DOI: 10.20378/irb-92881

# FORSCHUNGSBERICHT
# aus dem
# INSTITUT FÜR PSYCHOLOGIE

92/6

Ute Schmid and Uwe Konerding

## A Method for Analyzing Computer Users' Behavior Sequences[1]

1000 Berlin 10, Dovestraße 1-5 Telefon (030) 314-22370

92/6

Ute Schmid and Uwe Konerding

# A Method for Analyzing Computer Users' Behavior Sequences[1]



[1] We would like to thank Prof. Dr. Arnold Upmeyer for giving us permission to use the data gained in his research project "Computer aided diagnosis systems for the evaluation of CAD-systems" (DFG grant Ma 305/25-1f) and Prof. Dr. Klaus Eyferth and Dipl. Psych. Detlef Widowski for giving us permission to use the data gained in their research project "Cognitive processes in reading and understanding computer programs" (FIP of the Technical University Berlin grant 2/16).

Furthermore the authors would like to thank Dr. Hans-Günther Roth for contributing the idea of using a-priori defined sequences for validating the MDS-solution.

We gratefully acknowledge Dipl.-Psych. Gisela Böhm for reviewing our manuscript and Dipl.-Psych. Karin v. Rosen for correcting our English.

## Abstract

This paper presents a formal approach to analyzing behavioral sequences produced by computer users. Following this approach, the behavioral sequences are represented as square transition matrices, distances between the individual matrices are calculated, and the resulting distance matrix is scaled multidimensionally. The emerging scales reflect differences in the users' problem solving behavior and can be used as a guideline for detecting behavioral aspects influencing quality of task solution. The method was exemplarily applied to two different kinds of behavioral data, the first reported by Upmeyer, Meseke & Enzmann (1988) and the second by Eyferth, Widowski & Schmid (1991).

Keywords: computer user, behavior sequences, multidimensional scaling

The main problem in research of human-computer interaction is the identification of user characteristics influencing the effectiveness and correctness of performance. Such characteristics are of interest in the domain of both user interaction with software systems (e.g., Card, Moran & Newell, 1983, Polson & Kieras, 1984) and programming behavior (e.g., Johnson, 1988, Kessler & Anderson, 1989). In both domains, knowledge concerning relevant user characteristics and their relationship to aspects of performance is a necessary prerequisite for developing instruction programs and support strategies, and in the domain of interaction with software systems such knowledge may help constructing software which is sensitive to different kinds of users (cf. Potosnak, 1986).

The user characteristics mostly investigated are general personality factors and measures of previous experience (cf. Shneiderman, 1980, chap. 2.5 and chap. 3.5). However, focussing research on such predefined features, unduly restricts the scope of considered relevant characteristics. Especially characteristics of computer users' behavior during the process of problem solving have up to now been mostly neglected. In order to overcome these limitations it is necessary to conduct explorative empirical studies which are based upon data reflecting the user-computer interaction process as detailled and complete as possible. Measures describing the interaction process should be constructed and the relationship between these measures and quality of performance should be investigated.

The construction of measures reflecting relevant information of detailled behavioral data, however, is rather difficult, because the total information contained in an individual problem solving sequence is extremly complex. Furthermore, sequences produced by

different users vary with respect to their input elements and their lengths. Of course, it is possible to construct measures which at least reflect a part of the information contained in the sequences, as for example "number of corrections", "number of calls for help" (Moll, 1987) or "number of backward movements in produced texts" (Green, Bellamy & Parker, 1987). Possible sets of scales constructed in such a way, however, may be too selective to reflect those aspects of the sequences which are most crucial for discriminating users' problem solving behavior. Therefore, scaling procedures are needed which are not constrained by too many a-priori assumptions and which are sensitive to detecting yet unregarded, relevant behavioral aspects.

The purpose of this article is to propose a bipartite method which may help to identify relevant features of the users' problem solving behavior. The first part of the method is a procedure for the construction of scales which reflect differences between the users with respect to the succession of their behavioral elements. The second part of the method is a strategy for interpreting these scales. For the illustration of the proposed scaling procedure, two empirical applications are presented. The first example deals with users' problem solving behavior while generating a technical draft with a software system, and the second example with the reading behavior of programmers in a program memorization task.

