



## APPLICATIONS OF GENERAL SYSTEMS THEORY TO THE DEVELOPMENT OF AN ADJUSTABLE TUTORIAL SOFTWARE MACHINE

HANS J. VOS

Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands  
[Fax: +31 053 356531]

(Received 19 November 1992; accepted 30 July 1993)

**Abstract**—A fundamental problem with the development of courseware is that no instruments exist which enable the likely performance of courseware to be estimated. For such a purpose, the construction of a model of computer-assisted instruction, such as a qualitative block diagram, is essential. The block diagram is then formalized using general systems theory as a framework. The general systems model is then transformed into a set of cooperating procedures in a computer program, which is documented with examples.

### INTRODUCTION

Even though special instruments have been created to support the development of courseware, such as authoring languages and authoring systems[1], a fundamental problem is encountered during the development of courseware. It is more or less consciously realized by most courseware developers that no special instruments exist by means of which the likely performance of courseware is analysable during design[2-4]. This is a serious problem, for the production of courseware is a complex and costly process. Various sources[5,6] report development ratios of production time to net connect instruction time between 80 and 200.

To meet this need to test the effects of alternative design choices at the development stage, the construction of a model is essential. The instructional design variables of this model can then be manipulated to test their impact on student performance.

As a first step in building a model of the CAI process, a block diagram will be constructed. Subsequently, this block diagram will be transformed into a connected system of adjustable units using General Systems Theory (GST) as a framework. In this framework, the blocks and arrows of the diagram will be replaced by elementary systems and system equations.

This formal model of the courseware system under development provides the developer with a basis for testing the effects of implementing various design options. Finally, the GST model can easily be transformed into a set of cooperating procedures which can be conceived of as an adjustable tutorial software machine.

### GENERAL SYSTEMS THEORY AS A FRAMEWORK FOR MODELLING

The construction of a model is the first step in the development of analytical tools for testing the likely performance of courseware under design. Block diagrams provide a graphic description of a system by identifying the important elements and their relationships for a given problem. Conceptually, the most simple model is a qualitative model or block diagram consisting of blocks and arrows as represented in Fig. 1.

In the figure, the blocks refer to functional parts of the CAI system and the arrows connecting the blocks indicate that there is a relationship of some kind. Tutorial schemes[7-9] can be considered to belong to this class of model.

The teaching material is localized in the subject matter block. The subject matter will be represented by a collection of four elementary instructional frames, in a way to be described later. The CAI-form is considered to be a tutorial under which drill-and-practice can be subsumed.

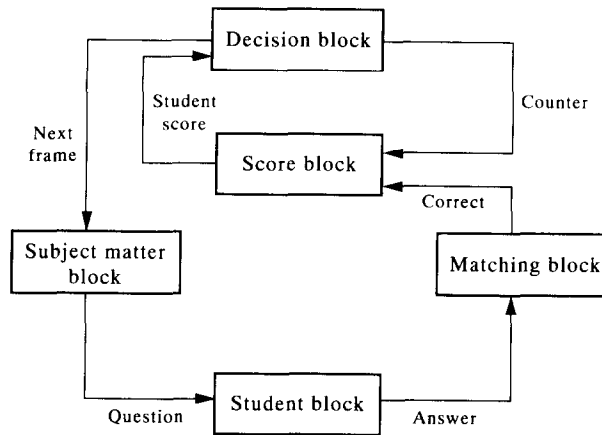


Fig. 1. Block diagram of a system-controlled tutorial CAI system.

Furthermore, it is assumed here that the CAI-form is system-controlled; that is, the computer guides the learner through the instructional material.

Typically, for tutorial CAI a small piece of the subject matter together with one or more questions are presented to the student represented by the student block. The actual answer given by the student to a question is compared with the correct answer stored in the matching block. The result of the matching procedure is sent to the score block.

The score block and the decision block interact with each other by means of a simple feedback mechanism. The past history of the student is collected in the score block, which is used in the decision block to decide how to proceed with the instruction. The next frame to be presented to the student is based on decision rules depending on the student score. These rules reflect part of the teaching strategy and establish the route the student is going to follow through a network of frames. The decision block determines whether or not the student score counter has to be adjusted.

The qualitative model of Fig. 1 is inadequate for the purpose of testing the effects of alternative design choices at the development stage. It lacks the necessary degree of formalization, but it can support the construction of a formal model, which is best suited to describe the dynamic behaviour of complex systems with feedback phenomena and many interactions between components [10–12]. Such a formal model can represent a connected system of adjustable units. The units are derived from a formal description of the blocks in Fig. 1 and they become adjustable by formalizing the relationships.

Once such an adjustable tutorial system has been created, it can be used to bring together the appropriate instructional events. Thus the developer can analyze the likely performance of the courseware under design.

