

# How to Measure the Behavioural and Cognitive Complexity of Learning Processes in Man-Machine Systems

Matthias Rauterberg and Roger Aeppli

Work and Organization Psychology Unit, Swiss Federal Institute of Technology (ETH)  
Nelkenstrasse 11, CH-8092 Zurich, Switzerland, email: rauterberg@ifap.bepr.ethz.ch

**Abstract:** A framework to conceptualize measure of behaviour complexity (BC), system complexity (SC) and task complexity (TC) was developed. From this framework cognitive complexity (CC) is derived as  $CC = SC + TC - BC$ . A learning experiment was carried out to investigate the development of mental structures. Six subjects were carefully instructed to operate a commercial database management system (DBMS). The long-term knowledge about general DBMS- and computer experience was measured with a questionnaire once at the beginning of the investigation. On three weeks in a row all subjects had to solve the same task twice repeated in an individual session, overall there are six solutions of the same task. At the beginning of each of the three individual sessions the short-term knowledge about the task and the tool was measured with a short questionnaire. With the special analysing program AMME the logical structure of each empirically observed task solving process was extracted. This logical structure is given as a Petri net. The behavioural complexity (BC) of this net structure can be measured with the McCabe-measure. With our framework the cognitive complexity (CC) can be derived from the empirically gained BC. The main results are: (1) The time structure and BC measure different aspects of the learning process; (2) the time structure is overall-positively correlated with BC and negatively correlated with CC; and (3) as well the long-term- as the short-term knowledge has an increasing predictive power with the time structure, but not with BC and CC.

## 1 Introduction

We live in a dynamic and irreversible changing world. We are information processing systems and have a huge learning potential. What happens to humans, if they have to behave in an approximately static environment? If we need growth (in a psycho-dynamic sense) and development, how long we are able to tolerate contexts that fix and constrain our activities? There is a lot empirical evidence that humans are getting bored if the context is characterized by repetitiousness, lack of novelty, and monotony [Smith 1981]. Ulich differentiates between boredom and monotony [Ulich 1987]. Boredom emerges from the feeling of not having enough possibilities to be active. Monotony emerges from the feeling of doing always the same things. "Monotony is a consequence of standardisation of the work process" ([Ulich 1987] p. 8). On the other side there are all research results of stressed and over-loaded workers.

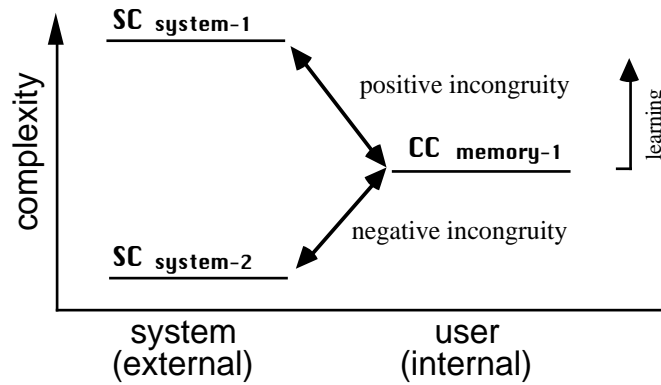
We have to realize and to accept that humans do not stop learning after end of school. We are compelled to learn and to make experiences our whole life. Human information processing can not be independent of this life-long learning. In this sense, humans are open systems. In his law of requisite variety Asby [Ashby 1958] pointed out, that for a given state of the environment, an open system has to be able to respond adaptively, otherwise the adaptability and the ability of the system to survive is reduced. A learning system, without input or with constant input, either decays or (in the best case) remains the same. Learning and the need for variety implies, that with constant variety of the context the requisite variety of the system tends to decay over time. This is a strong argument against 'one best way' solutions in work design (cf. [Ulich 1987]).

