Analytical Evaluation of Interactive Systems regarding the Ease of Use*

Nico Hamacher[1], Jörg Marrenbach[2]

[1]EMail: hamacher@techinfo.rwth-aachen.de
[2]EMail: marrenbach@techinfo.rwth-aachen.de

Department of Technical Computer Science
Aachen University of Technology, Germany
Ahornstr. 55, D-52074 Aachen
Tel.: +49-241-44026105
Fax: +49-241-8888308

Abstract

This paper describes a new tool for the generation and analysis of normative user models based on the GOMS theory for the evaluation of interactive systems and the analysis of the usability. Furthermore, a comparison of user models and generated action protocols is facilitated. Additionally, design alternatives can be compared. The results of the analysis are visualised in various ways.
The introduced tool supports the system engineer in a considerable way to evaluate interactive systems and produces suitable analysis data as a base for decisions while the systems are developed.

1. Introduction

Today, usability engineering of interactive systems is paying increasing attention. In general usability is determined by testing prototypes. The disadvantage of this approach is that these tests can be performed only in late stages of the development process. An early analysis of usability would be a significant advantage in regard of saving time and resources. In this paper the tool TREVIS (**T**ool for Rapid **Ev**aluation of **I**nteractive **S**ystems) is presented. TREVIS enables the design engineer to model the behaviour of an user while interacting with a device and derive usability measures from this simulation

2. Usability Evaluation

To analyse the usability of interactive systems an empirical evaluation is commonly used. This type of evaluation requires a prototype and a couple of qualified testing subjects. Mostly, this procedure is very expensive and time-consuming. Furthermore, this evaluation is feasible only in late stages of the development process when a prototype is available (fig. 1), so that the results of this evaluation often cannot be used for a redesign in a sufficient way. Even though if this empirical evaluation cannot be omitted, because of the generation of plenty of useful information for improving the usability, an earlier evaluation would be very helpful for the design and specification of the system.

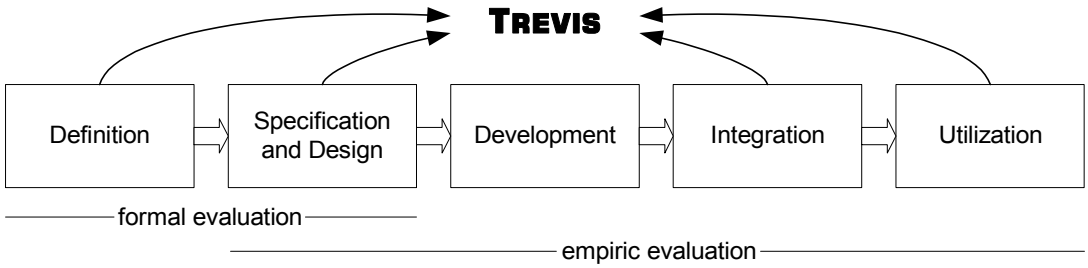

Figure 1: The different phases of a development process.

* In: Stephanidis C. (Eds.): Proceedings of HCI International 2001. Universal Access in HCI. Towards an Information Society for All , Volume 3, pp. 585-589, August 5-10, New Orleans, Lawrence Erlbaum Associates, ISBN 0-8058-3609-8

To analyse the usability in an early stage of the development process a formal evaluation is needed. One method of formal evaluation is developed by Card, Moran and Newell for modelling interactions between an user and an interactive system. It is called GOMS, which is an abbreviation for the components of the model: Goals, Operators, Methods and Selection rules. Over the years it has been shown that GOMS is able to sufficiently describe the interactions of an operator. Its simple and plain structure makes this method easy to understand particularly for development engineers which do not have a psychological background.

A GOMS model is also called user model. So far some GOMS variants have been introduced. The dialect with the most extensive analysis results is NGOMSL (Natural GOMS Language), first introduced by Kieras in 1988. An analysis based on NGOMSL generates qualitative as well as quantitative predictions, like execution and learning time. The execution time describes the time to reach the goal whereas the learning time specifies how much time an operator needs to learn the whole task.
Although the use of GOMS models is not very complicated, it is very tedious to build these models manually. Hence, a tool is needed which integrates the GOMS theory into the development process and supports efficiently creation and analysis of user models.