To formalize the block diagram, General Systems Theory is used as a framework for a mathematical description of the complex [13–16]. Here only those concepts will be considered which are needed to formalize the block diagram.

#### SOME GENERAL SYSTEMS THEORY NOTIONS

To classify the blocks of Fig. 1, some elementary systems will be introduced. The first is the well-known black box (see SC in Fig. 2), where only the relationships between the input and the output variables are known.

Next is the black box with memory. In addition to the input and output variables, it possesses state variables (see  $s^{SC}$  in Fig. 2), which are those variables necessary and sufficient to determine the output, together with the input variables. They contain the relevant past history, e.g. the raw score of the student.

The last elementary system to be considered is the decision system (see DC in Fig. 2), which can be met both with and without state variables (see  $s^{DC}$  in Fig. 2). Decisions are taken by means of if-then rules. Thus, the decision-making process becomes an automated procedure.

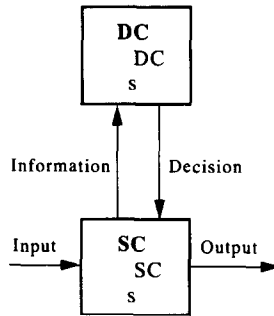


Fig. 2. Interaction between a black box, SC, with state variable  $s^{sc}$  and a decision system, DC, with state variable  $s^{dc}$ .

Elsewhere [17–24] the author has indicated how these decision rules can be made adaptive, by means of an iterative updating following each student's response to a question.

The following symbols will be used to indicate the system variables at time  $t$ :

- $x(t)$  = input variable;
- $y(t)$  = output variable;
- $s(t)$  = state variable;
- $u(t)$  = decision variable;
- $z(t)$  = information variable.

By means of these variables the arrows of Fig. 1 can be classified. To describe the relationships between the arrows, it is necessary to formulate the general form of the two fundamental system equations:

$$y(t) = f[x(t), s(t), u(t)]: \text{output equation};$$

$$s(t+1) = g[x(t), s(t), u(t)]: \text{state equation}.$$

The information and decision equations, the latter standing for the decision rules, are special cases of the output equation; hence, they have the same general form.

#### THE SUBJECT MATTER REPRESENTATION BY A COLLECTION OF FOUR ELEMENTARY FRAMES

An essential role in all four elementary frames is played by the pointers, of which types I–III possess 2 and type IV possesses 4. A pointer refers to the next frame to be presented to the student after his/her response. Types I–III build up a hierarchy, by which is meant that the higher ones incorporate minimally all the properties of the lower ones.

It is assumed that the subject matter is composed of frames and every frame, in turn, of levels. The frames are used to partition the instructional material, for instance on the basis of a task analysis. Every frame is of one of the four elementary types. Frames at the same level contain teaching material of the same difficulty, while frames at higher levels contain more difficult material. In this way every frame can unambiguously be indicated by both a frame and level-number.

The first elementary frame is of type I. It consists of several questions and is specially suited to drill-and-practice. In this type, pointers are distinguished. An up-pointer refers to a frame at a higher level, while a down-pointer refers to a frame at the same or a lower level. However, not every student response activates a pointer and in such a case the next question in the frame is presented to the student. Whether or not a pointer will be activated is decided on the basis of the student-score in the decision block.

In addition to all the properties of type I, the type II allows the presentation of a preceding text at the start of the first question. Analogous to type II, type III combines pure drill-and-practice and a tutorial frame. In addition to the properties of type II, this type of frame allows additional information about the chosen alternative to be given, depending again on the decision block.

The final elementary frame is type IV which is somewhat different from the first three types. It is specially suited for pure tutorial purposes. The essential difference from the other three types

is that a pointer is used to relate every alternative to the next frame to be presented. In the case of a multiple-choice question with four alternatives this means that the '1', '2', '3' and '4' responses are related to the 'A', 'B', 'C' and 'D'-pointers, respectively.

Which frame is actually presented to the student depends for all four types on the past history of the student and the decision rules in the decision block sending steering signals to the subject matter block. This means that the designer has to prepare all the frames containing the instructional material as well as specifying all possible routes which link the frames together. Also, the designer has to specify the decision rules completely. The decision rules can easily be changed by the designer without changing the instructional material.

FORMALIZATION OF THE BLOCK DIAGRAM USING GENERAL SYSTEMS THEORY

Having introduced the elementary frames as the tools to construct the instructional material, the block diagram of Fig. 1 can now be formalized in terms of General Systems Theory. Each block and arrow will be interpreted in terms of both the elementary systems and system variables. The GST model is represented in Fig. 3.