## 2 Activity and Incongruity

Investigators of novelty assume, that living systems (like humans) are motivated by an information seeking behaviour. In situations, which are characterized by sensory deprivation, humans are intrinsically looking for stimulation. They increase the complexity of the context or the perception of it. On the other side, humans try to avoid situations with a high amount of stimulation, dissonance, or stress. Hunt designated this amount of increased complexity as 'incongruity' [Hunt 1963].

If the complexity of the mental model ('cognitive complexity') is less complex than the complexity of the context (e.g., an interactive system), then humans try to optimize this positive incongruity [Rauterberg 1995]. Seeking behaviour starts, when the positive incongruity sinks below an individual threshold or changes to negative incongruity (deprivation). Behaviour of avoidance can be observed, when the positive incongruity exceeds an individual threshold (dissonance, stimulation overflow). Most of daily situations can be characterized by positive incongruity.

We shift the semantic and theoretic problems from incongruity to complexity [Rauterberg 1995]. Doing this, we can define incongruity in a more precise way. Incongruity is the difference of internal and external complexity [see Fig. 1]. Now we have to look for a good definition of complexity (see the discussion in [Rauterberg 1992]).



**Figure 1.** The difference between the external and the internal complexity is called *incongruity*.

In man-computer interaction we are able to measure the complexity of human behaviour (e.g., explorative activities; see [Rauterberg 1993]). With some plausible assumptions we are also able to estimate the complexity of user's mental model (see [Rauterberg 1992]). The complexity of the context (the internal structure of the interactive software) can be measured.

### 3 Learning and Activity

Learning is a permanent process that changes our long-term knowledge base in an irreversible way. The structure of our long-term memory changes to more complexity and higher abstraction. The basic idea of our memory model is that the variety on one level can be reduced to the invariant structure. This invariant structure forms the next higher, more abstract level of learning. Learning implies abstraction. Humans under non standardised conditions evolve very abstract invariants during their lifetime. Learning as a driving force for irreversible developments is the most underestimated factor in human behaviour, especially in the work and organisational context.

Learning increases constantly the complexity of the mental model. This is an irreversible process. One consequence is, that the contextual complexity must increase appropriately to fit the human needs for optimal variety. Based on the empirical result in [Rauterberg 1993], that the complexity of the observable behaviour of novices is larger than the complexity of experts, we concluded that the behavioural complexity is negatively correlated with the complexity of the mental model. Thus it is possible to estimate the cognitive complexity based on the measurement of the behavioural complexity, the measurement of the system complexity and the measurement of the task complexity (for a more detailed discussion see [Rauterberg 1992]).

