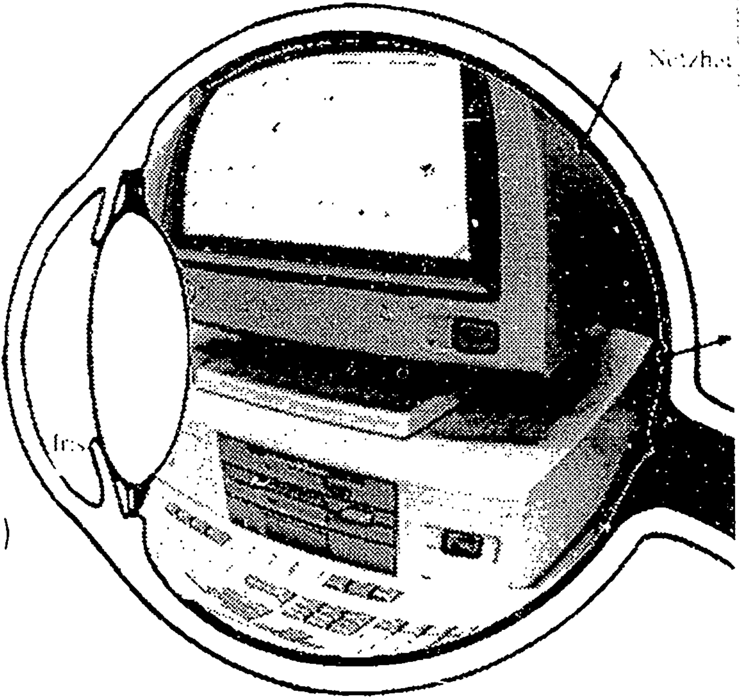
ED 367 305

Principles and tools for instructional visualisation



Jan Schoenmaker
Ivan Stanchev (eds)

IR016579

faculty of
EDUCATIONAL SCIENCE AND TECHNOLOGY

Andersen Consulting - ECC
The Netherlands

University of Twente

ANDERSEN CONSULTING
ARTHUR ANDERSEN & CO. SC

Adaptable Educational Computer Simulations

Willem Jaap Zwart, Albert F.J. Rhemrev, André M. Berloth

Introduction

Using interactive computer simulations, learners can explore a domain by conducting experiments and observing the effects of these interventions. In this way the learner gains insight into the dynamic behavior of a system. Educational computer simulations consist of a mathematical model representing the real system and an educational representation of the system in which the possibilities to intervene and observe are realized (van Schaick Zillesen, 1990). The instructional strategy commonly used with this kind of educational software is 'discovery learning' or 'exploratory learning'.
The use of this kind of computer simulation is generally recognized as having great potential for learning and instruction (e.g. Reigeluth & Schwartz, 1989), because it is expected to motivate, to invite the learner to actively discover the subject matter and to allow for unlimited practice in applying known concepts and in discovering relations. Essentially, computer simulation is a dynamic visualization of a system in the real world. An important design issue in the development of interactive computer simulations is the selection of visualization techniques used in the program. The selected techniques should be adequate representations of the shown phenomena and must be understood and interpreted by the learner. Furthermore, the teacher, in his role as second author of educational software, should be able to change the visualization of the system in order to adapt the simulation to the needs of his students and to increase the correspondence with his other educational materials.

In the Optical DataBase project (ODB-Project, Bestebreurtje & Verhagen, 1992) an attempt is made to develop methods and techniques for the design of reusable multimedia databases. These databases should be suitable for multiple target groups and multiple instructional strategies. Therefore, the teacher is allowed to change the content of the database and the instructional patterns for the learner. The databases are multimedia: text, computer graphics, video, sound and interactive computer simulations are integrated into one system.