A few more arrows have been included in the GST model to complete the description. The blocks of Fig. 1 can be represented by the following elementary systems:

- subject matter block: black box SC. sub;
- student block: black box SC. stu;
- matching block: decision system DC. mtc;
- score block: black box SC. sco with memory  $s^{sco}$ ;
- decision block: decision system DC. dec with memory  $s^{dec}$ .

The following notation is used for each system variable: the type of the preceding block is a superscript and the type of the succeeding block is a subscript. Furthermore, a system variable can have several components, each indicated by a number. For instance,

$$u_{sco}^{mtc}(2,t)$$

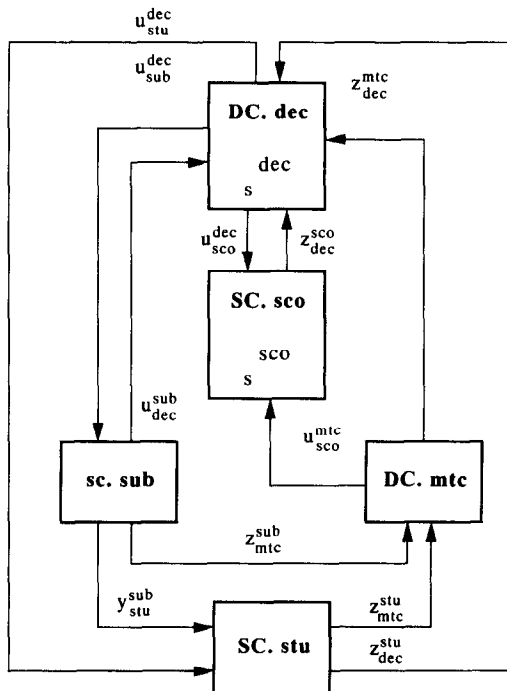


Fig. 3. GST model of a system-controlled tutorial CAI system.

```

PROGRAM GST (Input, Output);
VAR
    Declaration of variables described in Appendix A
    Declaration of procedures described in Appendix B
BEGIN
    initialize (student-decision, student-history)
WHILE NOT student-decision.terminate DO
    BEGIN
        subject-matter (student-decision, block);
        readln (input, answer);
        matching (frame, answer, matching-result);
        score (matching-result, student-decision, student-history);
        decision (student-history, frame, student-decision);
    END
END.

```

Fig. 4. Program for the GST model.

stands for the second component of the decision variable at time  $t$  preceded by the matching block and succeeded by the score block. Listings of the system variables and equations are attached as Appendices A and B with brief explanations.

The cycle through the GST model starts at the subject matter block SC. sub. On the basis of the decision variable

$$\begin{matrix} \text{dec} \\ u(t), \\ \text{sub} \end{matrix}$$

the associated frame, level, and question-number is selected by means of a retrieval mechanism in SC. sub. If it is decided in DC. dec to give additional information, this is also picked up in SC. sub and presented to the student.

Once the system equations have been written down, the GST model can easily be converted into a computer program. The main program can be written in Pascal by developing procedures in a modular way for each elementary system. The input variables that appear are local variables in the procedure. The output, decision, and information variables, however, are global variables. The system equations are used in the procedure body. Before running the main program, it is necessary to initialize the components of some system variables. The main program is shown in Fig. 4.

This computer program can be conceived of as the adjustable software machine introduced earlier. To give a flavour of how this conversion can be done, a simple example is provided in Appendix C with a brief explanation of how flexible courseware can be developed from subsets of this set of cooperating procedures. Only the system equations belonging to the score block and those system equations belonging to the decision block referring to the frames of types I-III are converted into flexible courseware. The complete program GST is available upon request from the author.

## DISCUSSION

It is possible to define unambiguously an abstract representation of tutorial CAI and to develop an execution mechanism for courseware that is both manipulable and based upon a modular and explicit mathematical model. This model, if integrated into a courseware development environment, may be used in connection with analytical tools to trace the likely effects of various design options, such as the decision rules for steering the instructional process in system-controlled CAI.

As far as this work is concerned, only the choices and effects of decision rules in system-controlled CAI are manipulable and traceable. The model is rather simple, though it may seem complicated. To cover tutorial processes in full would require a considerable extension of the present model and a still more general approach to the construction of system cells[25].

Further extensions can be expected from studying the context of learner-controlled CAI, where

learners select their own routes through the instructional material, and a means to represent the individual behaviour of the learner throughout the instructional process [26,27]. These authors have shown that the elementary systems used in the construction of the GST model (Fig. 3) can be considered to be situated in the nodes of a lattice of models of the instructional process.

*Acknowledgements*—The author is indebted to J. Kingma for his assistance in programming the example from Appendix C. The computer program GST is available upon request from the author.

