usability Inspection Methods. John Wiley & Sons, Inc, New York

Nielsen, J. (1997). Usability Testing. In Salvendy, G. (Ed.) *Handbook of Human Factors and Ergonomics*, 1543-1568. New York: John Wiley & Sons.

Neerincx, M.A. & de Greef, H.P. (1993). How to aid non-experts. In Ashlund, S., Mullet, K., Henderson, A., Hollnagel, E., White, T., editors, *Proceedings of INTERCHI'93*, pages 165-171, New York. ACM.

Neerincx, M.A., (1998). Design for All: Human Factors of Special Needs in Software Development. *TNO-report*, TM-98-C068, TNO-HFRI, Soesterberg, The Netherlands

Yamamoto, S. & Kuto, Y. (1992). A method of evaluating VDT screen layout by eye movement analysis, *Ergonomics*, 35(5/6): 591-606