The simulations implemented in this project allow for multimedia visualizations as well. An example of this kind of simulations is the renneting simulation (figure 1). The purpose of renneting is the separation of cheese material in the milk (casein-micelles) from the liquid portion (whey). This can be accomplished by the addition of a specific enzyme which causes the casein-micelles to stick together. The process is influenced by several factors including pH of the milk, temperature and amount of rennet and amount of calcium chloride added to the milk. The model is capable of calculating the changes in the milk with time. The calculated entities involve the amount of casein left in the milk, the amount of para-casein, the amount of para-casein that is matted together and the viscosity of the fluid. The last variable is the most important indicator of the progress of renneting. The aforementioned influences make up a complicated system, in which the behavior is not easily predicted.

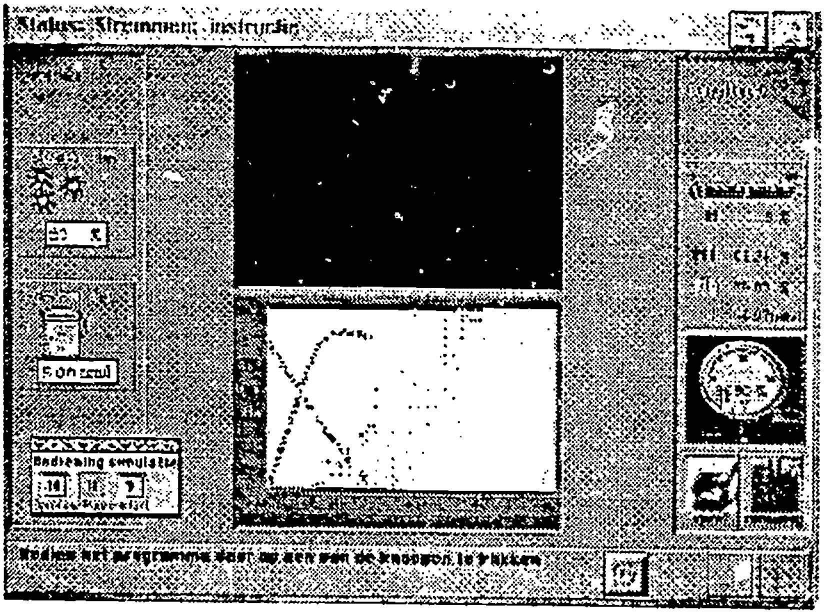


figure 1: the renneting simulation

The following visualization techniques are used in this simulation:
- graphs showing the change in variables
- digits, showing the current value of variables
- video segments used to illustrate the location of the process, the addition of the enzyme and to show tests which indicate the readiness of the renneting
- audio segments played at certain threshold values in variables

For the implementation of the computer simulations in the multimedia databases a flexible design system is developed using object oriented techniques. The working title of the system is OSS: Object-oriented Simulation System. An important part of the system is concerned with the selection of visualization techniques. The system allows for the connection of visualization tools to variables in the model. The selection of which tools to connect to a variable can be done by the teacher, prior to the use of the simulation, or can be done by the learner himself during run-time. This process of selection of visualization techniques can be done manually or can be executed with the help of a reasoning system.
This paper describes OSS in general and the display-manager, the part of the system responsible for the selection of visualization techniques in particular.

Instructional design for computer simulations

Instructional design theories aim at prescriptions for the process of designing instruction. Although educational computer simulations are seen as an important instructional application, instructional design theories for educational computer simulations are scarce. Especially the choice of visualization techniques is a issue that seems to be underexposed. Winer and Vázquez-Abad (1981) try to develop a theoretical framework for educational simulations. They state that educational simulations should "represent the actual variety of the system under study" (p.116). They observe that many simulations do not meet this requirement, thus severely constraining the range of interaction and the 'discovery' value of the simulation. They use the levels of knowledge representation of Bruner (symbolic, iconic and enactive), to describe the levels of interaction that learners should experience. These levels provide the learner "with a rich, integrated framework of knowledge on which to build a complete understanding" (p.116). The prescriptions for the use of visualization techniques are limited to the statement that graphic or pictorial representations can be easily incorporated.

