A LATTICE REPRESENTATIONAL DEFINITION OF A HIERARCHY OF INSTRUCTIONAL PROCESSORS USABLE IN EDUCATIONAL COURSEWARE

I. P. F. De Diana and H. J. Vos
Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

(Received 22 May 1987; accepted 26 October 1987)

Abstract—The basic "recognize-act-recognize-end" cycle can be recognized in elementary as well as in more advanced forms of CAI. This article attempts to offer a unifying formal framework in which different elaborations of this cycle (embodied in a "processor") can be placed. Three different levels of elaboration are distinguished which can be considered to be situated into the nodes of a lattice of models of the instructional process. A formal definition of such a framework can serve at least two functions. In the first place a uniform and precise definition of various elaborations can be given and new elaborations can be created in a logically funded way. Secondly, such a framework can support the modelling of instructional processes and the stimulation of student behavior. Thus, pre-testing of courseware could become feasible. Aspects of the framework have been used to implement two prototypes of support systems for the development of CAI courseware.

INTRODUCTION AND OBJECTIVES

In Computer Assisted Instruction, as far as the execution of the instructional process is concerned, one can speak of a basic "recognize-act-recognize-end" cycle in which, based upon the instructional situation of the learner, some information is presented, his or her reaction is analyzed and some new information is given. This basic model can already be found in early literature about CAI, for instance Stolurow[1,2] and Merrill[3]. This elementary cycle can also be recognized in more complicated forms of CAI, such as those encountered in Intelligent Computer Assisted Instruction, for instance in O'Shea and Sleeman[4] and in O'Shea et al.[5], albeit in a more complicated form.

The embodiments of this basic cycle functions as an instructional processor, by means of which the instructional process is brought from one process stage into the next. In this article three levels of elaboration of the basic cycle are given. These levels are based upon the extent of sophistication of the decision making process involved in the instructional strategy. A distinction will be made between "instrumental decision making", "normative decision making" and "adaptive normative decision making".

In our opinion there are at least two incentives to try to describe this cycle more formally and to place various elaborations of it into common definitional framework.

In the first place, a precise and uniform description of this basic cycle that underlies most CAI forms could be given. Such a description can elucidate which (types of) relations exist between some less and some more advanced forms of CAI and how they can be transformed into each other. Furthermore, these relations could be transformed into each other. Furthermore, these relations could be formalized as will be shown in this article.

Secondly, such a formal description makes it possible to model various elaborations of this cycle. When this modelling can take place in an environment for the developing of courseware, the door to student-simulation, i.e. simulation of how students with various characteristics would behave in various elaborations of such a cycle, could be opened. This result makes pre-testing of educational courseware feasible; getting information about the quality of courseware while the product is still in the design stage.

Presently, we are developing two lines of research based upon the formal description of the basic instructional cycle. The first one is of a theoretical nature and serves mainly to discern between elaborations of the cycle and new functions therein. This article is a contribution in this first line.

The order of the names of the authors is alphabetical; they are equally responsible for the content.
More complicated elaborations than the ones given in this article are feasible, even though in our opinion the borders of present day CAI technology have been fairly reached. Furthermore, a standardization of functions carried out in the cycle could be realized, enabling a faster design of new types of elaborations and the establishment of a library of standardized system functions.

The second line of research is of an applied nature and is directed at developing prototypes in line with the formal model definitions. De Diana and Vos[6] have described the development of "an adjustable tutorial machine", to be classified in the framework of the present article as "an instrumental decision (strategy) making model". Van Schaik and De Diana[7] report on the development of the Prolog based Engine system, to be classified in this framework as "a normative decision (strategy) making model". This second line of research is directed at developing instruments for pre-testing based upon student-simulation too. De Diana[8] has presented a set of mathematical packages for pre-testing tutorial courseware, connected with the EDUC system, an instrumental decision making system in the line of De Diana and Vos[6]. In van Schaik and De Diana[7] directions are given for developing pre-testing packages connected with the Engine system which is a normative decision making system. Extensions for this system in order to upgrade it into "an adaptive normative decision (strategy) making model" and to connect suitable pre-testing packages to it are under way.

The elaborations given in this article are based upon mathematical techniques; techniques for describing the framework as well as techniques for the realization of functions embodied in the framework. This does not mean that in our opinion only these types of technique are valid ones. Rather they serve an exemplary function.

STEERING THE INSTRUCTIONAL PROCESS USING AN INSTRUMENTAL DECISION MAKER

De Diana and Vos[6] have constructed a block diagram to represent an abstract tutorial system as a first step in the development of an instructional processor based upon an instrumental decision maker. This block diagram was subsequently transformed into an adjustable tutorial processor by making use of the General Systems Theory as a conceptual framework. In this framework, the blocks and arrows of the diagram have been replaced by elementary systems and system equations. Thus, it has been possible to describe unambiguously a model of the instructional process and to develop actually an instructional processor.

