M. Fjeld, S. Schluep & M. Rauterberg (1998): Quantification of task related activity by statistical and analytical methods. In Proceedings 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, pp.591-596. Oxford: Elsevier Science, Ltd.

Quantification of task related activity by statistical and analytical methods

M. Fjeld[1], S. Schluep[1] & M. Rauterberg[2]

[1]*Institute of Hygiene and Applied Physiology (IHA)*
Swiss Federal Institute of Technology (ETH)
Clausiusstr. 25, CH-8092 Zurich, Switzerland
{fjeld, schluep}@iha.bepr.ethz.ch

[2]*Center for Research on User-System Interaction (IPO)*
Technical University Eindhoven (TUE)
NL-5612 AZ Eindhoven, Netherlands
rauterberg@ipo.tue.nl

Abstract: Behaviour of expert and novice database users solving the same task was recorded. Several successful strategies were identified. Since there are more users than strategies, some users applied the same strategy. The aim was to develop methods grouping users with common strategy. Following three approaches (correlation, intersection, and exclusion), a metric among task solving behavioural sequences was defined. Measured data was organised in matrix systems relating all users. Statistical and analytical interpretation of matrices showed distinct groups. A common denominator for a group can indicate a strategy.

Keywords: automatic recognition, model based recognition, behavioural science, user interface, problem solvers, multidimensional systems, statistical analysis.

1. INTRODUCTION

The aim of this paper is to develop automatic methods finding task solving strategies. Such knowledge is of interest to understand how users behave in a newly designed system, and to thereby giving them better support. Also, it may also help understanding how expert users behave in highly complex systems.

Under certain conditions, strategies may also be obtained by protocol analysis (Ericsson and Simon, 1984). Protocol analysis implies manual inspection of video and verbal utterances in addition to logfiles. With simple tasks, this work can be overcome. For more complex tasks, protocol analysis has proved cumbersome. Semi-automatic generation of process models was studied by Ritter and Larkin (1994). Guided by their work, further principles for automatic recognition of user strategies and plans will be suggested.

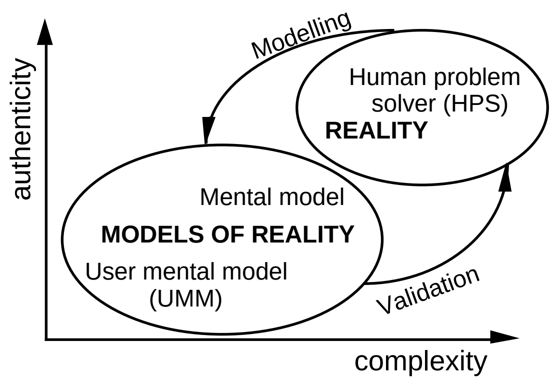

Fig. 1. A scheme showing the relation between models of reality and real humans (HPSs). Models are meant to represent objects and processes existing in reality.

This paper treats computer mediated, everyday task solving. A special case of mental models, called user mental model (Tauber, 1985) (UMM, Fig. 1) is introduced. UMMs can bring understanding about strategies people use when solving specific problems. UMMs can be represented in many ways, using plain text, Petri nets or state-transition vectors. In this paper, the latter representation was chosen to elaborate UMMs based on observable task solving strategy.

In general, a lot of task solving behaviour that is not strictly *task related* can be observed. If one single HPS is studied, it is hardly possible to single out the successful *strategy* from the *remaining behaviour*. One approach may be to study many users solving the same task. Since they all solve the same problem, it is likely that their common behaviour is what was required to solve the task.

A *strategy* is defined to be one (of many), possibly error free, successful task solving behavioural sequences for the current system and task. As soon as a complete strategy is accomplished, task solving is over. If users follow different strategies, a group of users may have one strategy in common, an other group a second one. Modelling means finding a measure for relation among users and thereby grouping users with common strategy. The common denominator for each group indicates a strategy.

Which strategy a user prefers, as well as other behaviour can tell us something about the particular HPS; for instance how a strategy was acquired. Given a behavioural task solving sequence, it is of interest to separate the *strategy* (which is more related to the task-system combination) from the *remaining behaviour* (which is more related to the HPS). The remaining behaviour may consist of partial strategies or strategies that would be successful within an other system and/or task.

In this paper, human perception and verbalisation will not be considered as part of the problem solving. Purely based on observable task solving behaviour, the aim is to develop automatic methods, applicable with simple as well as with complex tasks. Protocol analysis will only be used to validate the elaborated automatic methods.