The instructional design theory for educational simulations by Reigeluth and Schwartz (1989) focuses on a classification of educational simulations and the learning processes with these simulations. They distinguish:

- procedural simulations
- process simulation
- causal simulations.

These classes of simulations vary in learning goals and instructional strategies. The level of interaction allowed to the learners changes accordingly. In their view procedural simulations need the highest level of interaction and causal simulations the lowest. This level of interaction is expressed by the number of variables accessible to the learner. No prescriptions on the way of interaction are given.
In the area of learning processes Reigeluth and Schwartz distinguish **acquisition**, **application** and **assessment**. In the phase of application generalization, automatization and utilization are important. All of these processes need adequate visualization, but except general statements on the relation between fidelity and the prior knowledge of the learner, no prescriptions on this are given.

van Schaick Zillesen (1990) gives a general model of educational computer simulations (figure 2)

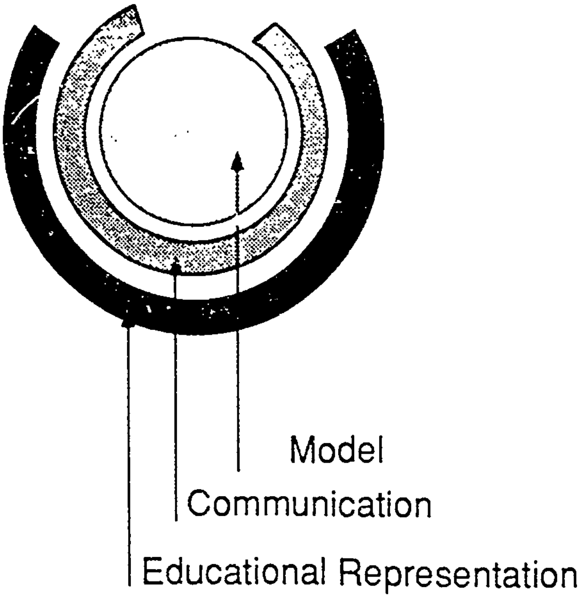


figure 2: general model of educational computer simulations

In this model the following parts are separated:
1 a mathematical model
2 an educational representation of the model
3 an interactive communication layer between components 1 and 2

The interactive communication layer arranges the following design decisions:
- the selection of variables to present to the student
- the way of presenting these variables
- the way the student can interact with the model
- the communication with other educational programs.

Van Schaick Zillesen emphases the division of labor during the development of educational computer simulations. A team of specialists in the area of modeling, didactics, graphical design and educational technology should cooperate in this development process.

He states (1991) that computer simulations have two main advantages above real experiments:
- visualization
- didactics.

Visualization of abstract systems with a set of objects which can be observed and manipulated eases the construction of a mental model of the system. Although it is not clear from his model of educational simulation whether the actual visualization takes place in the educational representation or in the interactive communication layer his separation between model and representation is a useful construct in making computer simulation adaptable in this respect.

In the second generation instructional design model (ID$_2$) by Merrill, Li and Jones (1991) a similar separation of domain knowledge and representation is proposed. The instructional transaction theory assumes that the lack of separation in current authoring tools is the main reason for the high development costs of educational software. They propose a library of 'transaction shells' which are reusable software components suitable for a certain type of educational goal. Every transaction shell consists of an implemented abstract instructional algorithm. This abstract algorithm can turn in to a concrete educational application by filling the transaction shell with:

- **a knowledge base**, a system independent description of the subject matter
- **a resource database**, containing system dependent representations of the subject (e.g. pictures, sounds etc.)

and

- **a database with instructional parameters**, describing for instance the sequence of the subject matter components and the instructional strategy used for these components.