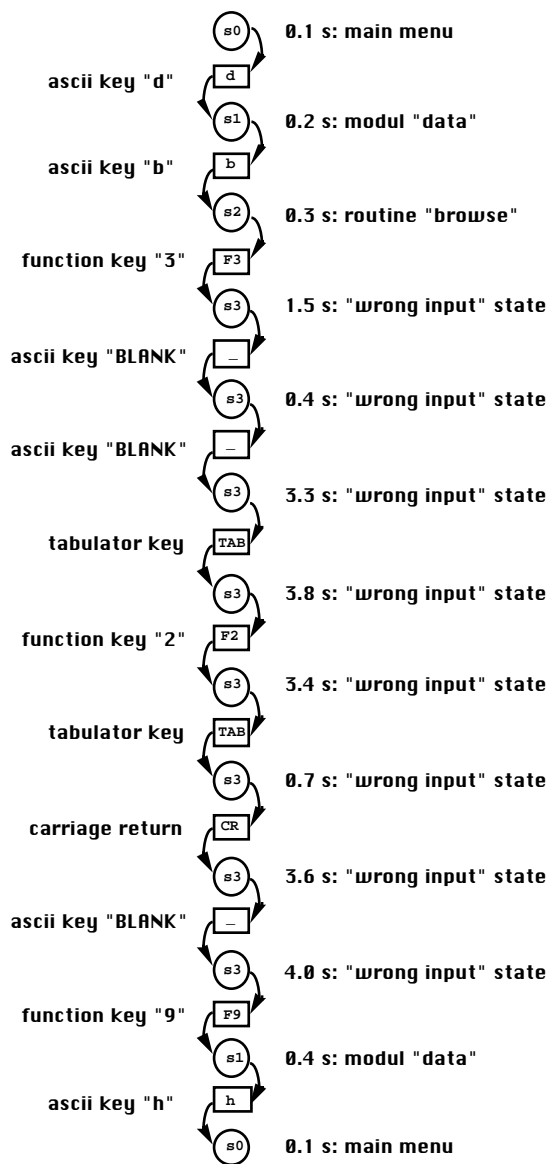
### 4 The Measurement of Complexity

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.

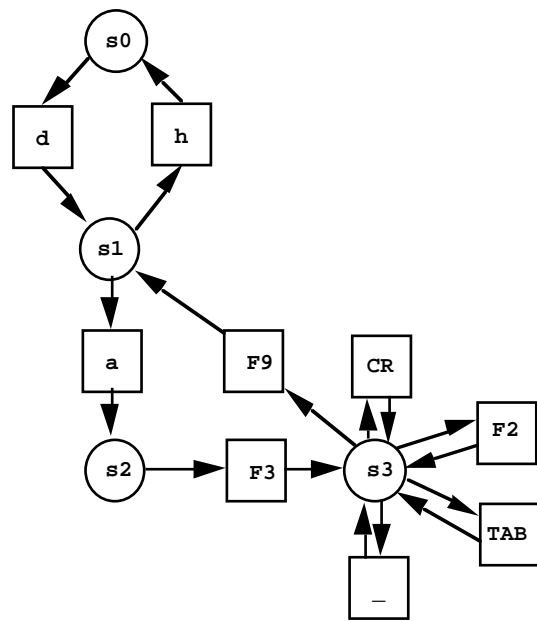
A net can be described as a mathematical structure consisting of two non-empty disjoint sets of nodes (S-elements and T-elements), and a binary flow relation (F). The flow relation links only different node types and leaves no node isolated [Petri 1980]. Petri nets can be interpreted in our context by using a suitable pair of concepts for the sets S (signified by a circle '( )') and T (signified by a square '[ ]') and a suitable interpretation for the flow relation F (signified by an arrow '->').

Bauman and Turano [Bauman and Turano 1986] showed, that Petri nets are equivalent to formalism based on production rules (like Cognitive-Complexity-Theory [Kieras and Polson 1985]). In this sense, our approach can be subsumed under 'logic modeling', too. Applying the path algebra approach of Alty to analyze the adjacency matrix [Alty 1984], we can get all "elementary paths in the network (e.g., paths which do not traverse an arc more than once). This algebra exhibits closure and the closure matrix gives all possible elementary paths between nodes" ([Alty 1984], p. 125).

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. The result is the structure of the performance net. Each state corresponds to a system context, and each transition corresponds to a system operation. This sequence is called a 'process' [see Fig. 2]. An elementary process is the shortest meaningful part of a sequence: (s') -> [t'] -> (s'').



**Figure 2.** Part of the interactive behaviour with a relational database system recorded in a logfile. The whole process of this example is based on 12 transitions and 12+1=13 states. The number on the right side is the 'time per keystroke' in seconds (s).



**Figure 3.** The 'folded' Petri net of the example sequence in Figure 1.

**Description of the situation in Figure 2:** The user's task was to search for a given data record in a particular file. He started correctly in the main menu (s0) and activated with the menu-option 'd' the module 'data' (s1). In the next step he activated with the menu-option 'b' the output routine 'browse' (s2). In this system state (s2) the user realized that an incorrect file was active, so he tried to change the active file by pressing the functionkey F3. This operation is normally correct, but only in the main menu and on the module level, not on the routine level. The system reacted with a help-message "Wrong input, press any key to continue". This message was incorrect, because only the functionkey F9 could be used to return. The system persisted in the 'wrong input state' (s3) and waited, until the user found the 'mouse-hole' F9 by trial and error. It is interesting to note, that the user went directly back to the main menu to continue his sub-task ('to change the active file'), although he could do this operation in (s1), too.