2. SYSTEM DESCRIPTION

The system studied is a relational database program with 153 different dialogue states. A transition consists of a dialogue state and one (of many) possible user actions in that state. All possible transitions of the system are represented by a state-transition vector space. A state-transition-vector (STV, Formula 1) summarises a subject's task solving behaviour.

$$\left\{e_i^p\right\} \qquad (1)$$

$e_i^p \in N_0$ STV component

$p \in \left\{N_1, ..., N_6, E_1, ..., E_6\right\}$ user index

$i = 1, ..., n$ transition index

$n = 978$ number of transitions

The total number of possible transitions for the complete database program is given by n. An STV component's value tells how many times a certain transition was activated to solve the task. Talking about a *user* without any further detail, refers to the corresponding STV of that user. In both cases, it means the observed task solving behaviour of the user, expressed by an STV.

Since the order of activated transitions is not contained in the STV, order of user behaviour is only partly conserved. It is stored in an implicit form, given by the system dialogue structure and embedded in the STV structure.

3. TASK DOMAIN

An empirical investigation was carried out by Rauterberg (1992) to compare different types of expertise. For the reconstruction of UMMs, logfiles of six novice {N1,...,N6} and six expert {E1,...,E6} users were used, all solving the same task. The task was to find out how many data records there are in a given database consisting of three files. As soon as the required results were found, task solving activity was finished. An example UMM of a task solving process, based on E4, is presented in Rauterberg et al. (1997). For E4, 15 *different* transitions (number of positive STV components) were activated to solve the task. (Over all the users, *different* transitions was between 14 and 42). Some transitions were activated repeatedly, so the *total* number of activated transitions (sum of STV components) is 25. (Over all the users, *total* number of activated transitions was between 25 and 175).

4. BASIC QUESTIONS AND METHODOLOGY

Studying an STV of one user can tell us which system states the user passed by, which transitions that were activated in those states and how many times that occurred. Different users working with the same system are directly comparable, since their behavioural sequences only differ by the value of the vector components. In order to group users according to their behaviour, there is first a need to relate behavioural sequences.

Table 1: Relation among task solving behaviour has various *qualitative* aspects with corresponding *quantities.*

Quality	Quantity
Interaction	Correlation
Similarity	Intersection
Difference	Exclusion

Table 1 shows a few qualities of relationship and corresponding quantitative methods. A classical method is that of correlation. An alternative is to look for analytical methods. For instance, STVs can be considered as sets and represented by ellipses, as in Fig. 2, or by rectangles as in Fig. 4. The area of an ellipse or rectangle corresponds to the sum of the STV components values. Intersection area can be understood as (symmetric) similarity between two STVs. Exclusion areas can be understood as the (asymmetric) differences between two STVs.

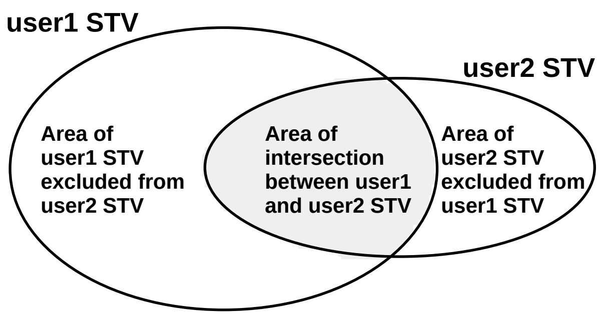

Fig. 2. Intersection area and exclusion areas between user1 and user2 STV.

Based on such considerations, the following questions are raised and corresponding methods are suggested:

1. What is the interaction among two behavioural sequences? Method: *correlation*.
2. What do two behavioural sequences have in common (similarity)? Method: *intersection* (Fig. 2).
3. What do two behavioural sequences not have in common (difference)? Method: *exclusion* (Fig. 2).

Table 2: The three suggested methods, metrics, metric nature and grouping algorithms. CORR means a standard correlation method, the other metrics are defined by Formula 2, 3 and 4.

Method	Metric	Metric nature	Grouping algorithm
Correlation	CORR	Statistical	Statistical
Intersection	$M_{p,q}^{IS}$, $M_{p,q}^{BIS}$	Analytical	Statistical
Exclusion	$M_{p,q}^{EX}$	Analytical	Analytical

For each method, a metric (Table 2) is elaborated. The order of the metric may be symmetrical (the metric applied from user1 STV to user2 STV is the same as the metric applied from user2 STV to user1 STV; correlation and intersection) or asymmetric (the metric applied from user1 STV to user2 STV is *not* the same as the metric applied from user2 STV to user1 STV; exclusion). Each of these metrics can be applied between all possible pairs of STVs, giving a matrix system. A grouping algorithm is applied on this matrix, indicating groups with related behaviour.