Each database should be created by a specialist. The knowledge base by a subject matter expert, the resource database by media experts and the database with instructional parameters by an educational technologist. For the construction of the knowledge base a syntax is given (Jones, Li & Merrill, 1990) consisting of three types of frames (entities, processes and activities) with four kind of elaborations (attributes, components, hierarchies and associations). Examples of instructional parameters are given for one instructional algorithm. Every transaction shells should however have its own instructional parameters. Furthermore, it should be possible to change the parameters during the delivery of instruction based on student model. Sample instructional parameters in the 'portray knowledge' group are concerned with the representation of the system (Merrill, Li and Jones, 1992):

- **view** (with possible values structural, physical and functional) indicates the principal way the knowledge is represented
- **mode** (with possible values language, symbolic and literal) refers to the nature of a view
- **fidelity** (with as possible values the range [low ..... high]) refers to how close the representation resembles the real thing.

No prescriptions for the content of the resource database are given.

Applied to educational computer simulation and integrated with the model of van Schaick Zillesen the following match can be made (Zwart, 1992). For educational computer simulations:

- the knowledge base should contain a mathematical model of the system
- the resource database should include certain representation techniques to visualize the behavior of the model
- the instructional database should included parameters for the selection of appropriate visualization techniques.

From this it is clear in which part of a flexible educational computer simulation system the selection of visualization techniques should take place.

Collis and Stanchev (1992) pay attention to *visualization*, *interactivity* and *intelligence* as trends in computer based educational simulations. Although their paper addresses these issues in the context of micro-computer based laboratories (MBLs) the trends are applicable to educational computer simulations based on mathematical models as well. With respect to the visualization in computer simulations they observe an increase in quality and quantity of visualizations. The speed of appearance and the graphics in simulation software are improving enormously. Integration of simulation with interactive video is used more and more. They distinguish five major types of visualization:

- still and moving graphics
- sketches and drawings
- digitized photographs
- animations
- moving video.

Collis and Stanchev sketch a cognitive-instructional grid including simple to more complex cognitive skills and different instructional strategies ranging from motivating learners to stimulation higher order thinking. In the mapping of visualization techniques on this grid they assume that "more and moving is better in terms of visualization, ... no matter where one is on the cognitive-understanding grid" and that the more complex the instructional task the more complex visualizations are desirable. These general assumptions are then refined upon by the hypothesis that complex visualizations are particular useful for orientation and motivation (simple tasks) and for higher order thinking (complex tasks), but the more simple visualizations are suitable for enlarging knowledge and problem solving ("middle" tasks).

Min (1992) states that "a complete simulation learning environment consists of a series of elements, the most important of which are:
- highly visualized, graphical output
- highly visualized, conceptual, underlying mathematical models...." (p. 177).

He argues that different kinds of visualization and presentation should be presented to the learner *parallel*. As visualization techniques he names a conceptual scheme of the model, output curves and "video-messages" as feedback. The wide range of different kinds of visualizations range from abstract to concrete. However, no prescription on the selection of these techniques is given.

We can conclude that current instructional design theories give little explicit prescriptions on the selection of visualization techniques in educational computer simulations. We doubt if in the future general applicable prescriptions will appear. Too many characteristics of domain and learner are involved in this selection. The rather limited results of research in the area of interaction between treatment and aptitude directs us towards more flexible learning environments in which design decisions can be made in a later phase of the life cycle of educational software. The best thing to hope for in the near future are hypothesized guidelines, which need to be proofed by experimental research.

Consequently, our system for the development of educational computer simulations is flexible in the choice of visualization techniques.

The system allows for adaptation, both in the preparation of a simulation by a teacher and during the use of the simulation by the student, in:

- which variable to display
- which parameters can be changed
- how parameters and variables are visualized
- what should happen during the simulation run
- which model to use

Set-up of the system

The goal of our Object-oriented Simulation System (OSS) is to ease the development of adaptable interactive educational computer simulations, that can be integrated within multi-media databases. Therefore, OSS has a strong separation of tasks over software components, which can be configured separately. The communication between the components is standardized. Components can be implemented in different programs, using multi-tasking techniques. Although it is not the main development goal of OSS, the combination of a strong separation of tasks and a standardized communication protocol allows for flexible configuration of components into a simulation environment as well.