## THE SCALING PROCEDURE

The only prerequisite for applying the scaling procedure is the representation of the behavioral data as a sequence of well defined elements. These may be,

for example, the commands of a software system or the syntactical units of a programming language. The complete scaling procedure consists of three successive steps: first, representing sequences as transition matrices; second, calculating distances between transition matrices; third, multidimensional scaling of the distances.

## Representing Sequences as Transition Matrices

The first step of the procedure serves to represent the different sequences in a comparable format. For this purpose, the same concepts are applied as those upon which the definition of a Markov chain is based (cf. Ethier & Kurtz, 1986). Each meaningful segment of the sequences that was produced at least once by at least one user during the experimental sessions is considered as a possible state of the sequence. This set of states constitutes the vocabulary for describing the individual sequences. Each sequence can then be represented as a matrix that contains information concerning the frequency of transitions for each ordered pair of states. The resulting individual matrices are all square and have the same number of rows and columns, being equal to the number of possible states.

According to the view presented here the most decisive information concerning the transitions from one state to the other is the absolute frequency normed with regard to the total number of transitions within the respective individual sequence. Consequently each element of the transition matrix can be calculated by

$$t_{ij} = \frac{f_{ij}}{l_p - 1} \qquad (1)$$

with
$t_{ij}$ = element of the transition matrix,
$f_{ij}$ = absolute number of transitions from state i to state j,
$l_p$ = total number of actually realized segments in sequence p (i.e. $l_p$-1 is the total number of transitions produced by subject p).

Dividing the absolute frequencies by the total number of transitions produces measures which reflect how typical the corresponding transitions are with regard to the respective sequence. Information concerning the length of the sequence is neglected.

## Calculating Distances between Transition Matrices

The second step of the procedure serves to determine measures of proximity between the different individual sequences. For this purpose we propose the Euclidian distance, that is

$$d_{pq} = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} (p_{ij} - q_{ij})^2 \right)^{1/2} \qquad (2)$$

with
$d_{pq}$ = distance between the sequences p and q,
$n$ = number of possible states,
$p_{ij}$ = cell element representing the transition from state i to state j in sequence p,
$q_{ij}$ = cell element representing the transition from state i to state j in sequence q.

This step of the scaling procedure results in one single symmetric square matrix of distances between the sequences. The number of rows and columns of this matrix is equal to the number of investigated subjects.

### Multidimensional Scaling of Distances between Sequences

The third and last step of the scaling procedure serves to render scale values assigned to the sequences and to reflect the dominant features which determine the distance between the transition matrices. For this purpose a multidimensional scaling approach (MDS) is applied. Because the matrix is symmetric and the distances are measured at least on interval level, a metric scaling model (Roskam, 1972) can be used. For the determination of the required number of dimensions, both the goodness of fit of the configuration as well as parsimony and interpretability of the representation should be taken into account (Shepard, Romney & Nerlove, 1972, pp. 9). The resulting dimensional values represent relations between the individual sequences and may serve as guides for detecting relevant user characteristics.

## A STRATEGY FOR INTERPRETATION

The scaling procedure presented so far may serve quite different purposes. In the context discussed here the MDS-solution will be applied for the detection of new behavioral aspects which influence the quality of performance. For this purpose an analytical strategy consisting of three steps is proposed.

## Testing the Diagnostical Information of MDS-Dimension(s)

The first step serves to investigate whether the MDS-dimension(s) contain information diagnostically relevant to users' quality of performance. In the case of a single dimensional solution, this can be tested by calculating a product-moment-correlation with the dimension for each considered measure of performance. In the case of a solution with more dimensions, a multiple correlation with the dimensions as predictors should be calculated for each measure of performance.

If relevant user characteristics are reflected in different patterns of transition between the possible states of the behavioral sequence, the squared correlation coefficients should be significantly higher than zero. If correlations are not significant, a further analysis of the dimensional values is not promising and can therefore be dispensed with.

