Parallel or event-driven goal setting mechanism in Petri net based models of expert decision behaviour

M. Rauterberg, M. Fjeld, and S. Schluep

Researchgroup Human Machine Interaction (MMI)
Institute for Hygiene and Applied Physiology (IHA)
Swiss Federal Institute of Technology (ETH)
Clausiusstrasse 25, CH-8092 ZURICH, Switzerland
+41-1-632 70 82, rauterberg@iha.bepr.ethz.ch

ABSTRACT

Decisions and actions produced by computer users (*observed process*) contain much information about their mental models, individual problem solving strategy and underlying decision structure for a given task. Our tool AMME analyses observed processes and automatically extracts a Petri net description of the task dependent decision structure (*logical structure*). This net is extended by goal setting structures (modelling) and executed (*simulated process*). The aim is functional equivalence between observed and simulated processes. Two modelling strategies, *event-driven* and *parallel* goal setting, are presented.

Keywords: Behaviour driven modelling, cognitive modelling, Petri net, goal setting strategy.

INTRODUCTION

What mental models are and how they work, is quite unclear. Most of the known modelling approaches are based on the assumption that the "mental model maps completely to the relevant part of the conceptual model, e.g. the user virtual machine (Carroll, et al., 1991). Unexpected effects and errors point to inconsistency between the mental model and the conceptual model" (Van der Veer, et al., 1990). This one-to-one mapping between the mental model and the conceptual model of the interactive system implies a *positive* correlation between the complexity of the observable behaviour and the complexity of the assumed mental model. But this assumption seems to be wrong.

Based on empirical results (Rauterberg, 1993), the complexity of observable *novice* behaviour is larger than that of *experts*. Hence, the behavioural complexity is *negatively* correlated with the complexity of the mental model. If the cognitive structure is too simple, the concrete task solving process must be filled up with heuristic or trial and error behaviour. Learning how to solve a specific task with a given system means that the behavioural complexity decreases and the cognitive complexity increases. Now, one of the central question is: What kind of knowledge is stored in the cognitive structure?

THE MEASUREMENT OF COMPLEXITY

The symbolic representation of the machine system consists of the following elements: 1. objects (things to operate on), 2. operations (symbols and their syntax), and 3. states (the 'system states'). The mental model of the user can be structured in representing: objects, operations, states, system structure, decision and task structure. A Petri net can be described as a mathematical structure consisting of two disjoint, non-empty sets of nodes, S-elements (signified by a circle or an oval '( )') and T-elements (signified by a square '[ ]'), and a binary flow relation F (signified by an arrow ‚->‘) (Petri, 1980). Bauman, et al. (1986) showed, that Petri nets are equivalent to formalism based on production rules (like CCT of Kieras, et al. (1985)). In this sense, our approach can also be subsumed under 'logic modelling'. The main operations (relations) between two Petri nets are *abstraction*, *embedding* and *folding* (Genrich, et al., 1980).

The *folding operation* in the Petri-net theory is the basic idea of the approach presented in this paper. Folding a process means to map S-elements onto S-elements and T-elements onto T-elements while keeping the F-structure.

Each state corresponds to a system context, and each transition corresponds to a system action. This sequence is called a 'process'. An example of the task solving process of an expert user is presented here (see also Fig. 1). The user's task is to find out how many data records there are in a given data base (DB) consisting of file A, file B and file C of a given database (DB). First, the system goes from (Main menu) to (Start menu) with system transition ['G_2'], where it opens the DB. Then the system goes back to the (Main menu) with ['M_3']. Now, the user selects function key ['F_3'] and goes to the file selection menu, where he selects file A by pressing key ['1'], bringing him back to the (Main menu). Pressing ['I'] brings him to the (Info) module. Pressing ['d'], all DB information is displayed sequentially on three different screens (3* ['M_22']) and the system automatically goes to (Info) with ['M_11']. But part of the *task relevant* information is visible only on the first screen, which is overwritten by the two following ones. Trying once more to see all information, the user tries ['d'] again, but the same reoccurs. After a third time pressing ['d'], the user correctly follows up with a blank ['BL'] to interrupt the scrolling. Hence, the first part of the task relevant information stays on the screen. Pressing ['F_10'], the rest of the information is displayed (2* ['M_22']) and the system automatically goes to (Info) by ['M_11']. Pressing ['h'] twice brings the user to (Main menu) and then to (Start menu). Pressing ['F_10'] brings the user to (MsDOS).