3. System Architecture

For the evaluation of usability three criteria are considered essentially. According to ISO 9241 part 11 these are *effectiveness*, *efficiency* and *satisfaction*. User performance is measured by the extend to which the intended goals or subgoals of use are archived (*effectiveness*) and the resources such as time, money or mental effort that have to be expended to archive the intended goals (*efficiency*). *Satisfaction* is measured by the extend to which the user finds the use of the system acceptable. Regarding these criteria the tool TREVIS (**T**ool for **R**apid **Ev**aluation of **I**nteractive **S**ystems) is described that supports the synthesis and analysis of user models based on NGOMSL.
TREVIS includes four main modules: the user model editor, the device model converter, the handbook generator, and the analysis module. The system architecture is depicted in figure 2.

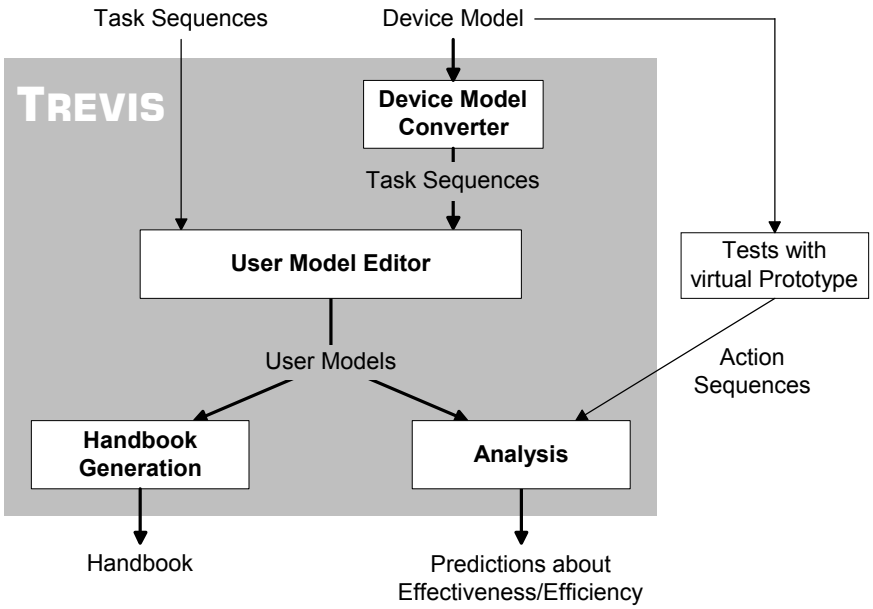

Figure 2: Architecture of TREVIS

Based on the task sequences as one result of the requirement analysis the user models can be created manually in the user model editor. The tool supports this process e.g. by offering a graphical editor and a library for reusing components. Moreover, user models can be stored in projects, where the project represents the interactive system and the user models describe the tasks which have to be done.
In contrast to other GOMS-editors which only allow to edit the user models manually, TREVIS is able to generate user models semiautomatically. A device model contains details about the inner work of the device. Using the device model converter the task sequences can be generated semi automatically. The user models can also be created from these sequences.
In the analysis module the following four different analysis methods are included, which depend on the development phase, in which TREVIS will be used in:

- The user model analysis generates qualitative as well as quantitative predictions, like execution and learning time (as already described with NGOMSL).
- A comparison between different user models is implemented in the design analysis module, which can be used as a basis for design decisions. Although this comparison presentation is a helpful functionality, no other tool includes it.
- In the action sequence analysis, action sequences resulting from testing a prototype can be imported and analysed. A grouping of different sequences is possible, e.g. to perform an analysis of significance. With this feature, TREVIS is also applicable in late stages of a development process as depicted in figure 1.
- A fourth method analyses these action sequences in comparison with the user models. This analysis shows the differences between the actions specified in the user models and the activities, the users performed while interacting with a prototype. Based on this analysis, predictions about the effectiveness and efficiency can be made.

Figure 3 shows the user interface of the tool with the project management area on the left and the editing area on the right side. In the middle the different user models and action logs are shown.

