USER INTERFACES

AMME: An "Automatic Mental Model Evaluator" to Analyse User Behaviour Recorded on Logfiles.

Matthias Rauterberg
Work and Organizational Psychology Unit, Swiss Federal Institut of Technology (ETH), Zurich, Switzerland

Aim

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

State of the art

There are different formalisms for constructing user models; TAC, ETAC, CLC, COMS, CCT, the different kinds of grammars (BNF, EBNF, etc.), and state transition nets. Using any of these formalisms the analyst must always de- sign the pure (="error free") user model in a top down approach. Then he can try to prove his model with "error free" empirical data. This is difficult and insufficient, and one of the conse- quences is that most of the formal models exist only as paper versions and have not been im- plemented as computer programs.

Bottom up approach

If there is a possibility to construct user models in an automatic, bottom up approach, then the han- dling with formal models becomes easy. A se- quence 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 form a Petri-net ("folding" operator; Oberquelle et al. 1983). The "folded" Petri-net is a formal description ("model") of the procedural knowledge of the user's behaviour.

Why Petri-nets ?

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 subnets. 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 analyse our user model in a dynamic fashion.

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 1981). 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 another important feature. Examples will be presented. Analyzing logfiles in an automatic way enables the investigator to calculate applied statistics. Results will be presented (Rauterberg 1992).

References:

Komwachs, K. (1987): A quantitative measure for the complexity of man- machine interaction process. in: Bullinger, H-J. & Shackel, B., eds. "Human-Computer-Interaction INTERACT87", Elsevier Science (North-Holland), 109-116.

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

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

Rauterberg, M. (1992): A Method of a Quantita- tive Measurement of Cognitive Complexity .In: Proceedings of the 6th European Conference on Cognitive Ergonomics held in Budapest, September 6-11, in press.

In: H. Luczak, A. Cakir & G. Cakir (eds.), Work With Display Units--WWDU'92 (Abstract Book, p. E44). Berlin: Technische Universität - Institut für Arbeitswissenschaften.