## REFERENCES

1. Barker P. and Singh R., Authoring languages for computer-based learning. *Br. J. Educl Technol.* **13**, 167–196 (1982).
2. Bork A., *Learning with Computers*. Digital Press, Bedford, Mass. (1981).
3. Bork A., Producing computer-based learning materials at the Educational Technology Center. *J. Computer-Based-Instruction* **11**, 78–81 (1984).
4. Kontos G., Instructional computing: in search of better methods for the production of CAI lessons. *J. Educl Technol. Syst.* **13**, 121–133 (1985).
5. Kearsley G., Authoring systems in computer-based education. *Commun. ACM* **4**, 429–437 (1982).
6. Moonen J. and Gastkemper F., *Computer Directed Education*. Het Spectrum, Utrecht, The Netherlands (1983).
7. Hartley J. and Sleeman D., Towards more intelligent teaching systems. *Int. J. Man–Machine Stud.* **5**, 215–236 (1975).
8. Stolorow L., Models for instructional design: a systems approach to instruction. In *Instructional Design* (Edited by Merrill M. D.), pp. 81–91. Prentice Hall, Englewood Cliffs (1971).
9. Wagner W., Design considerations for instructional computing programs. *J. Educl Technol. Systems* **10**, 61–73 (1982).
10. Churchman C. W., *The Systems Approach*. Wiley, New York (1968).
11. Klir G. J., *Interscience, Trends in General Systems Theory*. Wiley, New York (1972).
12. Romiszowski A. J., *Designing Instructional Systems: Decision Making in Course Planning and Curriculum Design*. Kogan Page, London (1981).
13. Banathy B. B., *Developing a Systems View of Education: The System Model Approach*. Fearon, Belmont, Calif. (1983).
14. Banathy B. B., Instructional systems design. In *Instructional Technology: Foundations* (Edited by R. M. Gagné), Chap. III. Lawrence Erlbaum, London (1987).
15. von Bertalanffy L. *General Systems Theory*, New York (1968).
16. Mesarovic M. D., Macko D. and Takahara, J., *Theory of Hierarchical Multilevel Systems*. Academic Press, London (1970).
17. van der Linden W. J. and Vos H. J., Optimal rules for test use in individualized study systems. In *Modern Methods for Test Construction and Use* (Edited by W. J. van der Linden), pp. 104–113. Swets and Zeitlinger, Lisse, The Netherlands (1986).
18. Vos H. J., The use of decision theory in the Minnesota Adaptive Instructional System. *J. Computer-Based Instruction* **15**, 65–71 (1988).
19. Vos H. J., The design of adaptive instructional systems for concept learning using decision theory. In *Research on Instruction: Design and Effects* (Edited by Dijkstra S., van Hout-Wolters B. H. A. M. and van der Sijde P. C.), pp. 141–154. Educational Technology Publications, New York (1989).
20. Vos H. J., Simultaneous optimization of decisions using a linear utility function. *J. Educl Statistics* **15**, 309–340 (1990).
21. Vos H. J., Simultaneous optimization of the aptitude treatment interaction decision problem with mastery scores. In *Objective Measurement: Theory into Practice* (Edited by Wilson M.), pp. 313–331. Ablex, Norwood, N.J. (1991).
22. Vos H. J., Simultaneous optimization of classification decisions followed by a mastery decision. Enschede, The Netherlands. In *Proceedings of the European Conference on Educational Research* (Edited by Plomp T., Pieters J. M. and Feteris A.), Vol. 2, pp. 725–727. Enschede, The Netherlands (1992).
23. Vos H. J., Simultaneous optimization of selection-placement decisions followed by a mastery decision. In *Computers in Psychology: Tools for Experimental and Applied Psychology* (Edited by Maarse F. J.), Vol. 4, pp. 235–244. Swets & Zeitlinger, Lisse, The Netherlands (1993).
24. Vos H. J. and van der Linden W. J., Designing optimal rules for instructional decision making in CAI systems. In *Developments in Educational Software and Courseware* (Edited by Moonen J. and Plomp T.), pp. 291–298. Pergamon Press, Oxford (1987).
25. Mendel J. M. and Fu K. S., *Adaptive Learning and Pattern Recognition Systems: Theory and Applications*. Academic Press, New York (1970).
26. De Diana I. P. F. and Vos H. J., A lattice representational definition of a hierarchy of instructional processors usable in educational courseware. *Computers Educ.* **12**, 427–434 (1988).
27. Vos H. J. and De Diana I. P. F., A lattice representational model of an instructional processor. In *Developments in Educational Software and Courseware* (Edited by Moonen J. and Plomp T.), pp. 283–291. Pergamon Press, Oxford (1987).