If the observable behavior can be recorded in a complete ...-> (state) -> [transition] -> (state) ->... process description [see Fig. 2], then the analysis and construction of the net structure of this process are simple: You have only to count the number of all different states and transitions used, or to mark on a list the frequencies of each state and transition used in the process. But, if the observable behavior can only be recorded in an incomplete (e.g., ...-> (state) -> [transition] -> [transition] ->... or ...-> (state) -> (state) -> [transition] ->...) process description, then the analysis and construction of the net structure of this process are difficult. You have to find out the correct state (transitions, resp.) between both transitions (states, resp.). Unfortunately, this is the most frequent case in practice. For these cases we need automatic tool support. In the last years we developed a tool, that gives us the possibility to analyze any processes with an incomplete process description, that are generated by finite state transition nets.

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 'decision structure'. Folding a

decision process extracts the embedded net structure and neglects the information of the amount of repetition and of the sequential order.

A simple pattern matching algorithm looks for all 'elementary processes' in the sequence. A composition algorithm (the folding operation) is now able to build up the Petri net combining all elementary processes. The result of a folding operation of our example sequence [see Fig. 2] is the Petri net given in [Fig. 3.] This special net with four different states and nine different transitions is the whole net structure we need to produce the process given in [Fig. 2]. Measurable features of the behavioral process are: number of states and transitions totally used, number of different states and different transitions used, dwell time per state and transition, etc. These measurements can be easily done based on a protocol of the user's behavior automatically recorded by an interactive software program (the dialog system) in a 'logfile'.

To measure complexity we use the Ccycle metrics of McCabe [McCabe 1976]. With Ccycle we have a useful quantitative metric to measure complexity. We are discussing the advantages and disadvantages of four different quantitative metrics in the context of an empirical investigation elsewhere (see [Rauterberg 1992]). The complexity measured with Ccycle is defined by the difference of the total number of connections (T: transition) and the total number of states (S: state). The parameter P is a constant to correct the result of Formula 1 in the case of a sequence (T - S = -1); the value of P in our context is 1.

$$\text{Ccycle} = T - S + P \quad \text{[Formula 1]}$$

The measure Ccycle of the example in Figure 3 is (9 - 4 + 1 = 6); the complexity of the net shown in Figure 3 is six. But, what could this number mean? McCabe [McCabe 1976] interprets Ccycle as the number of linear independent paths through the net. Other interpretations of Ccycle 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 "cognitive complexity (CC)". The cognitive structures of users are not direct observable, so we need a method and a theory to use the observable behaviour as one parameter to estimate CC. A second parameter is a description of the action or problem solving space itself. The third parameter is an "objective" measure of the task or problem structure.

We call the complexity of the observable behaviour the "behavioural complexity (BC)". This behavioural complexity can be estimated by analysing the recorded concrete task solving process, which leads to an appropriate task solving solution. The complexity of a given tool (e.g. an interactive system) we call "system complexity (SC)". The last parameter we need is an estimation of the "task complexity (TC)".

The necessary task solving knowledge for a given task is constant. This knowledge embedded in the cognitive structure (CC) can be observed and measured with BC. If the cognitive structure is too simple, then the concrete task solving process must be filled up with a lot of heuristics or trial and error strategies. Learning how to solve a specific task with a given system means that BC decreases (to a minimum = TC) and CC increases (to a maximum = SC). We assume, that the difference (BC-TC) is equal to the difference (SC-CC).