Our system is divided into three main components:

- the model manager, which handles calculation
- the interaction manager, which handles the interaction and visualization
- the communication manager, responsible for all communication between components.

The whole system is managed by the simulation manager. Furthermore, other programs can communicate with the simulation. For example an instruction manager can change the simulation during run-time or deliver additional domain knowledge. This allows an instructor to interact with the model and the students.

The minimal simulation environment consists of one model manager, one communication manager and one interaction-manager. However, it is also possible to let more models communicate with one interaction-manager, to let more interaction managers display the results of the models or to add other managers to the environment. All the different components talk to each other via the communication-manager: they only need to understand the language the managers use. The separate components of a simulation environment need not to be running on one and the same computer. The communication implementation allows components to communicate over a network. This allows for instance more than one student to work on the same simulation.

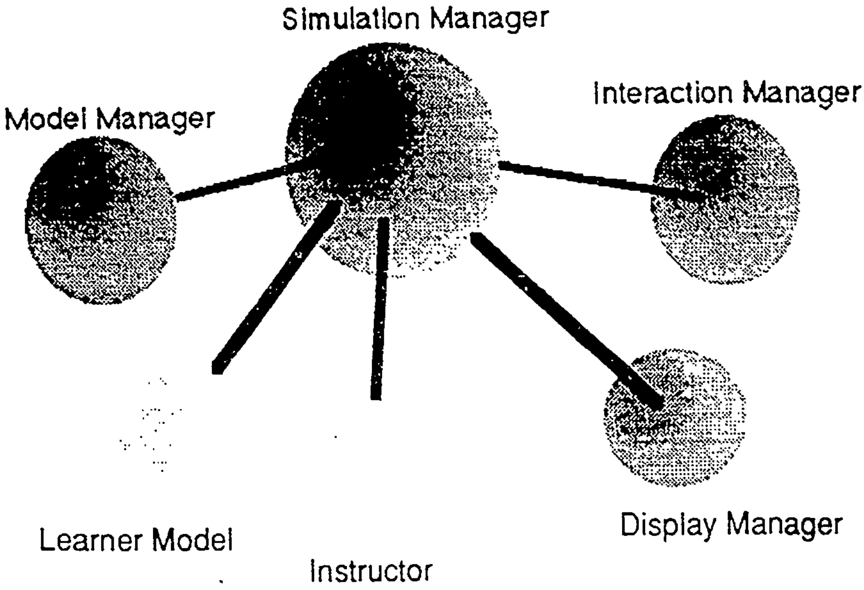


figure 3: general set-up of OSS

The interaction manager is the only manager that communicates with the user. For this purpose the interaction manager has 'interactors'. Each interactor has one input or output function. Every interactor has attributes that can be changed like its location, its color, its name etc.
The system is designed using Object Oriented Techniques and developed on a Apple Macintosh using C++ and MacApp. The communication system is implemented using Apple Events.

An example of non-standard component that can be used is the scenario-manager. This manager adds dynamic to the simulation. This manager makes it possible to let the simulation environment take actions triggered by conditions. Sample conditions included:
- the value of variable
- the change of a variable
- the comparison of two variables

Actions that can be executed included:
- the change of the value of a variable
- the execution of a model command, like Start, Stop and Pause
- the change of an attribute of an interactor

The information the scenariomanager needs is available in the communication protocol. The developer of the simulation simple makes a file with the actions and conditions and the simulation will behave accordingly.

```
# comment: simple sample scenario
if amount > critical_amount    start
while amount < 9               color of graph_1 is "black"
while amount < 9               max of graph_1 is 20
while amount < 9               min of graph_1 is 1
if dAmount changes             continue
if dAmount > 4                 store "phase1"
if pH is not -2                goto "phase1"
if time > 12                   hide clock
if realtime > 15               show help
```

figure 4: sample simulation scenario

The interaction manager is capable of displaying a variable using some 30 kind of interactors, including VLP-stills and running video, audio-feedback, the normal graphs and indicators, animated input etc.