## Comparing MDS-Dimension(s) with A-priori Defined Scales

The next step serves to compare the diagnostical value of the MDS-scales with that of other user scales. Such concurring scales could be measures for theoretically postulated aspects of users' problem solving behavior or measures constructed exploratively as descriptors of the users' solution process. The crucial question is, whether the dimensional values represent behavioral aspects which are relevant for predicting performance and which are <u>not</u> represented by the a-priori defined scales.

The comparison of dimensional values and a-priori defined scales can be done by means of a general F-test for an increment (Cohen & Cohen, 1983, p. 145). i.e.:

$$F = \frac{(R^2_{Y.AB} - R^2_{Y.A}) \;/\; k_B}{(1 - R^2_{Y.AB}) \;/\; (n - k_A - k_B - 1)} \qquad (3)$$

with

$R^2_{Y.AB}$ = squared multiple correlation with performance as criterion and MDS-dimension(s) and a-priori defined scales as predictors,

$R^2_{Y.A}$ = squared multiple correlation with performance as criterion and only a-priori defined scales as predictors,

$k_A$ = number of a-priori defined scales,

$k_B$ = number of MDS-dimension(s),

$n$ = number of sequences.

If there is no additional information in the MDS-scales, the statistic is F-distributed with $k_B$ degrees of freedom for the nominator and $n-k_A-k_B-1$ degrees of freedom for the denominator.

If the MDS-dimensions contribute significantly to the multiple regression coefficient, they can be used for identifying new aspects of users' problem solving behavior relevant to performance. Otherwise they can at least help toward a better understanding of the already regarded aspects. If it is, for example, possible to rotate the MDS-dimensions in such a way, so that one dimension correlates with the performance measures higher than any of the a-priori scales, then the interpretation of this dimension may provide a more precise conception of the crucial behavioral aspects.

## Interpreting MDS-Dimension(s)

In order to detect yet unregarded user characteristics influencing problem solving performance, the spatial representation has to be analyzed internally (Shepard, 1972, pp. 39). Therefore, it is useful to have a closer look at the user sequences represented as extreme values of the dimensions. Furthermore, it may be helpful to include behavioral sequences in the analysis which have an a-priori known meaning, as for example the solution process of an expert user or a normatively defined optimal solution. The consideration of such a-priori defined sequences could reveal characteristics which discriminate actual users from ideal users and therefore influence performance. In the case of a solution with more than one dimension, it may be useful to rotate the resulting dimensions with reference to such a-priori defined sequences.

To reduce the interpretational arbitrariness of the internal interpretation it may help to investigate the relationship of the dimensional values with the a-priori scales. If there are significant correlations between these scales and the dimensional values, they can be used to validate the results of the internal analysis.

## EXAMPLES

The method described above was applied to two different kinds of user sequences. The data were obtained in two different research projects. Upmeyer, Meseke and Enzmann (1988) collected data of users' input sequences while working with a software system. Eyferth, Widowski and Schmid (1991) collected data of programmers' reading sequences while memorizing a program text. Both projects registered users' behavior

in complete log-files (Gaines, 1981) and made these data accessible for the application of our scaling procedure.

## Example 1: Performance Strategies of CAD-Users

The study of Upmeyer et al. (1988) was designed to evaluate the software system AutoCAD which is a computer-aided design system (CAD) widely used for construction in areas such as electronic or mechanical engineering. AutoCAD, as presented in the study, offers a range of over 250 commands and parameters. The system is hierarchically organized in menus with respect to the function of the available commands. Two kinds of input medias, keyboard and mouse, are provided, i.e. commands can be given to the system by typing the command name with the keyboard or by selecting the command with the mouse in a menu.

### Description of the Study

Twentythree undergraduate engineering students of the Technical University Berlin and a CAD expert participated in the study. The students had finished one introductory CAD course before participation. The expert was an engineer with long experience working professionally with AutoCAD.

In the experimental sessions the students' task was to transfer a technical draft from paper to AutoCAD. The draft consisted of three main parts, the profile of a flange, the corresponding plane view, and a legend. The time for solving the task was restricted to 90 minutes. All inputs given by students with keyboard or with mouse were recorded in log-files. Furthermore, the sessions were video recorded.