To solve a task, a person needs knowledge about the dialogue structure of the interactive software (measured by SC) and about the task structure (measured by TC). SC is an upper limit for TC (SC >= TC); this aspect means, that the system structure constrains the complexity of the observable task solving space. Now we can state with the constraints (BC >= TC) and (SC >= CC), that:

$$BC - TC = SC - CC \quad \text{[Formula 2]}$$

Transforming this Formula 2 to get CC alone, results in Formula 3:

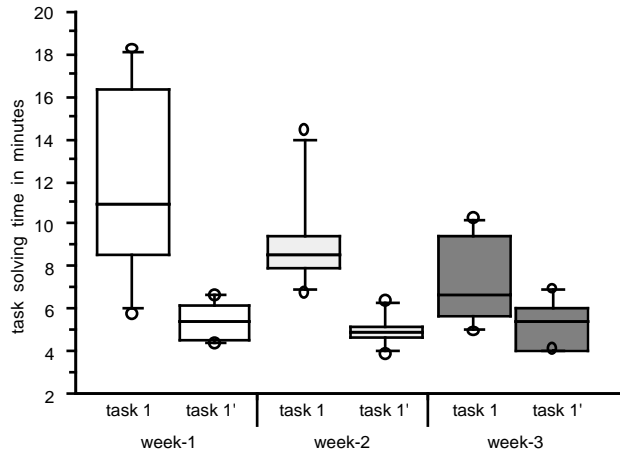
$$CC = SC + TC - BC \quad \text{[Formula 3]}$$

The parameters SC and TC can be estimated either in a theoretical or in an empirical way. The parameter SC is given by the concrete system structure, so we have to apply a given complexity measure to this system structure (theoretical way). The empirical way to estimate SC is to take the maximum of all observed BCs per task. To calculate TC all apriori descriptions of task structures are usable (theoretical way). If we have empirical data of different task solving processes of different persons, then we can estimate TC using the minimum of all observed BCs per task. Given a sample of different complete task solving processes, the best approximation for TC seems to be the minimal solution regarding a specific complexity measurement. One plausible consequence of this assumption is that CC is equal to SC in the case of "best solution" (TC=BC). This is the approach we are presenting here.

## 5 Results of a Learning Experiment

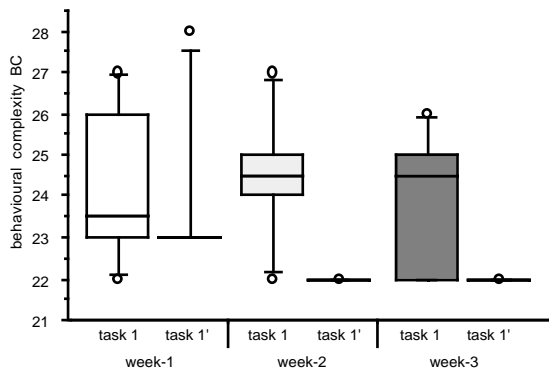
A relational database management system with a character-oriented menu interface was the test system. Six users (6 men; average age of 25 ± 3 years) had have to solve one task in two different versions on three weeks in a row [Rauterberg and Aeppli 1995]. The difference between task-1 and task-1' was only on the formulation level ('task

description'), but on the structural level both tasks were identically. The duration of the actual task solving session was about 30 minutes. Each keystroke with a timestamp was recorded in a logfile. Each user needed about 45 minutes for the whole experiment (2 tasks, individual sessions). We measured the task solving time per task. Based on the logfiles we could measure the behavioural complexity (BC) with our analysing tool AMME (cf. [Rauterberg 1993]). To estimate the cognitive complexity [see Formula 3] we have to determine SC and TC. SC can be estimated as the maximum of all observed BC's ( $SC = 28$ ). TC can be estimated as the minimum of all observed BC's ( $TC = 22$ ).