The aim of the folding operation is to reduce the elements of an observed empirical decision process to the minimum number of states and transitions, with the reduced number of elements being the 'logical decision structure'. Folding a decision process extracts the embedded net structure and neglects the amount of repetitions, sequential order, and temporal structure. The result of a folding operation applied to the behavioural sequence (Fig. 1) is the Petri net given in Fig. 2.

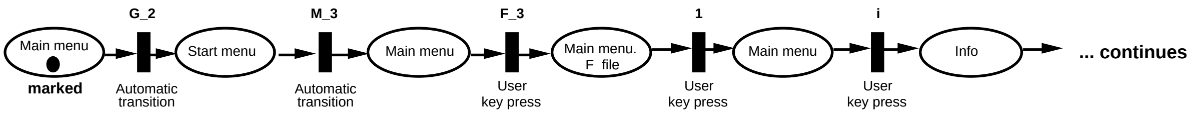

Fig. 1. Part of the original behavioural sequence of an expert with a relational database system (Rauterberg, 1992a).

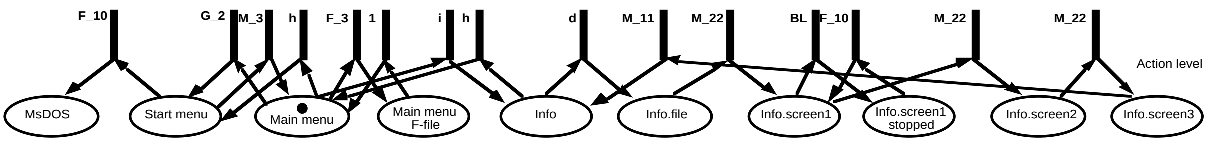

Fig. 2. *Model-1*: the pure 'logical structure' (the device model) of the logfile example sequence in Fig. 1.

C_{cycle} is the McCabe (1976) complexity measure (Formula 1), where F is the number of connections and [T+S] the number of nodes. (Different quantitative measures are discussed by Rauterberg (1992b).) P is a constant to correct the result of Formula 1 in the case of [F – (T + S) = -1]. In our context the value of P is 1.

$$\mathbf{C_{cycle} = F - (T + S) + P} \qquad \mathbf{(1)}$$

The C_{cycle} value of the model-1 in Fig. 2 is [30-25+1=6]; the complexity of the net shown in Fig. 2 is six. McCabe (1976) interprets C_{cycle} as the *number of linear independent paths* through the net. Other interpretations of C_{cycle} are *number of holes* in a net or *number of alternative decisions* carried out by the users.

Observing the behaviour of people solving a specific problem or task, is our basis for estimating 'model complexity‘ (C_{cycle}). The cognitive structures of users are not directly observable. Therefore, we need a theory with a method to estimate C_{cycle}, based upon observed behaviour. We call the complexity of the observable behaviour the 'behavioural complexity'. This behavioural complexity can be estimated by analysing the recorded concrete task solving process.

RECONSTRUCTION OF THE MENTAL MODEL

We carried out an empirical investigation to compare different types of interfaces (Rauterberg, 1992a). For the reconstruction of user's mental models we used the behavioural sequences of five different expert users, all solving the previously described DB task. The sequences were automatically recorded in ‚log files‘ by an interactive software program.

The logged sequence contains three different types of knowledge: (1) the pure logical structure (or device model) (Fig. 2) of the task, (2) the sequential structure of all goals, and (3) the temporal structure of all actions.

Action regulation theory (Hacker, 1994) offers a coherent body of principles for human-centred task and work design. For Hacker (1986) the work task is "the central category of psychological consideration of activity..., as decisive specifications for the regulation and organisation of the activities occur with the 'objective logic' of its contents". Therefore, in the context of action regulation theory, the *task* has great importance for the behavioural analysis. The concept of *complete task* calls for particular attention. Each complete task and action cycle starts with a goal setting part. In the analysis and design of human activities, the following characteristics of complete tasks must be taken into consideration:

(A) Task dependent setting of subgoals which are embedded in the superimposed task goal;

(B) Independent action preparation in the sense of taking on planning functions; and, selection of the means including the necessary actions for goal attainment;

(C) Mental or physical performance functions with feedback on performance pertaining to possible corrections of actions;

(D) Control with feedback on results and the possibility of checking the results of one's own actions against the set (sub-)goals.

Following the theoretical implications of action regulation theory, we can differentiate between (1) the action level (corresponding to C), (2) the goal instanciation level (corresponding to B), and (3) the cognitive level with the mental goal setting processes (corresponding to A). In the context of this paper we did not take into consideration the control and feedback level (D), but we plan to do this in our future work.