## Scaling of the Sequences

The segmentation of the log-files into meaningful elements is based upon the set of possible AutoCAD commands. That is, every use of the mouse button and every sequence of keystrokes finished with the RETURN key is considered as a meaningful segment of the sequence.

Altogether 150 different segments were produced at least once by at least one person. These segments constitute the possible states for defining the transition matrices. The number of segments actually produced per sequence varies between 382 and 1196 ($M$ = 747.35, $SD$ = 239.82). Each person used a different subset out of the set of possible states. As a result of multidimensional scaling a two-dimensional solution (coefficient of alienation = .19) has been accepted as an appropriate representation.

## Prediction of Performance

Two scales for completeness of solution are introduced as external measures describing quality of user performance: first, the number of produced draft elements; second, the number of completed subtasks.

To construct the measure number of produced draft elements, the presented drawing was divided in 93 basic units as lines, circles, arcs or hatchings. For the production of each element only one specific command with the accompanying parameters has to be used.

The measure number of completed subtasks relies on a more global partition of the technical draft. A subtask is defined as a cluster of draft elements sharing two features: first, each element of the subtask has to belong to the same of the three main parts of the draft; second, each element of the subtask has to be produced with the same type of specific command.

Utilizing the MDS-dimensions as predictors, the multiple regression coefficient is $R = .72$ ($F(2,22) = 10.23$, $p < .01$) for the number of produced draft elements and $R = .75$ ($F(2,22) = 12.21$, $p < .01$) for the number of completed subtasks as a criterion. The MDS-solutions therefore can be regarded as meaningful predictors of the quality of performance.

## Comparison with A-priori Defined Scales

Four scales were defined a-priori which may reflect some relevant aspects of the input sequences: first, the ratio between the number of mouse and the number of keyboard inputs (keyboard coefficient); second, the number of actually produced segments (number of inputs); third, the number of different commands; and fourth, the frequency of changes between the three main parts of the draft (changes between drawing parts).

The keyboard coefficient is a measure proposed by Shneiderman (1984) as an indicator for expertise. He claims that users preferring the keyboard are usually more experienced than users prefering the mouse. The remaining three scales are newly introduced in this study for the explorative analysis of data.

Utilizing both measures of performance as criteria, the multiple regression coefficients are significant for the a-priori defined scales alone as well as for both, the a-priori scales and MDS-dimensions as predictors. The increment of the multiple correlation coefficient by addition of the MDS-dimension as predictors is significant for both performance measures (table 1).

TABLE 1

**Increment of the Multiple Regression Coefficients by Adding MDS-Dimensions as Predictors (Example 1)**

| | Predictors: a-priori scales only[a] | Predictors: a-priori scales and MDS-Dimensions[b] | F for Increment[c] |
|---|---|---|---|
| Elements | .63* | .76** | 3.91* |
| Subtasks | .62* | .77** | 4.33* |

Note. Elements = number of produced elements;
Subtasks = number of produced subtasks.
N = 25.
[a]df = 4,20. [b]df = 6,18. [c]df = 2,18.
*p<.05. **p<.01.

Therefore, the MDS-scales can be regarded as containing more information with respect to behavioral aspects relevant to performance than the concurring a-priori scales.

## Interpretation

For the interpretation of the MDS-solution we introduced three sequences in the scaling process which have a-priori meaning: first, the sequence of the CAD expert; second, a normative sequence, generated by producing a correct solution with a minimum of inputs; third, the prototypical sequence of the sample, which is the empirically found sequence with minimal distance to all other empirical sequences.

As expected for a mathematically correct MDS-solution, the prototypical sequence is situated in the center of the spatial configuration (figure 1).