## APPENDIX A

### *System Variables of the GST Model*

A description of the output and information variables leaving the subject matter block:

$$y(1,t) = \text{presented question and its accompanying alternatives, possibly preceded by a text in case of type II–IV;}$$

$$y(2,t) = \text{possibly presented additional information in case of type III and IV;}$$

$z_{mic}^{sub}(1,t)$  = number of correct alternative (1, 2, 3 or 4);

$z_{mic}^{sub}(1+i,t)$  = whether or not alternative information belongs to the  $i$ th alternative (boolean variable);  $1 \leq i \leq 3$

$z_{dec}^{sub}(1,t)$  = whether or not selected frame belongs to type IV (boolean variable).

For types I–III:

$z_{dec}^{sub}(2,t)$  = number of questions in selected frame;

$z_{dec}^{sub}(3,t)$  = frame-number assigned to the up-pointer;

$z_{dec}^{sub}(4,t)$  = level-number assigned to the up-pointer;

$z_{dec}^{sub}(5,t)$  = frame-number assigned to the down-pointer;

$z_{dec}^{sub}(6,t)$  = level-number assigned to the down-pointer.

For type IV:

$z_{dec}^{sub}(7,t)$  until  $z_{dec}^{sub}(10,t)$  = frame-numbers assigned to the 'A', 'B', 'C' and 'D'-pointers, respectively;

$z_{dec}^{sub}(11,t)$  until  $z_{dec}^{sub}(14,t)$  = level-numbers assigned to the 'A', 'B', 'C' and 'D'-pointers, respectively.

Information variables leaving the student block:

$z_{mic}^{stu}(1,t) = z_{dec}^{stu}(1,t)$  = student response (1, 2, 3 or 4).

Decision and information variables leaving the matching block:

$u_{sco}^{mic}(1,t)$  = whether or not the student response is correct (boolean variable);

$z_{dec}^{mic}(1,t)$  = whether or not additional information belongs to the student answer (boolean variable).

Components of the information variable  $z_{dec}^{sco}(t)$  leaving the score block coincide with components of the state variable  $s^{sco}$ :

$z_{dec}^{sco}(1,t) = s_{dec}^{sco}(1,t)$  = number of correct answers in actual frame;

$z_{dec}^{sco}(2,t) = s_{dec}^{sco}(2,t)$  = total number of questions which have been asked until time  $t$ ;

$z_{dec}^{sco}(3,t)[i,j] = s_{dec}^{sco}(3,t)[i,j]$  = number of times the frame with frame-number  $i$  and level-number  $j$  has been visited.

The last block to be discussed is the decision block, where we start with a description of the components of the state variable  $s_{dec}^{sub}(t)$ :

$s_{dec}^{sub}(1,t)$  until  $s_{dec}^{sub}(3,t)$  = frame, level, and question-number of the actual frame, respectively.

The first component of the decision variable  $u_{sub}^{dec}(t)$  is declared as a defined type in Pascal:

$u_{sub}^{dec}(1,t)$  = switch = (down, (\*decrease level or stay at the same level\*); up, (\*increase level\*); next, (\*present next question in the frame\*); not-used (\*frame belongs to type IV\*));

$u_{sub}^{dec}(2,t)$  until  $u_{sub}^{dec}(4,t)$  = frame, level, and question-number of the possible next frame to present, respectively;

$u_{sub}^{dec}(5,t)$  = whether or not the instruction has to be terminated because of reaching the maximum level of mastery (boolean variable);

$u_{sub}^{dec}(6,t)$  = whether or not the instruction has to be terminated because of reaching the minimum level of mastery (boolean variable);

$u_{sub}^{dec}(7,t)$  = whether or not the instruction has to be terminated because of exceeding the allowed time (boolean variable);

$u_{sub}^{dec}(8,t)$  = whether or not the instruction has to be terminated because of reaching the maximum level of mastery, reaching the minimum level of mastery, or exceeding the allowed time (boolean variable);

$u_{sub}^{dec}(9,t)$  = additional information is presented (boolean variable);

$u_{sub}^{dec}(10,t)$  until  $u_{sub}^{dec}(13,t)$  = the frame, level, question, and alternative-number of the possibly additional information.

The components of the decision variable with destination the score block:

$u_{sco}^{dec}(1,t)$ ,  $u_{sco}^{dec}(2,t)$ ,  $u_{sco}^{dec}(3,t)$  and  $u_{sco}^{dec}(4,t)$  coincide with

$u_{sub}^{dec}(1,t)$ ,  $u_{sub}^{dec}(2,t)$ ,  $u_{sub}^{dec}(3,t)$  and  $u_{sub}^{dec}(9,t)$ , respectively.

Component of the decision variable with destination the student block:

$u_{stu}^{dec}(1,t)$  = presented text in case of stopping.