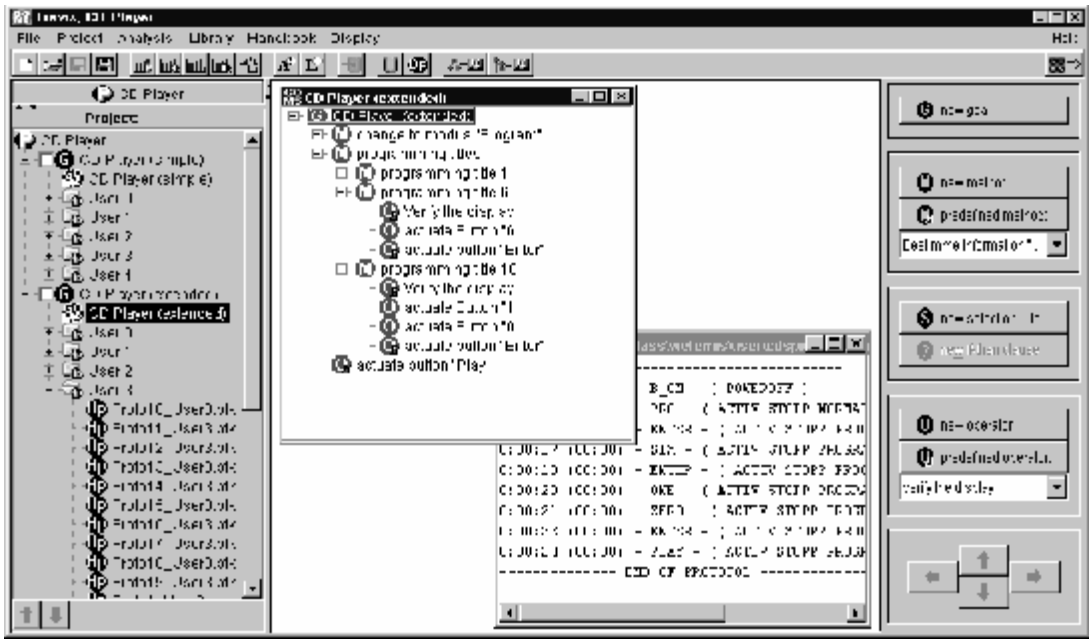

Figure 3: The user interface of TREVIS with an exemplary project, user model and action log.

With the possibility to analyse both user models and action sequences, TREVIS is the first tool, which can be utilised in early phases as well as in late phases of the development process. Beside this, a comparison of user models with action sequences has never been implemented before.

The user models contain the complete description of the procedural knowledge, which the user has to know in order to perform tasks using the device. Hence, Elkerton suggests to build a handbook by following and describing the tasks, a user has to do. Regarding this proposition, a handbook based on the user models can be created by the handbook generator.

4. Application and Evaluation Results

A CD player prototype was created to evaluate the tool features. The user interface of the two players differs primarily in the number of buttons. Prototype 1 (fig. 4, left) implements a simple CD player only with the most important buttons. The second prototype (fig. 4, right) includes additionally a number field to select the tracks in a direct way.

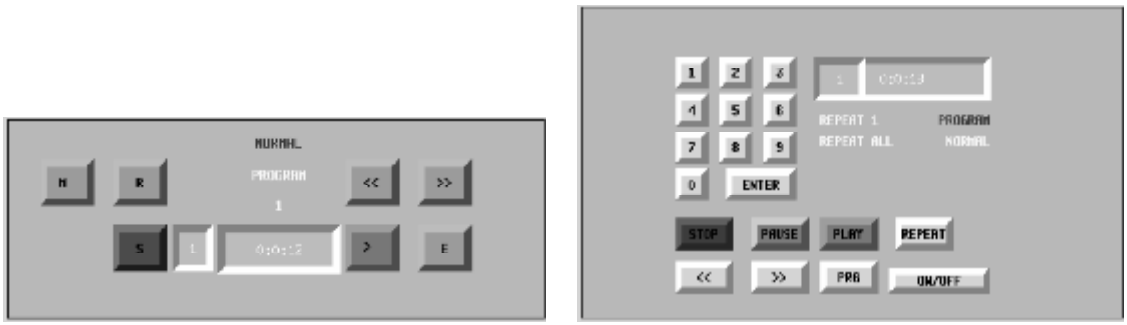

Figure 4: User interface of a simple CD player (left: prototype 1) and extended CD player (right: prototype 2)

Therefore, the user models for two different types of CD players were modelled. The modelled task is "to programm a couple of songs and to start this program".

The user models were analysed by the user model analysis and the design analysis. Further on, several people interacted with the prototypes and action logs were recorded, which were also analysed. Finally, for each CD player a comparison between the corresponding user model and the action logs was carried out. Two of the resulting diagrams are depicted in figure 5. The execution time calculated from the user models is shown as a line where the crosses specifying the execution time of the action logs. It can be seen, that most of the empirical times only vary up to 10% (grey region). Wrong actions, done by the users, were counted and diagrammed as columns.