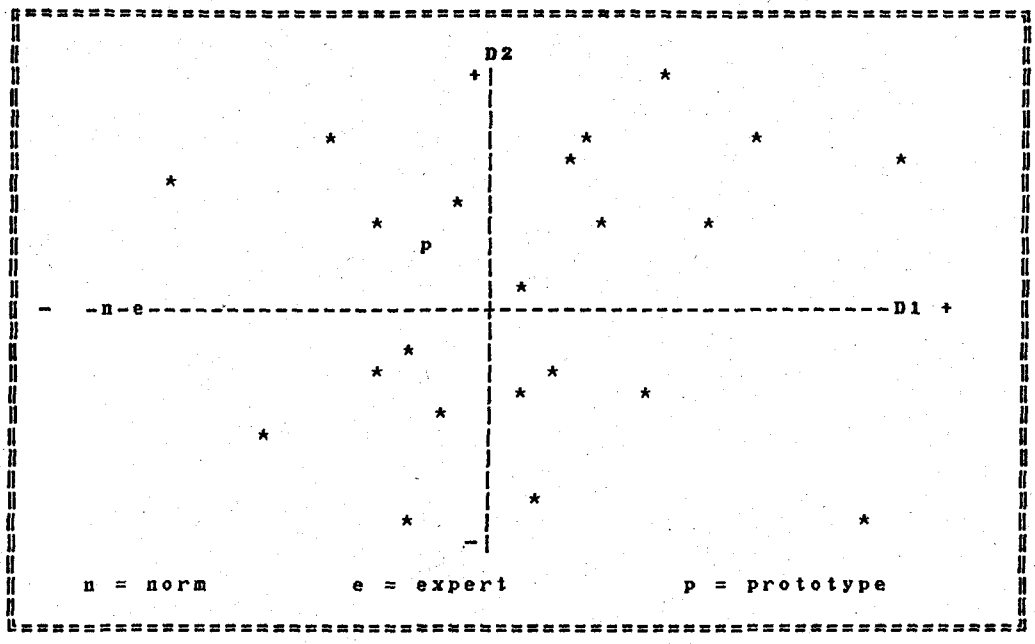


Figure 1., Two-dimensional solution of multidimensionally scaled differences between user sequences (Example 1)

As suggested by the a-priori meaning of the sequences, the expert and the normative sequence are situated close to each other in comparison with the other points. This indicates that the MDS-solution reflects psychologically relevant aspects of the users' problem solving behavior.

The location of the dimensions provided by the scaling procedure is rather favorable for the further interpretation process, because expert and normative sequences constitute the extreme points on the first dimension and the center points on the second dimension. Therefore, dimensions were not rotated. The data material applied for the internal analysis are the log-files as well as the videotapes recorded during the experimental sessions.

The inspection of the video protocolls for the extreme points on the first dimension shows that users situated opposite to expert and normative sequence (positive end of the first dimension) more often cancelled activated commands and browsed in the menus without actually selecting a command. In contrast, users

on the same end as the expert and the normative sequence (negative end of the dimension), mostly selected the appropriate commands at the first trial and seldom browsed through the menus. This suggests that the first dimension reflects users' knowledge about the relevant commands of the system and their proper use.

This interpretation is supported by the pattern of correlations with the a-priori scales (table 2).

TABLE 2

Correlations of MDS-Dimensions with A-priori Defined Scales (Example 1)

| | Dimension 1 | Dimension 2 |
|---|---|---|
| Keyboard coefficient | .55** | .08 |
| Number of Inputs | -.29 | .36 |
| Number of Different Commands | -.44* | .28 |
| Changes between Drawing Parts | -.09 | .44* |

Note. N = 25. df = 23.
*p<.05. **p<.01.

The first dimension correlates significantly with the keyboard coefficient and the number of different commands. Persons situated on the negative side of the first dimension preferably work with the keyboard and use a lot of different commands.

The inspection of the extreme points on the second dimension shows that users at the negative end selected a part of the draft and tried to complete this part before starting a new one. In contrast, users situated at the positive end of the axis started a lot of parts of the drawing and changed between these parts very often. The position of the expert and the normative sequence in the center of this dimension suggests that a compromise between a too persistent and too flexible organization of the drawing process is an effective strategy for mastering the given task. Altogether the

second dimension seems to reflect how users organize their problem solving behavior.

This interpretation is also supported by the correlations with the a-priori defined scales. The second dimension correlates significantly with the changes between the drawing parts (table 2); i.e. persons with high positive values on the second dimension change a lot between the main parts of the draft, whereas persons situated on the negative end of this dimension produced a low number of changes. The marginally significant correlation of the dimension with the number of inputs can be explained in that way that frequent changes between parts of the drawing make additional inputs necessary.