Finally, the subject matter block is reached again and one complete cycle through the GST-model has been accomplished.

If the decision rules have not decided to stop, i.e.  $u_{sub}^{dec}(8,t) = \text{false}$ , this is the starting point for a new cycle.

## APPENDIX B

### *System Equations of the GST Model*

To illustrate the use of the system equations, some of them will be discussed now. Doing so, we start with a description of the equations belonging to the matching block:

$$u_{sco}^{mtc}(1,t) = \text{true, if } z_{mtc}^{stu}(1,t) = z_{mtc}^{sub}(1,t); \quad (1.1)$$

$$z_{dec}^{mtc}(1,t) = \text{true, if } z_{mtc}^{sub}(1+i,t) = \text{true and } 1+i = z_{mtc}^{stu}(1,t). \quad (1 \leq i \leq 3) \quad (1.2)$$

The next block to be discussed is the score block, starting with the state equations:

$$\begin{aligned} s_{sco}(1,t+1) &= s_{sco}(1,t) + 1, \text{ if } u_{sco}^{dec}(1,t) = \text{next and } u_{sco}^{mtc}(1,t) = \text{true}; \\ &= s_{sco}(1,t), \text{ if } u_{sco}^{dec}(1,t) = \text{next and } u_{sco}^{mtc}(1,t) = \text{false}; \\ &= 1, \text{ if } u_{sco}^{dec}(1,t) = \text{up, down, or not-used and } u_{sco}^{mtc}(1,t) = \text{true}; \\ &= 0, \text{ if } u_{sco}^{dec}(1,t) = \text{up, down, or not-used and } u_{sco}^{mtc}(1,t) = \text{false}; \end{aligned} \quad (2.1)$$

$$s_{sco}(2,t+1) = s_{sco}(2,t) + 1; \quad (2.2)$$

$$s_{sco}(3,t+1)[i,j] = s_{sco}(3,t)[i,j] + 1, \text{ if } u_{sco}^{dec}(2,t) = i, u_{sco}^{dec}(3,t) = j,$$

$$u_{sco}^{dec}(4,t) = \text{false, and}$$

$$u_{sco}^{dec}(1,t) = \text{up, down, or not-used};$$

$$= s_{sco}(3,t)[i,j], \text{ if } u_{sco}^{dec}(1,t) = \text{next, } u_{sco}^{dec}(2,t) = i,$$

$$u_{sco}^{dec}(3,t) = j, \text{ and } u_{sco}^{dec}(4,t) = \text{false};$$



$$\begin{aligned}
 &= 1, \text{ if } u_{sco}^{dec}(2,t) = i, u_{sco}^{dec}(3,t) = j, u_{sco}^{dec}(4,t) = \text{true, and} \\
 u_{sco}^{dec}(1,t) &= \text{up, down, or not-used;} \\
 &= 0, \text{ if } u_{sco}^{dec}(1,t) = \text{next, } u_{sco}^{dec}(2,t) = i, \\
 u_{sco}^{dec}(3,t) &= j, \text{ and } u_{sco}^{dec}(4,t) = \text{true.} \tag{2.3}
 \end{aligned}$$

The three information equations for  $z_{dec}^{sco}(1,t)$ ,  $z_{dec}^{sco}(2,t)$ , and  $z_{dec}^{sco}(3,t)$  [i,j] coincide with the state equations for  $s_{sco}^{sco}(1,t)$ ,  $s_{sco}^{sco}(2,t)$ , and  $s_{sco}^{sco}(3,t)$  [i,j], respectively.

The last block to be discussed is the decision block, starting with the decision equations with respect to the subject matter block:

$$\begin{aligned}
 u_{sub}^{dec}(1,t) &= \text{up, if } z_{dec}^{sco}(1,t) \geq \text{round}(0.6 * z_{dec}^{sub}(2,t)) \text{ and } z_{dec}^{sub}(1,t) = \text{false;} \\
 &= \text{down, if } s_{dec}^{dec}(3,t) \geq \text{round}(0.8 * z_{dec}^{sub}(2,t)), \\
 z_{dec}^{sco}(1,t) &\leq \text{round}(0.4 * z_{dec}^{sub}(2,t)), \text{ and } z_{dec}^{sub}(1,t) = \text{false;} \\
 &= \text{not-used, if } z_{dec}^{sub}(1,t) = \text{true;} \\
 &= \text{next, else.} \tag{3.1}
 \end{aligned}$$

In words, the specified decision rules for the types I-III run as follows:

The up-pointer is activated if more than 60% of the questions in the frame have been answered correctly. The down-pointer is activated either if more than 80% of the number of questions in the frame have been answered as well as less than 40% of them have been answered correctly or if all the questions in the frame have been answered. In all other cases, the next question in the frame is presented.