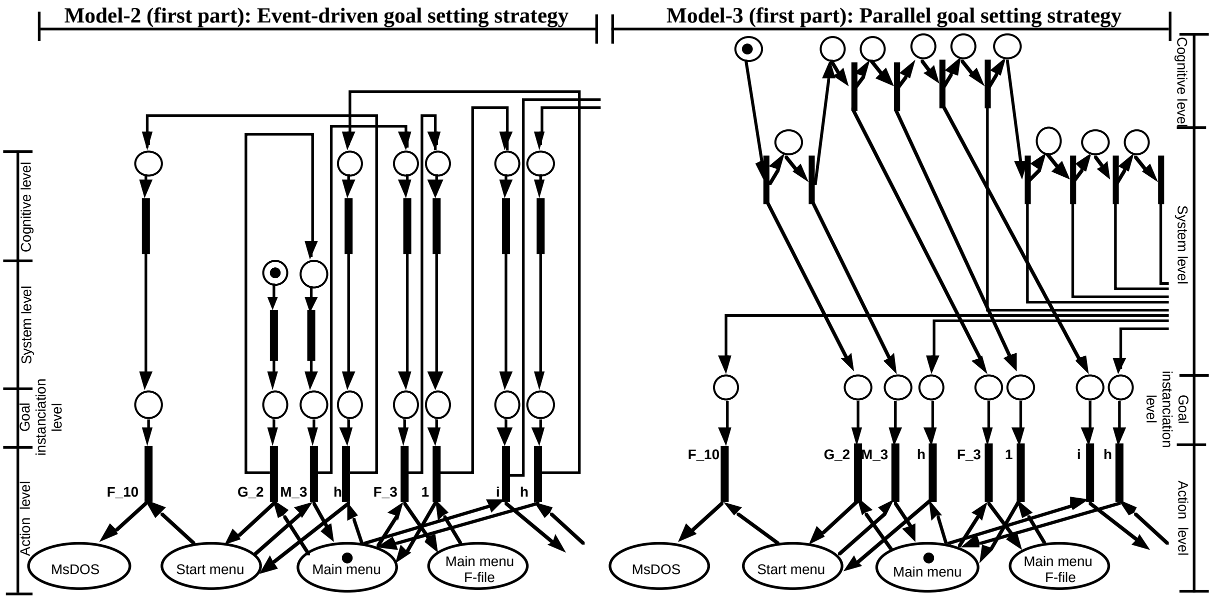

Fig. 3. a) *Model-2*: this Petri-net is equivalent to model-1 with added structure to model event-driven goal setting. b) *Model-3*: this Petri-net is equivalent to model-1 with added structure for parallel driven goal setting.

Model-1: The pure logical structure is automatically extracted with our tool AMME (Rauterberg, 1993, 1996) from the ‚log files‘. This net is called model-1 (see Fig. 2). Model-1 does not contain any knowledge about goals and time. It is a subset of the complete system description (S-, T-, and F-rules: the rule base to describe the whole interactive system behaviour as part of AMME (Rauterberg, 1996)). However, we know that a mental process takes place at a cognitive level. So we have to find a way to model these mental processes by adding structure to model-1. Petri nets enable us to do this in a unified way: any further levels can be described and modelled with the same elements as at the action level.

All observed actions (e.g. ['G_2'] and ['M_3']) that are automatically carried out by the system itself, are primarily not part of the cognitive level. Therefore, the system level must be a part of our model. Although the user cannot control the system level actions, he can take them into consideration. So, at the goal instanciation level a selected action is either set from the cognitive or from the system level.

In a first modelling approach, *event-driven goal setting* , we added a minimal structure to model-1, to increase the *functional equivalence* between the simulated, new model and the original behavioural sequence. This new model, based on model-1, we called model-2. In a second approach, *parallel goal setting* , we elaborated model-3.

Model-2: Our *event-driven goal setting* example is given in Fig. 3a. Each action (or event) at the action level *directly* activates a mental process at cognitive level or a system action at system level. For the next action to take place, a mental process or system action instanciates a corresponding goal at the goal instanciation level. Hence, there is an upward, direct synchronisation from action to cognitive and system level and a downward synchronisation via the goal instanciation to the action level.

Model-3: The *parallel goal setting* process (see Fig. 3b) was introduced by von Cranach's discussion of plans, 'anticipating representations', and intention (von Cranach, et al., 1982). For each goal at the goal instanciation level to be set, an anticipated counterpart must be set beforehand at the cognitive or system level. Parallel goal setting means that the mental goal setting process is running parallel to the action level, where the dialogue actions are executed. Hence, there is no upward synchronisation from action to cognitive or system level, only downward via goal instanciation to action level.