## Example 2: Reading Strategies for Program Memorization

In the second study, Eyferth et al. (1991) investigated the relationship between reading strategies in a program-memorization-task and the quality of reproduction. The program language used in the project is ELAN (Elementary LANguage).

### Description of the Study

Fortyeight undergraduate students of computer science at the Technical University Berlin participated in the experiment. The students had had introductory programming courses with ELAN.

In the experimental sessions they read an ELAN program which was to be reproduced later. The reading time was limited to half an hour. While reading the program the monitor was masked so that only one line could be read at one time (Widowski & Eyferth, 1986).

Students could choose the line they wanted to read next by using the mouse. They could read every line as often as they wished.

As an independent variable two different versions of the ELAN-program were introduced (24 subjects for each version). While both versions fulfill the same function, one is programmed in an applicative and the other in an imperative style. Imperative and applicative are concepts used in computer science for the classification of programming languages. In this context it will only be relevant to notice, that the program of the imperative version is characterized by the use of loops (iteration) and of the definitions of new variables within the program text. In contrast, the applicative version is characterized by the use of subroutines which call themselves (recursion) and by dispensing with the definition of variables within the text.

## Scaling of the Sequences

The elements of the transition matrix are the lines of the programs. The imperative program version consists of 58 lines, the applicative version of 55 lines.

For the imperative version the minimal number of read lines was 201, the maximal number 606 ($M$ = 381.75, $SD$ = 114.8). For the applicative version the minimum was 174, the maximum 506 ($M$ = 356, $SD$ = 90.88). The multidimensional scaling was applied independently to each program version. For both versions, applicative and imperative, the single dimensional solution has been accepted (coefficient of alientation for applicative version = .29; coefficient of alienation for imperative version = .25).

## Prediction of Performance

Widowski constructed two scales to describe the quality of reproduction: first, the number of reproduced tokens; second, the number of reproduced subroutines. A third scale, the score of a program questionnaire, measures program understanding (see Eyferth et al., 1991).

The measure number of reproduced tokens is based upon a segmentation of the programs in basic units (tokens). A token is defined as the smallest unit of the program text which contains semantic information (i.e. keywords of a programming language, variable or constant names, brackets etc.).

The measure number of reproduced subroutines reflects the number of the independently functioning parts of the program, which are correctly coded. Both program versions consist of 7 subroutines.

The programm questionnaire score was obtained by 34 questions about the function of the program and its subroutines. The questionnaire is meant to control whether the subjects had really understood the program or only had committed the representation of the program to their memory without understanding its meaning.

The correlations of these performance measures with the MDS-dimension are significant for both versions (table 3).

TABLE 3

**Correlations of Performance Measures with MDS-Dimensions (Example 2)**

| | Tokens | Subroutines | Questionnaire |
|---|---|---|---|
| Applicative[a] | .57** | .48* | .61** |
| Imperative[a] | .61** | .43* | .42* |

Note. Tokens = number of reproduced tokens;
Subroutines = number of reproduced subroutines;
Questionnaire = program questionnaire score.
Applicative = MDS-dimension of the applicative version;
Imperative = MDS-dimension of the imperative version.

[a] n = 24.

*p<.05. **p<.01.

This suggests that the scale values reflect aspects of the program reading process relevant for the quality of memorization.

## Comparison with A-priori Defined Scales

To investigate determinants of reading which influence program reproduction, Widowski used two a-priori defined scales: first, the number of lines read; second, the number of hark backs (see Eyferth et al., 1991).

The measure number of lines read describes the total amount of read lines whereas number of hark backs reflects the number of times when a subject stopped reading the text top-down and turned back to a line higher above in the program text.

For both program versions the multiple regression coefficients are not significant, only when the a-priori scales are used as predictors (table 4).