For each group suggested by the grouping algorithm, a strategy may be approximated. The procedure is to create an STV with maximum number of non-zero components smaller or equal to the STVs components of that group. This step will not be explored any further in this paper.

The following presentation will proceed from more statistically based to more analytically (non-statistically) based methods.

4.1 Correlation method

In this method correlation applied among pairs of STVs measures degree of interaction among users. The matrix is then analysed by multi-dimensional-scaling (MDS, Systat, 1989) to indicate groups of users.

Metric

Pearson correlation is applied to measure interaction between two STVs. This procedure gives a diagonal dominant, symmetrical matrix with possible values between minus one (opposite interaction), via zero (no interaction) and one (complete interaction). For Fig. 3 the observed values are between -0.003 and 0.948 (without considering the diagonal elements).

Grouping algorithm

The correlation matrix is interpreted by two dimensional MDS, giving the plot of Fig. 3.

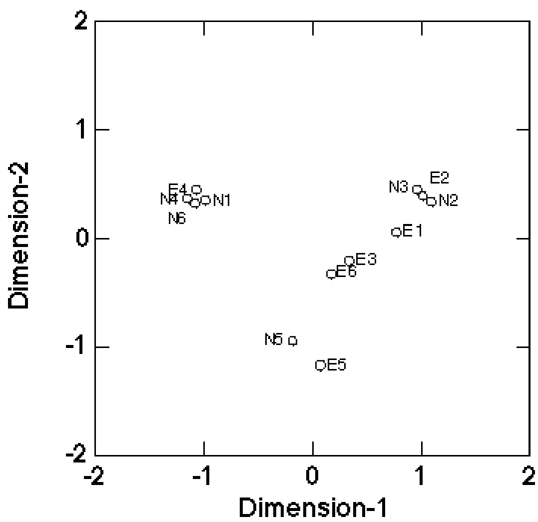

Fig. 3. MDS (r=1, Kruskal, Mono) plot with a Pearson correlation matrix gives RSQ=0.870.

Outcome

Fig. 3 shows that users may be grouped: {N1, N4, N6, E4}, {N2, N3, E1, E2, E3, E6} and {N5, E5}. Some of the users, like N5 and E5, may be interpreted as partly members of other groups, indicating parts of or combinations of strategies. According to the proportion of variance (RSQ=0.870), MDS explains some of the user data variance, but 13% remains unexplained.

4.2 Intersection method

This method is based on the observation that if two users follow the same strategy, that strategy will belong to the intersection of the two users STVs. However, this is no condition for this method to work. Two STVs may have a significant intersection, without having solved the problem with the same strategy.

Two STVs common part is the same, so the interaction matrix is symmetric and can be analysed by MDS.

Metric

Similar behaviour is measured by summing up the smaller STV components of the two STVs, thus considering the number of activated transitions common to both users.

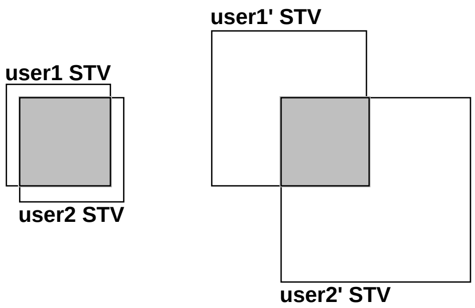

Fig. 4. User1 and user2 STVs represented by rectangles. In both situations, the intersection is the same, but the similarity between STVs are not. This indicates that a *normalisation* is required.

User1 and user2 STV can also be seen as sets, represented by rectangles (Fig. 4). For the intersection area (measured quantity) to be a valid measure of similarity (desired quality), a *normalisation* is required. It is possible to scale degree of intersection by the larger (max.), the average (mean) or the smaller (min.) sum of the intersection areas. Scaling by the smaller of the areas corresponds to scaling by the maximum possible intersection. Expressed in the state-transition vector space gives the intersection metric of Formula 2.

$$M_{p,q}^{IS} = \frac{\sum_{i=1}^{n} \min\left(e_i^p, e_i^q\right)}{\min\left(\sum_{i=1}^{n} e_i^p, \sum_{i=1}^{n} e_i^q\right)} \qquad (2)$$