Adapting the representation

The display manager (an extension of the interaction manager) is the component of the system responsible for the selection of visualization techniques.
Its first main task is to add, delete and change interactors on the screen and to connect them to variables in the model. This task is implemented by giving commands, using the communication manager, to other parts of the system.
This mechanism allows the user or teacher to select a specific visualization technique for a specific variable directly.
Besides this direct interface a reasoning mechanism is implemented that can be used to select the most appropriate visualization technique based on characteristics of the domain, the learner and the visualization technique. By using characteristics a more abstract level of selecting visualization techniques is realized. The developer does not need to specify a visualization technique for every variable/learner combination, he just specifies the characteristics. The system is than capable of advising the user in the choice of visualization techniques. Figure 5 outlines the way this is implemented.

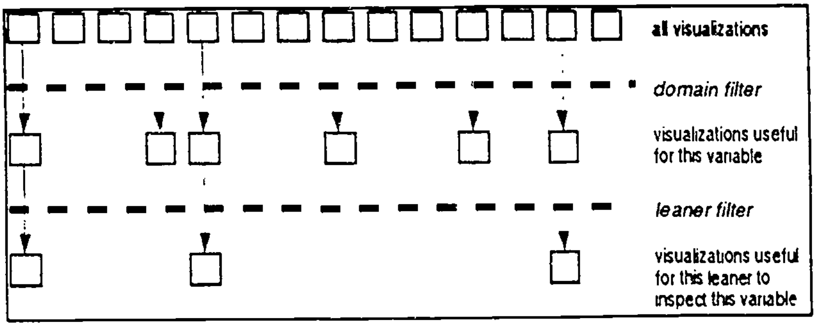


figure 5: outline of selection algorithm

The interactors, learners and variables are all characterized on the following items (Zwart, 1992):

- Output-mode (enactive, iconic, symbolic)
- Order of presentation (zero, first, quantitative)
- Time-dependency (momentous, dynamic)

The output-mode variable is based on the representation levels of knowledge by Bruner. For interactors it indicates the level of representation used as visualization technique. For learners it indicates the level of representation the learner can understand. For variables it indicates the preferred way of representing the variable.

The order of presentation variable is based on research by White & Frederiksen (1989). It describes the form in which relations are shown to the learner. In the zero order presentation only the presence or absence of a phenomenon is shown. In the first order presentation the direction of change in the phenomenon is shown and in the quantitative level the exact value of the variable is shown.

Learners, variables and interactors can have more than one value of a characteristic. For instance learners can be able to interpret both enactive and iconic representations or a variable can be shown momentous and dynamic.

Visualization techniques come in two forms. The first are standard techniques, that do not need additional (domain dependent) resources. Examples of these standard techniques are graphs, bars etc. All that is needed to use these techniques for displaying a certain variable is a connection between the interactor object and the variable. The characteristics of these techniques are set within OSS. The other kind of techniques do need additional resources. Examples are video segments and animations. Every visualization of this kind needs to be labeled separately.

The list of characteristics can be expanded.

Currently the algorithm is binary: a visualization techniques is either suitable or not suitable for this learner/domain combination. However, using the same labeling of visualization techniques, learners and variables with another algorithm a ranking of the appropriate techniques can be realized.

The current implementation of the display manager allows the user to change or add a visualization technique for a variable. Whenever a selection of visualization tools is needed the reasoning system builds a list of appropriate techniques, from which the user can select one item. Figure 6 shows a sample outcome of the selection process.