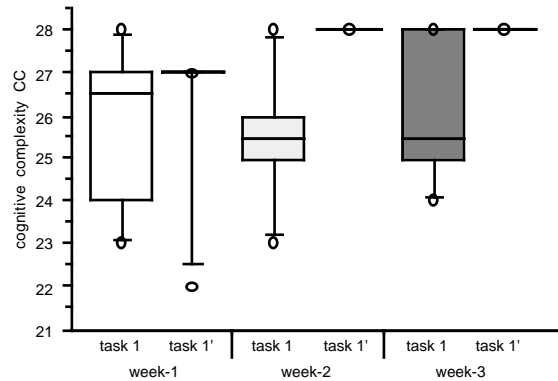


**Figure 4.** The Box plot of the average task solving time.

The difference of the task solving time between task-1 and task-1' is significant ( $meantask-1 = 9.5 \pm 3.8$ ,  $meantask-1' = 5.2 \pm 0.9$ ,  $F(1,10) = 15.3$ ,  $p \leq .003$ ). There is also a significant difference between the three days ( $meanday1 = 8.6 \pm 4.7$ ,  $meanday2 = 7.2 \pm 2.9$ ,  $meanday3 = 6.3 \pm 2.0$ ,  $F(2,20) = 3.6$ ,  $p \leq .045$ ). The interaction term is nearly significant ( $F(2,20) = 3.2$ ,  $p \leq .061$ ; see [Fig. 4]).



**Figure 5.** The average behaviour complexity BC.



**Figure 6.** The average cognitive complexity CC.

Only the difference of the behavioural complexity between task-1 and task-1' is significant ( $meantask-1 = 24.3 \pm 1.7$ ,  $meantask-1' = 22.6 \pm 1.4$ ,  $F(1,10) = 12.7$ ,  $p \leq .005$ ). All other differences are not significant (see Figure 5). Only the difference of the cognitive complexity between task-1 and task-1' is significant ( $meantask-1 = 25.7 \pm 1.7$ ,  $meantask-1' = 27.4 \pm 1.4$ ,  $F(1,10) = 12.6$ ,  $p \leq .005$ ). All other differences are not significant [see Fig. 6].

In the first session ('1. week') the normally positive correlation between  $time_{task-1}$  and  $BC_{task-1}$  ( $r = +.69$ ;  $r = \pm .87$ , and  $r = +.80$ ) changes to a negative correlation between  $time_{task-1'}$  and  $BC_{task-1'}$  ( $r = -.48$ ). This result is caused by one user with the most data base experience who produced the most complex net ( $BC=28$ ) in the shortest time (4.5 min). If we compare the correlation pattern of week-1 with the correlation pattern of week-2 and week-3 then we can observe a qualitative change from a positive to a negative correlation of the long-term and of the short-term knowledge with BC. We interpret this result as exploratory behaviour during the first session: the more knowledge, the greater the tendency to explore. Due to lack of variance for week-2 and week-3 the correlation between  $BC_{task-1'}$  and the other measures can not be computed.

To get an overview over the development of the different influences of long- and short-term knowledge on task solving time and BC, we calculated all relevant correlations (collapsing task-1 and task-1'). The influence of long-term knowledge on task time is increasing from week to week ( $r = -.02$  week<sub>1x1</sub>,  $-.28$  week<sub>1x2</sub>, and  $-.56$  week<sub>1x3</sub>). A very similar effect can be observed for the influence of the short-term knowledge about the task-tool mapping on task solving time ( $r = -.06$  week<sub>1x1</sub>,  $-.35$  week<sub>2x2</sub>, and  $-.49$  week<sub>3x3</sub>). The measurement of the short-term knowledge about the task-tool mapping shows a high retest reliability (inter-test correlation  $r = +.87$  week<sub>1x2</sub>,  $+.86$  week<sub>1x3</sub>,  $+.90$  week<sub>2x3</sub>).

Both–long-term and short-term knowledge–change from a positive to a negative correlation from week-1 to week-2 and week-3. If we assume that BC is a valid measure for the amount of exploratory behaviour, then we can come up with the following plausible interpretation for this result: First, an exploratory behaviour helps the user to learn as much as possible about the system; during this first phase the long- and short-term knowledge enables and guides the exploration. After this first exploration phase the user can produce the minimal solution based on his new knowledge about the system. This interpretation could explain, why the correlation between BC of week-1 and BC of week-2 is zero and increases to a significant correlation between week-2 and week-3 ( $r = -.03$  week<sub>1x2</sub>,  $+.63$  week<sub>2x3</sub>).