$p,q \in \{N_1,...,N_6,E_1,...,E_6\}, p \neq q$ user indices

Ignoring repetitive behaviour is a mean to reduce complexity. Replacing each STV component >1, by 1, results in a *binary* state-transition-vector. An intersection metric based on this vector is given by Formula 3.

$$M_{p,q}^{BIS} = \frac{\sum_{i=1}^{n} \min\left(e_i^p \cdot e_i^q, 1\right)}{\min\left(\sum_{i=1}^{n} \min\left(e_i^p, 1\right), \sum_{i=1}^{n} \min\left(e_i^q, 1\right)\right)} \qquad (3)$$

$p,q \in \{N_1,...,N_6,E_1,...,E_6\}, p \neq q$ user indices

Formula 2 and 3 both give a symmetrical matrix where the elements take possible values between zero (no similarity) and one (equality). For Fig. 5 (Formula 2) the observed values are between 0.078 and 0.929 (without considering the diagonal elements). For Fig. 6 (Formula 3) the comparable values are between 0.182 and 0.882.

Grouping algorithm

The symmetrical exclusion matrix is interpreted by MDS, giving the plots in Figs. 5 and 6. In both cases, the users can be divided into three groups, {N1, N4, N5, N6, E4}, {N2, N3, E1, E2, E5} and {E3, E6}. Again, some users, like E3 and E6, may be interpreted as members of other groups, indicating a combination of strategies.

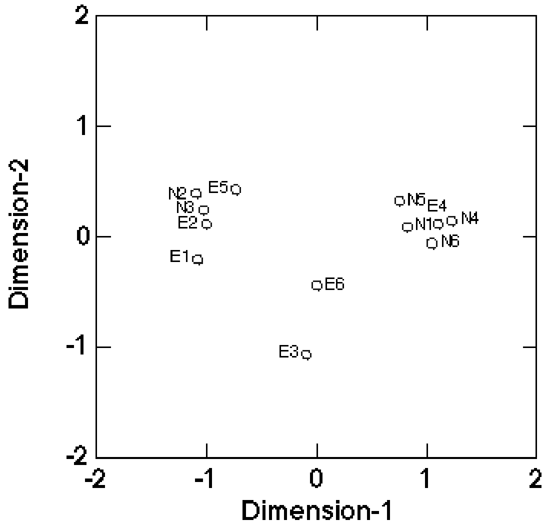

Fig. 5. MDS (r=1, Kruskal, Mono) plot with a normalised intersection matrix gives RSQ=0.975.

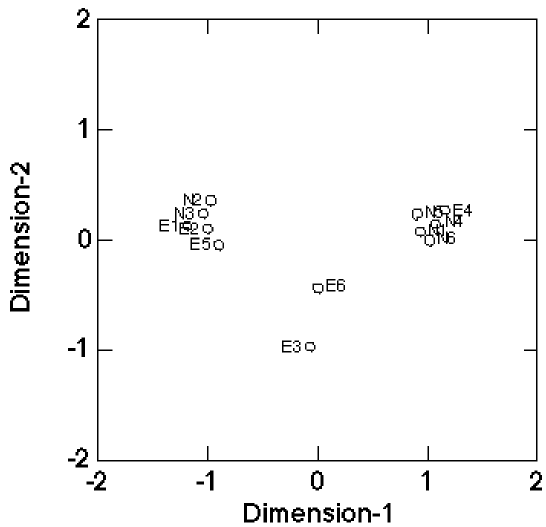

Fig. 6. MDS (r=1, Kruskal, Mono) plot with a binary normalised intersection matrix gives RSQ=0.995.

Outcome

According to the RSQ of Fig. 5 (RSQ=0.975) and of Fig. 6 (RSQ=0.995), most of the variance in the user data can be explained. However, the binary based plot of Fig. 6 is slightly better than that of Fig. 5. This is surprising, since the binary metric ignores information about repetitive behaviour. Maybe information about repetition is redundant in the context of this method.

4.3 Exclusion method

This method is based on the exclusion as a metric of difference. Exclusion among two users can be seen as two disjoint areas (Fig. 2), unless there is equality. The area of user1 STV (Fig. 2) excluded from user2 STV (Fig. 2), is not the same as the area of user2 STV excluded from user1 STV. Since the two exclusion areas are different, the resulting matrix is asymmetric and the method does not allow for MDS as grouping algorithm.

