M. Fjeld, S. Schluep & M. Rauterberg (in press): Action-driven quantification of task-solving behaviour. In D. Harris (ed.) Engineering Psychology and Cognitive Ergonomics, Vol. 4. Hampshire: Ashgate, pp. 253-261.

Action-driven quantification of task-solving behaviour

Morten Fjeld, Samuel Schluep
Institute for Hygiene and Applied Physiology (IHA)
Swiss Federal Institute of Technology (ETH Zurich)

Matthias Rauterberg
Center for Research on User-System Interaction (IPO)
Technical University Eindhoven (TUE)

Abstract

Detection of task-solving strategies is one way to reconstruct cognitive structures. When the number of users exceeds the number of possible strategies, many users will have a strategy in common. Based on these common features, the aim of this paper is to find automatic methods which group users sharing the same strategy. Following two different methods, *intersection* and *exclusion*, we define a comparative metric among observed behavioural sequences. For all the users observed, we represent these measures by a matrix system. Multidimensional scaling or analytic interpretation of this matrix indicates distinct user groups.

Keywords

Automatic recognition, user interface, statistical analysis, grouping

Introduction

The aim of this paper is to develop automatic methods to identify *groups* of users accomplishing a task by the *same strategy*. Based on this information, we assume that cognitive structures representing a strategy can be established. Such structures are of interest to understand how users behave in a newly designed system, and thereby, to give them better support. Also, these structures may help to gain knowledge about how experts behave in highly complex systems.

It seems to be generally accepted that a problem is a conflictive situation that should be somehow *changed* and *solved*. Solving the problem cannot be restricted to purely linear, deterministic methods. There may be good reasons to try to solve the problem in various ways. In many cases, creative insights may well deeply modify the terms of the problem and offer help to manage it. Hence, tactics and strategies may be more related to *learning* how to use a system or more related to pragmatic *task solving*.

A *tactic* is defined as 'the technique or science of securing the objectives defined by a strategy. A *strategy* covers long-term planning and thus embeds tactics. ... From a most embracing viewpoint, strategy, as well as tactics, depend on the best possible knowledge of the system' (François, 1997). In our context, a strategy is one, of many, possibly error free, successful task solving behavioural sequences for a system and a task.

Efficiency of task solving means to what degree it is related to the given task. Inefficient behaviour may contain information about various tactics and how a successful strategy was achieved. Inefficient behaviour can consist of unsuccessful trials, exploring capabilities of the applied tools, repetitions and retries. These can all be components of a learning process, and may give insights to a user's cognitive structure about system and task.

For an observer, it is hardly possible to single out a successful strategy from inefficient behaviour. One approach may be to study many users solving the same task. Since they all solve the same problem with the same system, it is likely that their common behaviour is what was required to solve the task. If users follow different strategies, a group of users may have one strategy in common, an other group a second one.

To *quantify* common behaviour, we need a method to relate task solving behaviour of different users. User behaviour can be described by some kind of set, so help can be found in mathematics, offering at least seven kinds of relation: 'empty, universal, reflexive, symmetric, asymmetric, hybrid and transitive' (François, 1997). Two of these (symmetric and asymmetric) give a basis for our methods (intersection and exclusion).

Our aim is to develop automatic methods purely based on observable task solving behaviour, applicable with simple as well as with complex tasks. Protocol analysis, human perception and verbalisation will only be used to validate the elaborated, automatic methods.

Task domain and strategies

An empirical investigation was carried out to compare two types of user interfaces (Rauterberg, 1992). Users had to solve different benchmark tasks with a character-based user interface (CUI) and a graphical user interface (GUI) of a relational data base. Their behaviour was logged in files. Here, we use the logfiles of six novice {N1,...,N6} and six expert users {E1,...,E6} solving a task with a CUI. The task was to find out how many data records there are in a database consisting of files A, B and C. The goal was to operate the system, so that the number of records for each of the files was displayed on the screen. Three basically different strategies (S1, S2 and S3) were used to reach this goal (Schluep, Fjeld and Rauterberg, 1998). After the required results were found, task solving activity was finished.

Although all users succeeded, individual *effort*, measured by the number of keystrokes, was variable, even with the same strategy (Table 1). Since the number of keystrokes necessary is constant for each strategy, this indicates that *inefficient* behaviour varied among users.

Table 1
Range of the number of keystrokes related to each strategy.

Strategy	User	# Keystrokes (min.)	# Keystrokes (max.)
S1	N1, N4, N5, N6, E4, E6	10	71
S2	N2, N3, E1, E2, E5	33	167
S3	E3	73	73

Inefficient behaviour can be described as: i) Correction of mistakes, ii) retries after unsuccessful trials, iii) explorative behaviour to find some hint or trying functions which seem to be promising, iv) navigation, e.g. determining which mode is active or how to get back to the main menu, and v) attempting other strategies, which were abandoned later on.