TABLE 4

**Increment of the Multiple Regression Coefficients by Adding the MDS-Dimension as Predictor (Example 2)**

| | Predictors: a-priori scales only[a] | Predictors: a-priori scales and MDS-Dimensions[b] | F for Increment[c] |
|---|---|---|---|
| **Applicative version[d]** | | | |
| Tokens | .28 | .65* | 11.72** |
| Subroutines | .20 | .51 | 5.95* |
| Questionnaire | .30 | .66** | 11.93** |
| **Imperative version[d]** | | | |
| Tokens | .44 | .67** | 9.45** |
| Subroutines | .41 | .56 | 4.06 |
| Questionnaire | .39 | .47 | 1.80 |

Note. Tokens = number of reproduced tokens;
Subroutines = number of reproduced subroutines;
Questionnaire = program questionnaire score.

[a]df = 2,21. [b]df = 3,20. [c]df = 1,20. [d]n = 24.

*p<.05. **p<.01.

The multiple regression coefficients with the a-priori scales and the MDS-dimension as predictors are significant for two of the three measures of performance in case of the applicative version and for one performance measure in case of the imperative version. This result is somewhat puzzling because the single correlations between dimension and measures of performance are significant in each case (table 3). For the applicative version this can be explained by the fact that the contribution of the a-priori scales to the multiple correlation is so small that it cannot be compensated in all cases by the contribution of the dimension. For the imperative version the contribution of the a-priori scales is slightly higher, but it seems to be determined by the same behavioral aspects as the contribution of the dimension.

This interpretation is supported by the correlations between the dimensions and the a-priori defined scales (table 5).

TABLE 5

Correlations of the MDS-Dimension with A-priori Defined Scales (Example 2)

| | No. of lines read | No. of hark backs |
|---|---|---|
| Applicative[a] | .07 | -.02 |
| Imperative[a] | .45* | .42* |

Note. Applicative = MDS-dimension of the applicative version; Imperative = MDS-dimension of the imperative version.

[a] n = 24.

*p<.05. **p<.01.

Both correlations are nearly zero for the applicative version but significantly different from zero for the imperative version. Consequently, the increment to the multiple correlation given by the dimension is significant in all cases for the applicative version but only once for the imperative version (table 4). Therefore, in the case of the applicative version the MDS-dimension seems to reflect new behavioral aspects, whereas in the case of the imperative version the dimension reflects similar behavioral aspects as the a-priori scales but it grasps better those features which influence performance. For this reason, an internal interpretation of the MDS-dimension may provide better insights for both versions.

## Interpretation

For both versions the MDS-solution was interpreted by investigating the sequences represented at the end of the dimension. In both versions the dimension reflects different distributions of the amount of reading over

the program parts. Subjects reading more often in complex subroutines represent one extreme of the dimension whereas subjects reading mostly simple routines represent the other extreme. For both versions the correlations of the dimension with the performance measures shows, that spending more effort on the complex routines influences the quality of reproduction.

As outlined above, the two program versions differ with respect to the intercorrelations between the dimension and the a-priori scales (table 5). Whereas the correlations are near zero for the applicative version, there are significant relationsships between the dimension and the number of lines read as well as the number of hark backs for the imperative version. This result suggests, that the different programming concepts require different reading strategies. For the applicative version the only important feature of an effective reading strategy is concentration on the complex subroutines. For the imperative version, on the other hand, it is additionally necessary to refer to earlier parts of the program text and read the lines of the program more often.

This finding is in keeping with the discussion of imperative and applicative programming style by Widowski (see Eyferth et al., 1991). He postulates that understanding imperative programs needs more mental effort, because the program structure requires that relevant information about variables be situated at the beginning of each subroutine. Therefore, the reader has often to go back to obtain the information relevant for the understanding of the program. In applicative programming, on the other hand, the variable concept is abandoned. Therefore, a sequential reading process from the top of the program to the end is facilitated.

## DISCUSSION

We proposed a procedure for the multidimensional scaling of complex behavioral sequences. The purpose of this procedure is the identification of aspects that influence performance. The procedure was exemplarily applied to two different kinds of behavioral data. In both cases, the resulting scales are better predictors for performance than those a-priori defined measures which are additionally considered in these investigations and which also represent the behavioral sequences. In both data sets, internal interpretation of the multidimensional scales has provided a better understanding of the behavioral aspects determining performance. We therefore believe that the application of this method is promising for analyzing the behavior of computer users and it may even prove useful for quite a variety of data.