5. Summary

In this paper the tool TREVIS is introduced, which enables the development engineer to evaluate interactive systems during the whole development process. Thereto it offers a method for the analytical evaluation by using GOMS models. With these user models interactions between an operating user and an interactive system can be described. A manual editing is implemented as well as a semiautomatical generation from formal specifications. Furthermore action sequences resulting from testing a prototype can be imported. With these data various analysis can be performed. Beside the user model analysis, a comparison between different user models and a comparison between user models and associated action sequences.
For the evaluation of the tool CD player prototypes were created and an exemplary testing is performed.
The results indicate, that TREVIS is a useful tool to support the development engineer during the development process.

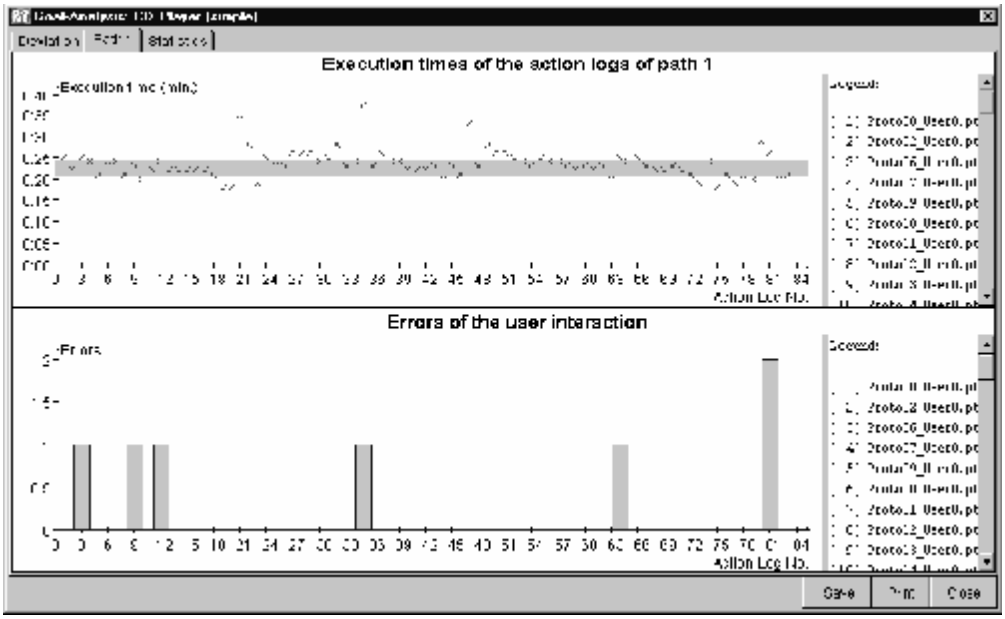

Figure 5: Comparison of the analytical calculated and rised execution time (prototype 1)

6. References

Card, S., Moran, T. and Newell, A. (1983): *The psychology of human computer interaction*. Lawrence Erlbaum.

Elkerton, J. (1988): Online Aiding for Human-Computer-Interfaces. In: M. Helander (ed.), *Handbook of human-computer interaction*. , pp. 345-362.Amsterdam: North Holland.

Hamacher, N. (2000): Entwicklung und Implementierung eines Werkzeugs zur Bewertung interaktiver Systeme basierend auf normativen Benutzermodellen. Department of Technical Computer Science, Diploma Theses, Aachen University of Technology.

ISO 9241-10 (1996): Ergonomic requirements for office work with visual display terminals - Dialogue principles. International Organisation for Standardisation, Genf.

ISO 9241-11 (1998): Ergonomic requirements for office work with visual display terminals - Guidance on usability. International Organization for Standardisation, Genf.

Kieras, D. (1988): Towards a practical GOMS model technology for user interface design. M. Helander (ed.), *Handbook of human-computer interaction*, pp. 135-157. Amsterdam: North Holland.

Leuker S. and Marrenbach J. (1998): Integrating User Models into Device Prototype Design. *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society IECON, Volume 4/4*, pp. 2532-2534, August 31 - September 4, Aachen.

Marrenbach, J. (1999): Rapid Development and Evaluation of Interactive Systems. *Proceedings of the 5th ERCIM Workshop User Interfaces for All , Volume Report 74*, pp. 81-86, Dagstuhl.

Marrenbach J. (1999): Konzept eines Werkzeugs zur formalen und empirischen Evaluierung der Gebrauchstauglichkeit interaktiver Endgeräte. Gärtner K.-P (Eds.): *Ergonomische Gestaltungswerkzeuge in der Fahrzeug- und Prozeßführung*, pp. 217-230, Stuttgart