$$u_{sub}^{dec}(2,t) = s_{sub}^{dec}(1,t) \text{ and } u_{sub}^{dec}(3,t) = s_{sub}^{dec}(2,t), \text{ if } u_{sub}^{dec}(1,t) = \text{next;} \tag{3.2}$$

$$u_{sub}^{dec}(2,t) = z_{dec}^{sub}(3,t) \text{ and } u_{sub}^{dec}(3,t) = z_{dec}^{sub}(4,t), \text{ if } u_{sub}^{dec}(1,t) = \text{up;} \tag{3.3}$$

$$u_{sub}^{dec}(2,t) = z_{dec}^{sub}(5,t) \text{ and } u_{sub}^{dec}(3,t) = z_{dec}^{sub}(6,t), \text{ if } u_{sub}^{dec}(1,t) = \text{down;} \tag{3.4}$$

$$\begin{aligned}
 u_{sub}^{dec}(2,t) &= z_{dec}^{sub}(6 + i,t) \text{ and } u_{sub}^{dec}(3,t) = z_{dec}^{sub}(10 + i,t), \text{ if } u_{sub}^{dec}(1,t) = \text{not-used and} \\
 z_{dec}^{stu}(1,t) &= i; (1 \leq i \leq 4). \tag{3.5}
 \end{aligned}$$

$$\begin{aligned}
 u_{sub}^{dec}(4,t) &= s_{sub}^{dec}(3,t) + 1, \text{ if } u_{sub}^{dec}(1,t) = \text{next;} \\
 &= 1, \text{ if } u_{sub}^{dec}(1,t) = \text{up, down, or not-used;} \tag{3.6}
 \end{aligned}$$

$$u_{sub}^{dec}(5,t) = \text{true, if } u_{sub}^{dec}(3,t) = \text{maximum level} + 1; \tag{3.7}$$

$$u_{sub}^{dec}(6,t) = \text{true, if } u_{sub}^{dec}(3,t) = \text{minimum level} - 1; \tag{3.8}$$

$$u_{sub}^{dec}(7,t) = \text{true, if } z_{dec}^{sco}(2,t) = \text{maximum number of allowed questions.} \tag{3.9}$$

The maximum level, minimum level and maximum number of allowed questions in our example were put on 3, 1 and 50, respectively.

$$u_{sub}^{dec}(8,t) = \text{true, if } u_{sub}^{dec}(5,t), u_{sub}^{dec}(6,t), \text{ or } u_{sub}^{dec}(7,t) = \text{true;} \tag{3.10}$$

$$\begin{aligned}
 u_{sub}^{dec}(9,t) &= \text{true, if } z_{dec}^{sco}(3,t)[i,j] = 2, u_{sub}^{dec}(2,t) = i, \\
 u_{sub}^{dec}(3,t) &= j, \text{ and } z_{dec}^{mtc}(1,t) = \text{true.} \tag{3.11}
 \end{aligned}$$

In words, this last decision rule runs as follows: additional information is presented only each second time a student arrives at a frame which actually contains additional information.

$$u_{sub}^{dec}(10,t) = s^{dec}(1,t), \text{ if } u_{sub}^{dec}(9,t) = \text{true}; \quad (3.12)$$

$$u_{sub}^{dec}(11,t) = s^{dec}(2,t), \text{ if } u_{sub}^{dec}(9,t) = \text{true}; \quad (3.13)$$

$$u_{sub}^{dec}(12,t) = s^{dec}(3,t), \text{ if } u_{sub}^{dec}(9,t) = \text{true}; \quad (3.14)$$

$$u_{sub}^{dec}(13,t) = z_{dec}^{stu}(1,t), \text{ if } u_{sub}^{dec}(9,t) = \text{true}. \quad (3.15)$$

The decision equations with respect to the score block for  $u_{sco}^{dec}(1,t)$

$$\text{until } u_{sco}^{dec}(4,t) \text{ coincide with the decision equations for } u_{sub}^{dec}(1,t), u_{sub}^{dec}(2,t), u_{sub}^{dec}(3,t) \text{ and } u_{sub}^{dec}(5,t), \text{ respectively.} \quad (3.16)$$

Decision equations with respect to the student block:

$$\begin{aligned} u_{stu}^{dec}(1,t) &= \text{print ("reached maximum level of mastery"), if } u_{sub}^{dec}(5,t) = \text{true;} \\ &= \text{print ("reached minimum level of mastery"), if } u_{sub}^{dec}(6,t) = \text{true;} \\ &= \text{print ("too long busy"), if } u_{sub}^{dec}(7,t) = \text{true.} \end{aligned} \quad (3.17)$$