## References

Card, St.K., Moran R.P., & Newell A. (1983). The Psychology of Human Computer Interaction. Hillsdale, NJ: Erlbaum.

Cohen, J., & Cohen, P. (1983). Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences (Sec. Ed.). Hillsdale, NJ: Lawrence Erlbaum.

Ethier, S.N., & Kurtz, T.G. (1986). Markov Processes - Characterization and Convergence. New York: John Wiley & Sons.

Eyferth, K., Widowski, D., & Schmid, U. (1991). Forschungsinitiativprojekt FIP 2/16 "Kognitive Prozesse beim Lesen und Verstehen von Computerprogrammen". Endbericht. TU Berlin.

Gaines, B.R. (1981). The technology of interaction-dialogue programming rules. International Journal of Man-Maschine Studies, 14 (1), 137-150.

Green, T.R.G., Bellamy, R.K.E., & Parker, J.M. (1987). Parsing and gnisrap: A model of device use. In G.M. Olson, S. Sheppard, & E. Soloway (Eds.), Empirical Studies of Programmers: Second Workshop (pp. 132-146). Norwood, N.J.: Ablex.

Johnson, W.L. (1988). Modelling programmers' intentions. In J. Self (Ed.), Artificial Intelligence and Human Learning (pp. 374-390). London: Chapman & Hall.

Kessler, C.M., & Anderson, J.R. (1989). Learning flow of control: recursive and iterative procedures. In E. Soloway and J.C. Spohrer (Eds.), Studying the Novice Programmer. Hillsdale, NJ: Lawrence Erlbaum.

Moll, T. (1987). On methods of analysis of mental models and the evaluation of interactive computer systems. In M. Frese, E. Ulich & W. Dzida (Eds.), Psychological Issues of Human Computer Interaction in the Work Place (pp. 403-417). Amsterdam: Elsevier Science Publishers.

Polson, P.G., & Kieras D.E. (1984). A formal description of users' knowledge of how to operate a device and user complexity. Behavior Research Methods, Instruments & Computers, 16 (2), 249-255.

Potosnak, K. (1986). Classifying users: A hard look at some controversial issues. In M. Mantei and P. Orbeton (Ed.), Human Factors in Computing Systems (Vol. III, pp. 84-87). Amsterdam: North Holland.

Roskam, E.E. (1972). Multidimensional scaling by metric transformation of data. Nederlands Tijdschrift voor de Psychologie, 27 (9), 486-508.

Shepard, R.N. (1972). A taxonomy of some principal types of data and of multidimensional methods for their analysis. In R.N. Shepard, A.K. Romney and S.B. Nerlove (Eds.), Multidimensional Scaling - Theory and Applications in the Behavioral Sciences (pp.23-47). New York: Seminar Press.

Shepard, R.N., Romney, A.K., & Nerlove, S.B. (1972). Multidimensional Scaling - Theory and Applications in the Behavioral Sciences. New York: Seminar Press.

Shneiderman, B. (1980). Software Psychology: Human Factors in Computer and Information Systems. Cambridge, Massachusetts: Winthrop Publishers Inc.

Shneiderman, B. (1984). The future of interactive systems and the emergence of direct manipulation. In Y. Vassiliou (Ed.), Human Factors and Interactive Computer Systems (pp. 1-27). New York: Ablex Publishing Corp.

Upmeyer, A., Meseke, B., & Enzmann, D. (1988). Teilbericht Projekt F "Computerunterstützte Diagnosesysteme für die Bewertung von CAD-Software". Forschungsbericht der DFG-Forschergruppe "Konstruktionshandeln" (1987-1988). TU Berlin.

Widowski, D., & Eyferth, K. (1986). Comprehending and recalling computer programs of different structural and semantic complexity by experts and novices. In H.P. Willumeit (Ed.), Human Decision Making and Manual Control. North-Holland: Elsevier.

Requests for reprints should be sent to Ute Schmid, who is now at the Department of Psychology, Technical University of Berlin, Dovestr. 1-5, 1000 Berlin 10, Germany.