Description of task solving behaviour

We define a *state* to be any n-dimensional space containing points representing the state description of a system. 'The totality of all possible states of a system, meaning the totality of different aspects the system can assume *for us* at an instant of *time*, forms a *set* called the *state space* of the system' (Rosen, 1972). 'A *state-transition* is the cross over from one state to another in a system. This cross-over can be strictly determined, probabilistic as in ergodic systems, or mean a deep change in the nature of the system, as in chaotic dynamics' (François, 1997). A cross-over in a database system is strictly determined.

A state-transition system is 'defined in terms of a set of overall states based upon a particular mask and a specification of conditions for transitions between these states' (Klir, 1991). The mask is a partial screen through which specific system states are observed. Masks can also be seen as variety reducers.

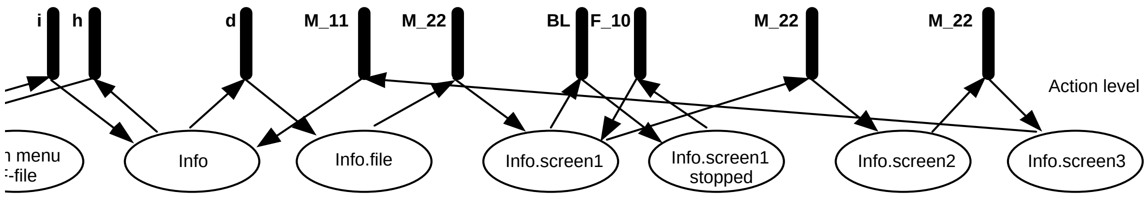

Figure 1 Part of state-transition model, as represented by a Petri-net.

The relational database program we used, can be modelled as a system with a mask of 154 dialogue states. Transitions symbolise the possible succession (or cross-over) of two states and the action (i.e. *system event* or *user keystroke*) associated with it. Figure 1 shows a fraction of the state-transition model of the program. Oval states represent dialogue states, bars represent transitions. The letter shown with the bars indicates action triggering the transition. The same action (e.g. system event M_22) may trigger different transitions.

Specifying the preceding state and the action identifies a transition unequivocally. Hence, we expand the *state space* (Rosen, 1972) by *actions*, giving a *state-transition* space. This enables us to represent user behaviour as a sequence of transitions between system states. In the *state-transition* space there is more than one transition possible in each state, so n (n=978) is significantly higher than the number of dialogue states.

A member of the *state-transition* space, *state-transition-vector* (STV, Formula 1), summarises a subject's task solving behaviour for one task. Each STV element represents the number of activation of a certain transition the user used to solve the task. Since the order of activated transitions is not contained in the STV, the order of user actions is only partly conserved. It is stored implicitly, given by the potential sequences of actions determined by the system dialogue structure.

$$\{e_i^p\}$$ **(Formula 1)**

$e_i^p \in N_0$	***STV component***
$p \in \{N_1, ..., N_6, E_1, ..., E_6\}$	***user index***
$i = 1, ..., n$	***transition index***
$n = 978$	***number of transitions***

Methodology

A classical method to compare task solving, user behaviour is correlation, which is a measure of proximity (Fjeld, Schluep, and Rauterberg, 1998). Analytical methods offer an alternative. Each ellipse in Figure 2 illustrates the set of transitions activated by a user. The intersection area (grey) represents similarity (what do two behavioural sequences have in common), the exclusion areas represent difference between two STVs.

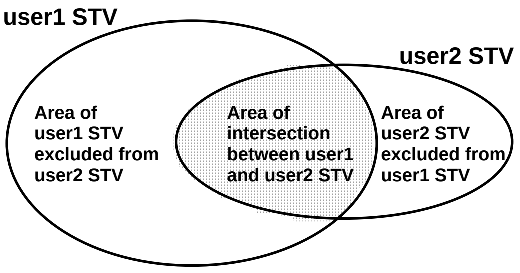
user1 STV
user2 STV
Area of user1 STV excluded from user2 STV
Area of intersection between user1 and user2 STV
Area of user2 STV excluded from user1 STV

Figure 2 Intersection and exclusion between user1 and user2.

For each method, we elaborate a metric (Table 2). The metric may be symmetrical or asymmetrical (metric applied from user1 to user2 does not give the same as when applied from user2 to user1). Applying a metric between all users gives in a matrix, analysed by a grouping algorithm.

Table 2
The two methods characteristics. See also Formula 1, 2 and 3.

Method/quantity of relation	Metric formula	Metric nature	Grouping algorithm
Intersection	$M_{p,q}^{IS}$ **and** $M_{p,q}^{BIS}$	**Analytical**	**Statistical**
Exclusion	$M_{p,q}^{EX}$	**Analytical**	**Analytical**

Intersection method If two users follow the same strategy, the strategy will be within their intersection area. Similar behaviour is measured by summing up the smaller STV element values of the two user STVs, thus considering the activation of transitions common to both users.