Metric

This method applies the exclusion metric of Formula 4 between all users p and q, giving an asymmetrical matrix (Table 3). The metric gives the difference between two STVs by estimating how much of user p STV (column, Table 2) is excluded from user q STV (row).

$$M_{p,q}^{EX} = \sum_{i=1}^{n} \left| \min\left(e_i^p - e_i^q, 0\right) \right| \qquad (4)$$

$p, q \in \{N_1, ..., N_6, E_1, ..., E_6\}$ user indices

Table 3: Matrix elements, given by Formula 4, quantify exclusion of user q (column) STV from user p (row) STV.

	N1	N2	N3	N4	N5	N6	E1	E2	E3	E4	E5	E6
E6	6	43	47	51	70	50	47	35	73	5	171	0
E5	17	15	14	69	48	67	23	7	64	21	0	24
E4	9	44	47	56	70	55	47	35	73	0	171	8
E3	17	41	41	68	77	62	28	28	0	21	162	24
E2	17	15	16	68	81	67	19	0	66	21	143	24
E1	20	16	19	73	85	72	0	7	54	21	147	24
N6	3	41	44	28	48	0	47	30	63	4	166	2
N5	3	39	42	41	0	33	45	29	63	4	132	7
N4	2	41	42	0	55	27	47	30	68	4	167	2
N3	16	11	0	68	82	69	19	4	67	21	138	24
N2	18	0	15	71	83	70	20	7	71	22	143	24
N1	0	41	43	55	70	55	47	32	70	10	168	10

Grouping algorithm

Grayscale representation (Fig. 7) is based on the exclusion matrix (Table 3) and generated by Mathematica (Wolfram, 1991) *ListDensityPlot* with the inverted exclusion matrix as input. The matrix is inverted to achieve a consistent plot. Fig. 7 is only meant as a visualisation of Table 3, and is not an exact mapping. Since division by zero is not defined, the diagonal elements of Table 3 were directly mapped to the darkest graytone.

A group is defined as users with few differences. Hence, the inverted quantity of exclusion is significant; *inclusion*. Table 3 and Fig. 7 will be now be interpreted as an indicators for *inclusion*. Lower values, respectively darker matrix elements correspond to higher degree of inclusion.

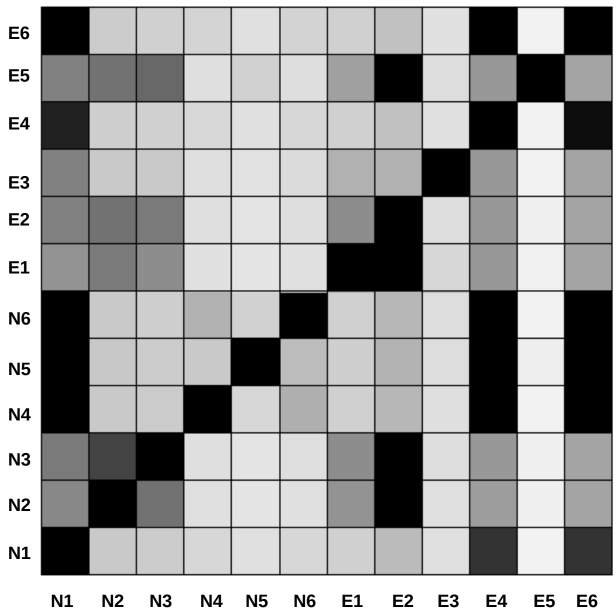

Fig. 7 Grayscale exclusion matrix. Darker elements mean lower exclusion, i.e. higher inclusion of user q (column) in user p (row) STV.

An iterative *predictor-corrector* algorithm is suggested to interpret Table 3. The *corrector* is an estimator for a threshold value so that considering elements between zero and corrector will give predicted number (*predictor*) of user groups. The stop criterion for the iteration method is that number of user groups given by the corrector, equals the value of the predictor. Research on converge is part of future work, so it is simply assumed. For each iteration the corrector is modified in order to meet the stop criterion, according to the following rules: If too few inclusion relations are considered (i.e. the corrector is too close to zero), the number of groups will be higher than the predictor. If too many inclusion relations are considered (i.e. the corrector is too far from zero), many or all of the users will be related by inclusion statements, and the number of groups will be lower than the predictor. The predictor is given the value three. By visual inspection of Fig. 7 it appears reasonable to consider the darkest matrix elements only. Since these elements have numerical values equal to or below 8 (Table 3), the initial value of the corrector is chosen to be 8. Each STV is similar to itself, so diagonal elements are ignored, giving Table 4.