Finally, the three state equations of the decision block for  $s^{dec}(1,t)$  until  $s^{dec}(3,t)$  coincide with the decision equations for  $u_{sub}^{dec}(2,t)$  until  $u_{sub}^{dec}(4,t)$ , respectively. (3.18)

## APPENDIX C

### *Example of Developing Flexible Courseware Based Upon the GST Model*

```

program GST (Input, Output);
(*-----
   System variables of the GST-model (see Appendix A)
   -----*)
(*****
* The PASCAL implementation of the system variables defined in Appendix A
* are declared as follows:
*****
* The naming convention used relates closely to that in Appendix A, for instance:
* The system variable  $y_{stu}^{sub}(1,t)$  is referred to in the program by the construct
* "Y[sub_stu_1]", which, in turn, indicates the variable sub_stu_1 in vector Y.
* For each system variable  $x(t)$ ,  $y(t)$ ,  $s(t)$ ,  $u(t)$ , and  $z(t)$  a one-dimensional
* array (i.e., a vector) is declared. These arrays contain all the system
* variables except for the ones that contain score information. The number of
* the latter variables depend on the maximum number of frames and levels,
* MaxFrames (i.e., 40) and MaxLevels (i.e., 3), respectively.
* For instance,  $z_{dec}^{sco}(3,t)[i,j]$  is coded as Zsco[i,j] in the program.
* For the state variables which contain the "memory" of the Machine two
* arrays are declared, namely Spre and Spost containing  $s(t)$  and  $s(t+1)$ .
*  $s^{sco}(s,t)[i,j]$  is coded as Spre.Ssco_3[i,j] and  $s^{sco}(s,t+1)[i,j]$  as Spost.Ssco_3[i,j].
*****)

```

```

(*-----
      Part of the System Equations of the GST-model (see Appendix B)
-----*)
(*****
 * The PASCAL implementation of the system equations specified in Appendix B is
 * given below by means of procedures each containing the equations for one block.
 * The statements in the implementation follow closely the equations defined
 * in Appendix B and are numbered accordingly.
*****)
(*-----
      Score Block
-----*)
procedure SC_sco;
begin
  if U[dec_sco_1] = cNext then
  begin
    if U[mtc_sco_1] = iTrue then
      Spost.s[sco_1] := Spre.s[sco_1]+1
    else
      Spost.s[sco_1] := Spre.s[sco_1]
    end
  end
  else
  if U[dec_sco_4] = iFalse then
    if U[dec_sco_1] = cNext then
      Spost.Ssco_3[U[dec_sco_2], U[dec_sco_3]] :=
        Spre.Ssco_3[U[dec_sco_2], U[dec_sco_3]]
    else
      Spost.Ssco_3[U[dec_sco_2], U[dec_sco_3]] :=
        Spre.Ssco_3[U[dec_sco_2], U[dec_sco_3]]+1
    else
      if U[dec_sco_1] = cNext then
        Spost.Ssco_3[U[dec_sco_2], U[dec_sco_3]] := 0
      else
        Spost.Ssco_3[U[dec_sco_2], U[dec_sco_3]] := 1; (*2.3*)

      Z[sco_dec_1] := Spre.s[sco_1];
      Z[sco_dec_2] := Spre.s[sco_2];
      Zsco[U[dec_sco_2], U[dec_sco_3]] :=
        Spre.Ssco_3[U[dec_sco_2], U[dec_sco_3]]; (*2.4*)
    end (* SC_sco *);

  (*-----
      Part of the Decision Block (Frames of types I until III)
-----*)
procedure DC_dec;
begin
  U[dec_sub_1] := cNext;
  if Z[sub_dec_1] = iFalse then
  begin
    if Z[sco_dec_1] >= Round (0.6 * Z[sub_dec_2]) then
      U[dec_sub_1] := cUp
    else
      if (Spre.S[dec_3] >= Round(0.8 * Z[sub_dec_2])) and
        (Z[sco_dec_1] >= Round (0.4 * Z[sub_dec_2])) then

```

```

    U[dec_sub_1] := cDown
end
else
    U[dec_sub_1] := cNotUsed (*3.1*)

case U[dec_sub_1] of

    cNext: begin
        U[dec_sub_2] := Spre.S[dec_1];
        U[dec_sub_3] := Spre.S[dec_2];
    end; (*3.2*)

    cUp: begin
        U[dec_sub_2] := Z[sub_dec_3];
        U[dec_sub_3] := Z[sub_dec_4];
    end; (*3.3*)

    cDown: begin
        U[dec_sub_2] := Z[sub_dec_5];
        U[dec_sub_3] := Z[sub_dec_6];
    end; (*3.4*)
end (* DC_dec *);

```