VALIDATION OF COGNITIVE GOAL SETTING STRATEGIES

To validate the different goal setting strategies, a simulation study of model-1,-2 and -3 was carried out. We modelled the task solving processes of five expert users, all solving the same task, as pure logical structure (model-1), with event-driven goal setting structures (model-2) and with parallel goal setting structures (model-3). With the Petri-net simulator PACE all the models (5(#subjects)*3(model-1,2,3)=15) were implemented and simulated. We generated six different task solving sequences with each model, giving a total of 90 simulated sequences. Each simulation stopped either because the net was dead or $N_{sim}=N_{org}$.

To estimate the similarity between the original sequence (e.g. Fig. 1) and each simulated sequence, we used the following procedure that guarantees a *functional equivalence* strictly based only on (re-)produced actions:

1. Number all transitions ['G_2' = '1', 'M_3' = '2', 'F_3' = '3', …, 'F_10' = '25'] in Fig. 1 consecutively. The number R is the rank-position of each transition [t] in the original behavioural sequence.
2. Assign these numbers to all generated transitions [t] of each simulated sequence. For example, one of the shortest simulated sequences we found, was generated with model-1: ['F_3', '1', 'G_2', 'F_10']. The rank positions R of these four transitions are: ['3', '4', '1', '19']. In general, if a transition appears m times in a original sequence and n times in the simulated sequence, the rules are as following: For the case n<=m, the simulated transitions get the ranks of the corresponding original transitions. For the case n>m, the first elements are handled as in the first case, whereas the rank of the elements m+1..n is m.
3. Calculate a 'similarity ratio' (SR; see Formula 2). SR is a sufficient measure for the similarity between the simulated sequence and the original sequence. N is the number of all fired transitions in a sequence. The maximum of R_{org} is equal to N_{org} (N_{org} in Fig. 1 is 25). SR is only valid for simulated sequences that fulfil the following condition: $N_{sim} \le N_{org}$. SR of the above example ['F_3', '1', 'G_2', 'F_10'] is 13%.

$$SR = \left[1 - \left\{ \sum_{t=1}^{N_{sim}} \left| R_{org,t} - R_{sim,t} \right| + \sum_{N_{sim+1}}^{N_{org}} \max\left(R_{org}\right) \right\} \Big/ N_{org}^{2} \right] * 100\% \qquad (2)$$

4. Average the similarity ratios of all simulated sequences per model (see Table 1). The results in Table 1 show that with increasing complexity of the mental model (C_{cycle}), the similarity ratio (SR) tends to 100%. We can also see that the standard deviation of SR decreases continually with increasing C_{cycle}.

Table 1. The model complexity (C_{cycle}) and similarity ratio (SR) of model-1, -2, and -3.

Model-no.	1	2	3
C_{cycle}: mean ± standard deviation	13 ± 5	43 ± 17	57 ± 25
C_{cycle}: min…max.	6…18	22…68	30…97
SR: mean % ± standard deviation	41 ± 28	66 ± 21	88 ± 11
SR: min…max. %	3…79	36…98	67…100
# simulated sequences	30	30	30

Three analysis of variances were done with StatView 4.02 (1993). Independent variables were model (1..3) and subject (1..5), dependent variable was similarity ratio (SR). Model was the only significant main effect (ANOVA, F=38.743, p<0.001), whereas subject (ANOVA, F=0.564, p≤0.690) and subject model interaction (ANOVA, F=1.068, p≤0.395) both were non-significant. So the similarity ratio increases from model-1 to model-3 and seems to be subject independent. We have not tested whether our results are task dependent. We got a clear, positive correlation between model complexity (C_{cycle}) and similarity ratio (SR) (R=0.660, 95% interval [0.524…0.763], p<0.001). The same analysis split by subject also gave a weaker, but still significant correlation (R=0.532, 95% interval [0.365…0.665], p<0.001).

From this outcome we can conclude that (1) model-2 is better than model-1, and model-3 is better than model-2, but (2) even the most elaborated model-3 can not guarantee a SR of 100%.

DISCUSSION

In the original sequence there was a cyclic behaviour (Rauterberg, et al., 1997). The user tried twice to see the task relevant information, but without success. The third time he succeeded. This is a typical learning effect based on trial and error. Analysing learning effects with Petri nets was achieved with other strategies (Rauterberg, et al., 1995). It seems to be difficult to simulate learning effects with event-driven goal setting, because repeated behaviour cannot be integrated in that modelling concept. For the parallel goal setting approach though, repeated behaviour is reflected in the model, with improved results from one to the next repetition. However, since there is no bi-directional synchronisation between the cognitive and the action level for the parallel goal setting approach, the similarity is still less than 100%. An integration of the advantages of both strategies will be a possible target for our further research.