The General Systems Theory (G.S.T.)-model of the instructional process was proposed according to Fig. 1.

A verbal description of this model will be given first and after having introduced the necessary concepts of General Systems Theory, an interpretation of the used symbols in Fig. 1 will be given.

The teaching materials are localized in the subject matter block RS.sub and are represented by a collection of instructional frames. Typically, for tutorial CAI a small piece of the teaching materials together with one or more questions are presented to the student represented by the student block RS.stu. The actual answer given by the student to a question can either be correct or false and the value of this binary (boolean) variable is sent to the decision block SC.dec. Which

![Fig. 1. G.S.T.-model of the instructional process (an instrumental decision maker).](image-url)
frame next to present to the student, that is, how to proceed with the instructional process, is decided in the decision block sending steering signals to the subject matter block. This decision is based on a detailed specification of decision rules depending on the student score. These rules reflect part of the teaching strategy and establish the individual route the student is going to follow through the instructional network of frames.

Two elementary systems have been used in Fig. 1 to be able to classify the blocks of the block diagram. The first elementary system to be considered is the well-known black box—in this case the student block RS.stu and the subject matter block RS.sub—is a real system (see RS in Fig. 2) of which only the inputs and outputs are known, the internal structure is not known.

The next elementary system to be considered is the black box with memory or system cell (see SC in Fig. 2); in this case the decision block SC.dec. In addition to the input and output variables this type of elementary system possesses state variables. These variables contain the relevant past history and serve as the memory of the system, for instance the raw score of the student. In the decision block routing decisions are taken by means of decision rules having an “if then” character, which means that certain prescribed actions are taken if certain conditions are fulfilled. Thus, the decision becomes an automated procedure. A decision maker who behaves according to these procedures will be called an instrumental decision maker. Quite often the state variable of the decision maker is not taken into account. The following symbols will be used to indicate the system variables of Fig. 2 at time $t$:

\[
\begin{align*}
x(t) &= \text{input variable;} \\
y(t) &= \text{output variable;} \\
s(t) &= \text{state variable;} \\
u(t) &= \text{decision variable;} \\
z(t) &= \text{information variable.}
\end{align*}
\]

By means of these variables the arrows of the block diagram can be classified. To describe the relations between the arrows, it is necessary to formulate the following two general system equations:

\[
\begin{align*}
y(t) &= f\{x(t), s(t), u(t)\}: \text{output equation;} \\
s(t + 1) &= g\{x(t), s(t), u(t)\}: \text{state equation.}
\end{align*}
\]

The information and decision equations, the later standing for the decision rules, are special cases of the output equation and, therefore, do have the same general form.

The modelling of the production of courseware on the basis of the G.S.T.-model of Fig. 1, means that the designer has to prepare all the instructional frames containing the teaching materials as well as all possible routes which link these frames together. Also, the designer has to give a detailed specification of the decision rules. These decision rules can easily be changed by the designer without changing the teaching materials. It should be stressed that this property makes the CAI-system into a connected system of adjustable units (blocks or elementary systems). The machine, in turn, collects the student data and makes routing decision on the basis of these data.
For some examples of decision rules for steering the instructional process in the context of system controlled CAI, the reader is referred to the report of De Diana and Vos[6]. This report contains also a listing of the system variables and the other system equations.

In the G.S.T.-model, some more notations have been introduced. Above the system variable the block of descent has been marked and beneath the block of destination. A system variable can be composed of several components, each indicated by a number. For instance,

\[ u^{\text{dec}}_{\text{sub}}(2, t) \]

stands for the second component of the decision variable at time \( t \) descent from the decision block and with the subject matter block as a destination.

**MAKING ROUTING DECISIONS BY MEANS OF A NORMATIVE DECISION MAKER**

Decision makers can be represented in many ways by a model. In the following, we will use the normative decision maker as a model. Besides information denoted by an information vector, as in instrumental decision making, the normative decision maker requires a model of the real system. With the help of this model, certain alternatives and their expected results can be analysed. The normative decision maker evaluates these expected results by means of a goal or value function, with the purpose of deciding on an optimum strategy, i.e. the "best" next frame to present.

A normative system is composed of a black box RS.stu, and a normative cell NC. A normative cell consists of a decision cell DC.dec and a system cell SC.mod (see Fig. 3) as a mathematical model of the real system RS.stu. The decision cell comprises a value function \( v^d \) and an algorithm \( A^d \).

