AMME: an "Automatic Mental Model Evaluator" to measure complexity of user behaviour recorded on logfiles.

Matthias Rauterberg
Work and Organizational Psychology Unit, Swiss Federal Institut of Technology (ETH), Nelkenstr. II. CH-8092 Zurich, Switzerland; (00)-41-1-254-7082; email: rauterberg@ezrzI.vmsmail.ethz.ch

1. Aim:

This stuy was carried out to support the "formal analysts" in studying user keystroke behaviour. The nonnal design cycle to construct a fonDal model is a top down approach. In this study we present an automatic bottom up approach to construct a fonDal description of user behaviour. The formalism we select is the Petri- net, to analyze the users' performance knowledge with finite place/transition-systems.

2. State of the art:

There are different formalisms for constructing user models; TAG, ETAG, CLG, GOMS, CCT, the different kinds of grammars (BNF, EBNF, etc,), and state transition nets. Using any of these formalisms the analyst must always design pure (="error free") user model in a top down approach. Then he can try to prove his model with the more or less "error free" empirical data. This is difficult and insufficient, and one of the consequences is that most of the formal models exist only as paper versions and have not been implemented as computer programs.

3. Bottom up approach:

If there is a possibility to construct user models in an automatic, bottom up ap- proach, then the handling with fonDal models becomes easy, A sequence of keystrokes, mouse clicks, etc. can be contemplated as a sentence derived from a defined grammar or as a process derived from a Petri-net. A state transition net, as a complete description of the software the user is interacting with, can be used to identify the equal states in the keystroke sequence. All parts of the whole keystroke sequence between two dialog states are elementary processes. All elementary processes can be combined to fold a Petri-net ("folding" operator; Oberquelle et aI. 1983). The "folded" Petri-net is a formal description ("model") of the procedural knowledge of the users behaviour .

4. Why Petri-nets ?:

Bauman and Turano (1986) showed, that Petri nets (Petri 1980) are equivalent to formalism based on production rules (like CCT of Kieras and PoIson 1985). To model user knowledge we use finite place/transition nets with marked tokens, Petri-nets have the following useful features: modelling of parallel actions, a clear semantic, powerful enough to handle with context sensitivity and, the possibility to embed sub- nets. We are using the Petri-net tool PACE, so we are able to model time aspects, PACE offers the possibility of simulation, so we can analyze our user model in a dynamic fashion.

5. Diagnostic features of a user model developed with AMME:

One of the most interesting aspects of the Petri-nets constructed with AMME is the possibility to measure the behavioral and cognitive complexity in a simple fashion (McCabe 1976, Kornwachs 1987). Now each analyst is able to investigate the learning process of a user in handling an interactive software. The possibility to detect an interactive deadlock is an- other important feature. Examples will be presented. Analyzing logfiles in an automatic way enable the investi- gator to calculate applied statistics. Based on the data of an empirical investigation (Rauterberg 1992) we can show that the observed "behavioural complexitY" correlates negatively with the "cognitive complexity".

6. References:

Baumann, R. & Turano T.A. (1986): Production based language simulation of Petri nets, *Simulation,* 47, 191-198.

Kieras D.E. & Poison P .G. 1985, An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies,* 22, 365-394.

Kornwachs, K. (1987): A quantitative measure for the complexity of man-machine interaction process. in: Bullinger, H-J. & Shackel, B., eds. 'Human-Computer-lnteracfion INTERACT'87', Elsevier Science (North-Holland), 109-116.

McCabe, T. (1976): A complexity measure. *IEEE Transactions on Software Engineering,* SE-2, 6, 308-320.

Oberquelle, H.; Kupka, I. & Maass, S. (1983): A view of human-machine communicafion and co-operation. *International Journal of Man-Machine Studies,* 19, 309-333.

Petri C.A. (1980): Introducfion to general net theory. in: W. Bauer (ed.) *Lecture Notes in Computer Science vol.* 84 *"Net Theory and Applications'* (Springer, New York), 1-19.

Rauterberg, M. (1992): An empirical comparison of menu-selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts, *Behaviour & Information Technology,* 11, 227-236.