CONCLUSION

We can conclude from the validation results that the SR value is considerably higher for the models with added goal setting (model-2 and model-3) than for the pure logical structure (model-1). So it seems to be possible to develop a completely automatic modelling tool based on a bottom up approach. Moreover, we see that with increasing model complexity (C_{cycle}), the mean value of SR increased and the standard deviation got smaller. Parallel goal setting performed better than event-driven goal setting. With the parallel goal setting strategy, we were able to include learning effects from the real task solving process.

REFERENCES

R. Bauman and T.A. Turano, Production based language simulation of Petri nets. Simulation 47 (1986) 191-198.

J. Carroll and J. Reitman-Olson, Mental models in human-computer interaction. In: M. Helander, Ed., Handbook of Human-Computer Interaction (North-Holland, 1991, 45-65).

M. von Cranach and R. Harré (eds.), The analysis of action. Cambridge University Press, 1982.

H.J. Genrich, K. Lautenbach and P. S. Thiagarajan, Elements of general net theory. In: W. Bauer, Ed., Lecture Notes in Computer Science 84 'Net Theory and Applications' (Springer, 1980, 21-163).

W. Hacker, Action regulation theory and occupational psychology. Review of German empirical research since 1987. The German Journal of Psychology 18(2) (1994) 91-120.

W. Hacker, Arbeitspsychologie. Huber, 1986.

D.E. Kieras and P.G. Polson, An approach to the formal analysis of user complexity, International Journal of Man-Machine Studies 22 (1985) 365-394.

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

C.A. Petri, Introduction to general net theory. In: W. Bauer, Ed., Lecture Notes in Computer Science 'Net Theory and Applications' (Springer, 1980, 1-19).

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

M. Rauterberg, A method of a quantitative measurement of cognitive complexity. In: G.C. van der Veer, M.J. Tauber, S. Bagnara and A. Antalovits, Eds., Human-Computer Interaction: Tasks and Organisation (CUD, Roma 1992b, 295-307).

M. Rauterberg, AMME: an automatic mental model evaluation to analyze user behaviour traced in a finite, discrete state space. Ergonomics 36 (1993) 1369-1380).

M. Rauterberg and R. Aeppli, Learning in man-machine systems: the measurement of behavioural and cognitive complexity. In Proceedings of IEEE International Conference on Systems, Man and Cybernetics--SMC'95 (Vol. 5, 1995, IEEE Catalog Number 95CH3576-7, 4685-4690). Piscataway: IEEE.

M. Rauterberg, A Petri net based analyzing and modelling tool kit for logfiles in human-computer interaction. In H. Yoshikawa & E. Hollnagel (Eds.), Proceedings 'Cognitive Systems Engineering in Process Control--CSEPC'96 (268-275). Kyoto University (1996).

M. Rauterberg and E. Ulich, Information processing for learning systems: an action theoretical approach. In Proceedings of IEEE International Conference on Systems, Man and Cybernetics--SMC'96 (Vol. 3, 1996, IEEE Catalog Number: 96CH35929, 2276-2281). Piscataway: IEEE.

M. Rauterberg, M. Fjeld and S. Schluep, Goal setting mechanism in Petri net models of human decision making. In Proceedings of IEEE International Conference on Systems, Man and Cybernetics--SMC'97 (1997, in press).

Van der Veer, S. Guest, P. Haselager, P. Innocent, E. McDaid, L. Oesterreicher, M. Tauber, U. Vos and Y. Waern, Desiging for the mental model: an interdisziplinary approach to the definition of a user interface for electronic mail systems. In: D. Ackermann and M. Tauber, Eds., Mental Models and Human-Computer Interaction 1 (North-Holland, 1990, 253-288).

TIME AND SPACE IN PROCESS CONTROL

csapc '97

EDITORS
SEBASTIANO BAGNARA
ERIK HOLLNAGEL
MICHELE MARIANI
LEENA NORROS

SIXTH EUROPEAN CONFERENCE ON COGNITIVE SCIENCE APPROACHES TO PROCESS CONTROL

BAVENO, ITALY
23/26 SEPTEMBER 1997

CNR
Istituto di psicologia – Consiglio nazionale delle ricerche

Roma, 1997, ISBN 88-85059-08-2