The optimization procedure runs as follows: \( A^d \) generations a \( u^{\text{dec}}_{\text{mod}} \). With the help of the model SC.mod and the input variable \( x^{\text{stu}}_{\text{mod}} \) from RS.stu, the expected result \( z^{\text{mod}} \) can be inferred. This \( z^{\text{mod}} \) is evaluated by the value function \( v^d \) of DC.dec. On the basis of this evaluation, either another \( u^{\text{dec}}_{\text{mod}} \) is generated by \( A^d \) or the optimization procedure is terminated. In the latter case, the optimal \( u^{\text{dec}}_{\text{mod}} \), which maximizes \( v^d \), is sent to the subject matter block RS.sub as \( u^d_{\text{dec}} \). On the basis of this decision variable \( u^d_{\text{dec}} \) the next frame is selected by means of a retrieval mechanism and presented to the student.

An instrumental system can be considered as a degeneration of a normative system. The decision-making process is then more or less automatized in the sense that on the basis of the information from the real system RS, a decision is taken immediately by means of a decision rule.
Definition of a hierarchy of instructional processors

Now, the normative system of Fig. 3 can be interpreted in the following way: \( x_{\text{stu}}^{\text{mod}} \) is the \( n \)-dimensional measurement vector of the student's relevant features (characteristics) that are supposed to have some predictive significance. Through a mathematical feature space transformation (FST) the vector \( x_{\text{stu}}^{\text{mod}} \) is transformed in the system cell \( \text{SC.mod} \) into a set of \( r \) condensed parameters, \( z_{\text{dec}}^{\text{mod}} \). These \( r \) parameters are supposed to describe the student (student parameterization) and constitute the student model \( \text{SC.mod} \), for instance: the fraction of correct responses given in the past by a student \( S \) as a measure of his learning ability, the fraction of correct responses given to a question \( Q \) in the past by all past students to all questions, the time \( t \) a student \( S \) has been already in the CAI-system, the number of times he/she has pushed the help-key, and so forth. The algorithm \( A_d \) determines via \( z_{\text{dec}}^{\text{mod}} \) what is the possible next frame \( u_{\text{dec}}^{\text{mod}} \) to present to the student.

The value function \( v^d \) is defined with the help of the so-called linear discriminant functions:

\[
v^d(z_{\text{dec}}^{\text{mod}}, w_i) = w_i^T z_{\text{dec}}^{\text{mod}} = \sum_{k=1}^{r} w_{ik} z_{\text{dec}}^{\text{mod}(k)}
\]

in which \( w_i = (w_{i1}, \ldots, w_{ir}) \) is the weight vector belonging to the \( i \)-th frame. Thus, we associate a unique discriminant function with every routing (assignment) decision. Next, for any decision situation facing how to proceed with the instruction, the normative system selects that decision \( u_{\text{dec}}^{\text{mod}} \), for which the corresponding \( v^d(z_{\text{dec}}^{\text{mod}}, w_j) \) is highest, i.e.

\[
v^d(z_{\text{dec}}^{\text{mod}}, w_i) > \max_{i \neq j} v^d(z_{\text{dec}}^{\text{mod}}, w_j)
\]

In this way a unique routing decision is associated with every pattern of information \( z_{\text{dec}}^{\text{mod}} \). This selected optimal assignment-decision is subsequently sent to the subject matter block as \( u_{\text{dec}}^{\text{mod}} \).

**PATTERN LEARNING RECOGNITION TECHNIQUES APPLIED TO TAKING OPTIMAL ROUTING DECISIONS**

It was assumed that the weight vectors used by a normative decision maker were constant and the same for all students.

However, the performance of the normative decision maker can gradually be improved through machine learning algorithms. This can be done by adjusting (updating) the components of the weight vector \( w_i \). The adaptation process is directed by information feedback derived through an evaluation of past routing decisions. Such learning mechanisms can be dealt with within the

![Fig. 4. An adaptive normative model of the instructional process.](image-url)
framework of General Systems Theory by means of a so-called adaptive normative system[9]. In addition, this type of elementary system disposes of an adaptation block which can be represented by a system cell SC.adt (see Fig. 4). The weight vector w coincides in this case with the information variable

\[ z_{\text{adt}}(i) \]

going to the decision cell of the normative cell. This information variable, in turn, coincides with the state variable

\[ s(i) \]

The learning mechanism proceeds in the following way:

The algorithm \( A^d \) seeks a \( \mu_{\text{mod}} \), so that the corresponding \( z_{\text{mod}} \) maximizes \( v^d \), whereupon this optimal \( \mu_{\text{mod}} \) is transferred to the subject matter block SC.sub as \( \mu_{\text{sub}} \) and the corresponding expected result \( z_{\text{mod}} \)—in this case the expected learning result of the student—to the adaptation block SC.adt. After the next frame has been presented to the student, the real result \( z_{\text{adt}} \) of \( x_{\text{sub}} \), i.e. the response of the selected question was either correct or false, is compared with the expected result \( z_{\text{adt}} \) in SC.adt. If they differ, the value function \( v^d \) can be adjusted via

\[ z_{\text{adt}}(i) \]