## 6 Conclusion

Three different results are important: (1.) Our assumption that "learning how to solve a specific task with a given system means that BC decreases and CC increases" seems to be correct. (2.) If all users reach the correct task solution (minimum of BC, see [Fig. 5]) then the variance between users disappear (see the columns of Task-1' of day-2 and day-3 in Figure 5). Furthermore, (3.) we can conclude from the empirical results that we must discriminate between the time structure (cf. [Johnson 1985]) and the logical structure of a task. The time structure can be measured with the task solving time. The logical structure of a task can be extracted with our analysing tool AMME, and the complexity of this logical structure can be measured with the McCabe-measure. At day-2 all users learned completely the logical task structure (minimum of BC, no variance, see [Fig. 5]). Learning the time structure means to accelerate the task solving process. One psychological dimension of the time structure seems to be self-confidence in doing the right things at the right time.

## 7 References

- [Alty 1984] J.L. Alty, The application of path algebras to interactive dialogue design, *Behaviour and Information Technology*, 3 (1984) 119-132.
- [Ashby 1958] R.W. Ashby, Requisite variety and its implications for the control of complex systems. *Cybernetica*, 1(2), (1958) 1-17.
- [Bauman and Turano 1986] R. Bauman and T.A. Turano, Production based language simulation of Petri nets, *Simulation*, 47 (1986) 191-198.
- [Genrich et al. 1980] H.J. Genrich, K. Lautenbach and P. S. Thiagarajan, Elements of general net theory, pp. 21-163. In: W. Bauer, Ed., *Lecture Notes in Computer Science 84 'Net Theory and Applications'* (1980).
- [Hunt 1963] J.M.V. Hunt, Motivation inherent in information processing and action. In: O.J. Harvey (ed.): *Motivation and social interaction: cognitive determinants*. New York (1963).
- [Johnson 1985] S. L. Johnson, Using mathematical models of the learning curve in training system design. In: *Proceedings of the Human Factors Society 29th Annual Meeting, Vol. II*, (1985) 735-739.
- [Kieras and Polson 1985] 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.
- [McCabe 1976] T. McCabe, A complexity measure, *IEEE Transactions on Software Engineering*, SE-2 (1976) 308-320.
- [Petri 1980] C.A. Petri, Introduction to general net theory, pp. 1-19. In: W. Bauer, Ed., *Lecture Notes in Computer Science 'Net Theory and Applications'* (1980).
- [Rauterberg 1992] M. Rauterberg, A method of a quantitative measurement of cognitive complexity, pp. 295-307. In: G.C. van der Veer, M.J. Tauber, S. Bagnara and A. Antalovits, Eds., *Human-Computer Interaction: Tasks and Organisation* (1992).
- [Rauterberg 1993] M. Rauterberg, AMME: an automatic mental model evaluation to analyze user behaviour traced in a finite, discrete state space. *Ergonomics*, 36(11), (1993) 1369-1380.
- [Rauterberg 1995] M. Rauterberg, About a framework for information and information processing of learning systems, pp. 54-69. In: E. Falkenberg, W. Hesse & A. Olive (eds.) *Information System Concepts – Towards a consolidation of views*. London (1995).
- [Rauterberg and Aeppli 1995] M. Rauterberg and R. Aeppli, Learning in man-machine systems: the measurement of behavioural and cognitive complexity, pp. 4685-4690. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. (vol. 5, IEEE Catalog No. 95CH3576-7, 1995).
- [Smith 1981] R.P. Smith, Boredom: A Review. *Human Factors*, 23(3), (1981) p. 329-340.
- [Ulich 1987] E. Ulich, Umgang mit Monotonie und Komplexität. *Technische Rundschau*, 5, (1987) 8-13.