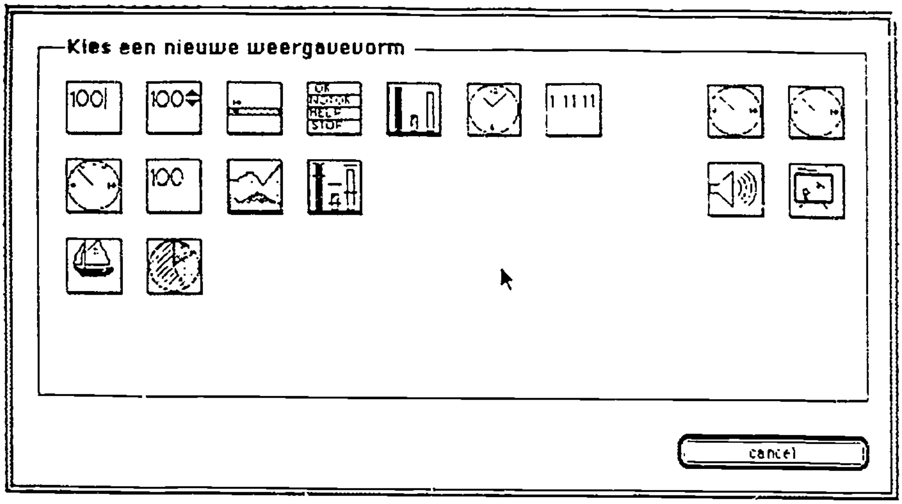


figure 6: Selection window of the display manager

Conclusions

The techniques implemented in the display manager of OSS provide a semi-automatic tool for the selection of visualization techniques. The labeling of learners, visualization techniques and elements of the domain allow the developer of educational computer simulation to specify attributes on a higher level. The characteristics can be seen as instructional parameters as introduced in the Instructional Transaction Theory.
With the absence of a prescriptive theory of visualization this system realizes the flexibility needed for adaptable educational computer simulations.

References

Bestebreurtje, R.B. & Verhagen, P.W. (1992). Developing the concept of an instructional multimedia databse for multiple targetgroups. Paper presented at the AECT-convention "Capture the Vision", Washington D.C.

Collis, B. & Stanchev, I. (1992). Trends and Techniques in Computer-Based Educational Simulations: Applications to MBL Design. Paper presented at the Microcomputer-Based Labs, Educational Research and Standards, NATO, Advanced Research Workshops, 9-13 November 1992, Amsterdam.

Jones, M.K., Li, Z. & Merrill, M.D. (1990). Domain Knowledge Representation for Instructional Analysis. Educational Technology, 29(10), 7-32

Merrill, M.D., Z. Li and M.K. Jones (1991). Second Generation Instructional Design (ID$_2$). Educational Technology, 30(2), 7-14

Merrill, M.D., Z. Li and M.K. Jones (1992). Instructional Transaction Shells: Responsibilities. Methods and Parameters. Educational Technology, 32(2), 5-34

Reigeluth C.M., & Schwartz E. (1989). An instructional theory for the design of computer-based simulations. Journal of Computer-Based Instruction, 16(1), 1-10.

van Schaick Zillesen, P.G. (1990). Methods and techniques for the design of educational computer simulation programs and their validation by means of empirical research. Ph.D. Thesis, University of Twente, Enschede.

van Schaick Zillesen, P.G. (1991). Ontwerpen van Educatieve Computer-simulaties. In: S. Dijkstra, H.P. Krammer & J.M. Pieters (eds.) De onderwijskundig ontwerper. Amsterdam: Swets &Zeitlinger B.V.

White, B.Y. & Frederiksen, J.R. (1989). Causal models as intelligent learning environments for science and engineering education. Applied Artificial Intelligence, 3, 167-190.

Winer, L.R. & Vázquez-Abad, J,. (1991). Towards a theoretical framework for educational simulations. Simulations and Games for Learning, 11(3). 114-119.

Zwart, W.J. (1992). Instructional Transaction Theory applied to Computer Simulations. Paper presented at the European Conference on Educational Research, 22-25 june 1992, Enschede