The information equation for the variable

\[ z_{\text{dec}}(i) \]

i.e. the adjustment of \( w_i \), which coincides with the state equation of

\[ s_{\text{dec}}(i) \]

can be determined by applying certain iterative learning schemes that are derived from stochastic approximation methods[10]. These methods are very well-known in the field of pattern recognition (PR) techniques[11]. The procedure of Robbins and Monro[12] provides us with the following learning scheme for iteratively determining the optimal weight vectors \( w_i \):

\[
\begin{align*}
\hat{w}_{ji, t+1} &= \hat{w}_{ji, t} + \frac{a_0}{t} z_{\text{adt}}(i, t), & \text{if } z(t) = \text{correct}; \\
\hat{w}_{ji, t+1} &= \hat{w}_{ji, t} - \frac{a_0}{t} z_{\text{adt}}(i, t), & \text{if } z(t) = \text{false};
\end{align*}
\]

(3)

in which \( j = 1, \ldots, r \) and \( a_0 \) is a constant, called the correction increment. At the beginning we set the weight components \( \hat{w}_{ji, 1} \) according to our best initial knowledge about what features are known to be important.

**A LATTICE OF INSTRUCTIONAL PROCESSORS**

Just as an instrumental cell SC.dec can be considered as a degeneration of a normative cell NC, an adaptation cell SC.adt can be considered as a degeneration of a learning cell LC.

The learning cell is a mathematical model of a learning process and disposes of, just as in a normative cell, both an algorithm \( A^l \) and a value function \( v^l \) representing the learning goal. When a learning cell degenerates into an adaptation cell, i.e. an automatization of the learning process, the learning goal and the algorithm of the learning cell are converted into a state equation, mostly by means of stochastic approximation.

By basic configurations, we mean configurations composed of one or more basic elements of a different kind, viz. the black box RS, the normative cell NC, the learning cell LC, and its degenerate versions SC.dec and SC.adt, respectively. A basic configuration dominates another one, if the latter
can be obtained from the former by performing one or more of the following two unary basic operations: degeneration or elimination of one basic element. It can be shown that the partially ordered set of binary relations thus defined constitutes a lattice [9]. This means that the basic configurations can be considered to be situated into the nodes of a lattice, i.e. a lattice of instructional processors which can be represented by Fig. 5. The nodes in the lattice represent the following basic configurations:

- 0: Primary system (RS);
- 1: Adaptive primary system (RS + SC.adt);
- 2: Instrumental system (RS + SC.dec);
- 3: Learning primary system (RS + LC);
- 4: Normative system (RS + NC);
- 5: Adaptive instrumental system (RS + SC.dec + SC.adt);
- 6: Learning instrumental system (RS + SC.dec + LC);
- 7: Adaptive normative system (RS + NC + SC.adt);
- 8: Learning normative system (RS + NC + LC).

An arrow with SC.dec or SC.adt means an elimination of the denoted system cell. An arrow without a marginal note represents a degeneration of one of the basic elements. As can be seen from this lattice, until now we have only used the nodes 2, 4 and 7 for developing an instructional processor.

**CONCLUSIONS**

While the research project is still unfinished, it has yielded already some encouraging results. It has been shown that a model of an instructional process, as can be used in CAI, can be translated into an instructional processor. As this model has been developed based on the General Systems Theory, it proved at least theoretically possible to situate this model into a hierarchy of (potential) instructional processor models, formalized as a lattice. Moreover, it has been indicated that it is possible to switch from one basic model to another by applying two or more unary basic operations.

De Diana and Vos [6] and De Diana [8] have reported on some aspects of the development of an instrumental decision making system, EDUC, that functions as an instructional processor during the execution of courseware. Van Schaik and De Diana [7] have reported on the development of the Engine system, which is a normative decision making system, functioning as an instructional processor during the execution of object-oriented courseware. Thus, it has been demonstrated that the theoretical approach towards the design of instructional processors is translatable into real-life courseware execution mechanism as well.
Another aspect of the potential value of this approach can be traced to the conceptual structure it offers for simulating the likely behavior of students during the instructional process, given a mode of functioning of an instructional processor and the characteristics of the courseware. As such, it can be seen as a promising step towards the development of instruments for analyzing the likely performance of courseware during design.

Further examination of the other models in the hierarchy and their accompanying instructional processors seems a valuable line of research, because the lattice of instructional processors contains more powerful models than the ones used until now.

REFERENCES