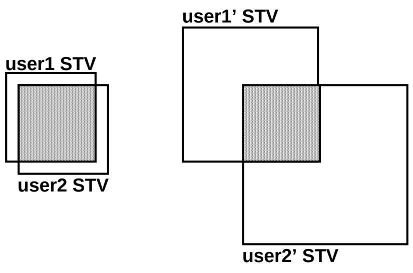

Figure 3 User1 and user2 STVs represented by rectangles.

User1 and user2 STV can also be seen as sets, represented by rectangles (Figure 3). In both situations, the intersection is the same (grey), but the similarity between STVs are not. This indicates that a normalisation is required. It is possible to divide degree of intersection by the larger (max.), the average (mean) or the smaller (min.) sum of the intersecting areas. We divide by the smaller of the areas, corresponding to the maximum possible intersection. In the state-transition space, this gives Formula 3. The observed values were between 0.078 (low) and 0.929 (high similarity).

$$M_{p,q}^{IS} = \frac{\sum_{i=1}^{n} \min\left(e_i^p, e_i^q\right)}{\min\left(\sum_{i=1}^{n} e_i^p, \sum_{i=1}^{n} e_i^q\right)} \qquad p, q \in \{N_1, ..., N_6, E_1, ..., E_6\}, p \neq q \quad \textbf{user indices} \qquad \textbf{(Formula 2)}$$

Grouping algorithm Interpreting the matrix by multidimensional scaling (MDS, dimensions: 2; r-metric: 1; iteration: 50; convergence: 0.005; regression: monotone), users seem to represent three groups, {N2, N3, E1, E2, E5} and {N1, N4, N5, N6, E4, E6} and {E3} (Figure 4). RSQ = 0.977, so we can explain most of the user data variance.

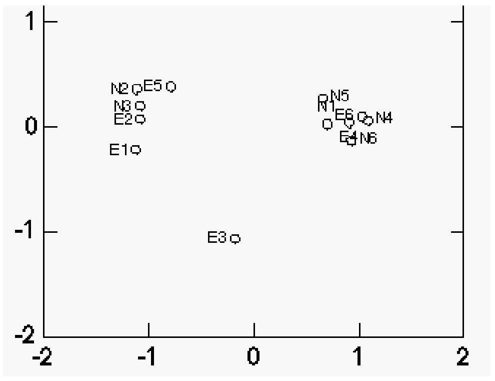

Figure 4 Multidimensional scaling of intersection matrix.

Exclusion method This method measures how much of one user behaviour is not shared by an other one. It gives an asymmetric matrix of distances between users, calling for another grouping algorithm than MDS. This method measures (Formula 3) how much of one STV (column index in Table 3) is excluded from a second one (row index), giving an mxm (m=12) asymmetric exclusion matrix (Table 4).

$$M^{EX}_{p,q} = \sum_{i=1}^{n} \left| \min\left(e_i^p - e_i^q, 0\right) \right| \qquad p,q \in \{N_1,\dots,N_6,E_1,\dots,E_6\}$$ **(Formula 3)**

Table 3
Numerical representation of the exclusion matrix.

	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 Elements of Table 3, except on the diagonal, below a *threshold* (Fjeld, Schluep & Rauterberg, 1998), indicate similarity. These elements give similarity relations (Table 4), so that element *column* 'is part of' element *row*. The first three relations are interrelated, giving one group. Relation four gives one group. E3 is part of no relation. This gives three groups: {N1, N4, N5, N6, E4, E6}, {N2, N3, E1, E2, E5} and {E3}.

Table 4
These four similarity relations can be derived from Table 3.

Relation	User STVs of each relation
1	**N1 is part of N4, N5, N6, E6**
2	**E4 is part of N4, N5, N6, E6**
3	**E6 is part of N4, N5, N6, E4**
4	**E2 is part of N2, N3, E1, E5**

Conclusion and future perspectives

For the present combination of system, task and user behaviour, it was possible to develop methods grouping users according to their task solving strategy. 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 if the methods perform with other, more complex tasks. It is also planned to study learning experiments, in order to recognise the acquisition process of strategies.

References

Fjeld, M., Schluep S. & Rauterberg M. (1998), 'Quantification of task related activity by statistical and analytical methods', in *Preprints of 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation on Man-Machine Systems*, pp. 591-596.

François, C. (1997) (ed.*), International Encyclopedia of Systems and Cybernetics,* K.G. Saur: Munich, pp. 297, 328 and 330.

Klir, G. (1991) (ed.), *Facets of system science*, Plenum Press: New York, p. 345.

Schluep, S., Fjeld, M. & Rauterberg, M. (1998), 'Discriminating task solving strategies using statistical and analytical methods', in T.R.G. Green, L. Bannon, C.P.Warren & J.Buckley (eds.) *Cognition and Co-operation*. University of Limerick: Ireland, pp. 121-126.

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

Rosen, R. (1972), 'Some systems theoretical problems in biology', in E. Laszlo (ed.) *The relevance of General Systems Theory*. Braziller: New York, p. 50.