Table 4: These four inclusion relations can be derived.

Relation no.	User q included in user p STV
1	q = N1, p∈ {N4, N5, N6, E6}
2	q = E4, p ∈ {N4, N5, N6, E6}
3	q = E6, p ∈ {N4, N5, N6, E4}
4	q = E2, p ∈ {N2, N3, E1, E5}

All users that are related by an inclusion relation are defined to belong to one group. Since the three first inclusion relations are interrelated, they give one group. The remaining, fourth similarity relation gives a second group. Users not appearing in any similarity relation each define a separate group.

Outcome

Hence, Table 4 gives these groups: {N1, N4, N5, N6, E4, E6}, {N2, N3, E1, E2, E5} and {E3}. The number of groups was assumed to be three, so the stop criterion has been met. If it had not been met, it would have been necessary to try with a higher or lower corrector (according to the above mentioned rules) and go back to the start of the predictor-corrector algorithm. The algorithm is repeated until the stop criterion is met (convergence).

4.4 Validation method

To validate the outcome of the three preceding methods, protocol analysis (Ericsson and Simon, 1984) of the task was performed. This is manual work, based on analysis of video and verbal utterances in addition to logfiles. It is mostly feasible for simple tasks, where users follow one or a few strategies. The analysis showed that there are three distinct strategies solving the task; S1, S2 and S3. Table 5 shows the users according to their strategy.

Table 5: Validation data show three distinct strategies and group the users according to the strategy they applied.

Strategy	Users succeeding by strategy
S1	N1, N4, N5, N6, E4, E6
S2	N2, N3, E1, E2, E5
S3	E3

5. DISCUSSION

The correlation and intersection methods do not correspond fully with the validation data (Table 5), whereas the exclusion method gives the same results. So, with the current combination of system, task and users behaviour, the exclusion method is the best one. This means, for this case, that the purely analytical method was the best one. Measured by the RSQ, intersection is better than correlation method, which is purely statistical. So, in the context of this paper, statistical methods offer less explaining power than the analytical methods.

However, the exclusion method does not say anything about possible combinations or parts of strategies applied. For such questions, the statistically based methods seem more relevant. This is confirmed by Hanson et. al. (1991), treating class (or: group) assignment with Bayesian methods: "Such classes are also 'fuzzy'; instead of each case being assigned to a class, a case has a probability of being a member of each of the different classes".

6. CONCLUSION

For the present combination of system, task and user behaviour, it was possible to develop methods grouping users according to their task solving strategy.

7. FUTURE PERSPECTIVES

Results for one task only were acquired. To make the methods more reliable, it is necessary to evaluate several tasks. For each task the methods will be validated by manual protocol analysis. It is of particular interest to find out the exclusion method performs with other, more complex tasks.

It is also planned to study learning experiments, in order to recognise the acquisition process of strategies.

ACKNOWLEDGEMENT

Morten Fjeld is a PhD fellow of the Research Council of Norway.

REFERENCES

Ericsson, K. A. and Simon H. A. (1984). *Protocol analysis, verbal reports as data*. The MIT Press.

Hanson, R., Stutz, J. and Cheeseman, P. (1991). *Bayesian Classification Theory*. Technical Report FIA-90-12-7-01, NASA, Ames Research Center, AI Branch.

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

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

Rauterberg, M., Fjeld, M. and Schluep, S. (1997). Parallel or event driven goal setting mechanism in Petri net based models of expert decision behavior. *Proceedings of CSPAC'97* (Bagnara, S., & Hollnagel, E., Mariani, M. and Norros, L. (Ed.)), Roma, CNR, pp. 98-102.

Ritter, F. E. and Larkin, J. H. (1994). Developing Process Models as Summaries of HCI Action Sequences. *Human Computer Interaction*, **9**, pp. 345-383.

SYSTAT Inc. (1989). SYSTAT®: The system for statistics. pp 93-166. SYSTAT PC program, version 7.0.1.

Tauber, M. J. (1985). Top down design of human-computer systems from the demands of human cognition to the virtual machine - an interdisciplinary approach to model interfaces in human-computer interaction. *Proceedings of the IEEE workshop on languages for automation*, Palma De Mallorca (E), pp. 132-140.

Wolfram, S. (1991). Mathematica®, A system for Doing Mathematics by Computer, 2nd Ed. Mathematica program version X3.0.1.1 for Silicon Graphics IRIX. AddisonWesley, pp. 164, 395, 819.