DOCUMENT RESUME

ED 351 004 IR 015 821

AUTHOR Zillesen, Pieter G. van Schaick
TITLE The Simultaneous Production Model; A Model for the Construction, Testing, Implementation and Revision of Educational Computer Simulation Environments.
PUB DATE [92]
NOTE 17p.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Simulation; *Computer Software Development; Educational Technology; Material Development; *Models; Systems Analysis; Systems Development

ABSTRACT

This paper introduces a hardware and software independent model for producing educational computer simulation environments. The model, which is based on the results of 32 studies of educational computer simulations program production, implies that educational computer simulation environments are specified, constructed, tested, implemented, and revised by teams of experts. The main domains of expertise, within which the simulation environments are formally defined are: curriculum, didactics, system analyses, modelling, software engineering, and graphic design. Each expert specifies, develops, tests, and revises the knowledge components of his/her own domain of expertise. The experts then meet together at discussion sessions to coordinate activities and evaluate the integrated prototype. Important phenomena of the model are: dynamic specifications, the use of fast prototyping techniques, object-oriented coding techniques, and adaptability. Four production stages are described: the initial specification phase, the prototyping cycle, the field test cycle; and the implementation cycle. A table displays the knowledge structures required for the production of educational simulations and four figures illustrate various phases of the production of a simulation environment. (Contains 30 references.) (Author/ALF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

THE SIMULTANEOUS PRODUCTION MODEL;
A Model for the Construction, Testing, Implementation and Revision of Educational Computer Simulation Environments

Pieter G. van Schaick Zillesen
Wageningen Agricultural University
Food and Bioprocess Engineering Group; Dept. of Food Science
P.O. Box 8129; NL 6700 EV Wageningen; The Netherlands

ED 353 904

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it

☐ Minor changes have been made to improve reproduction quality

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Pieter van Schaick Zillesen
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Abstract

A model for producing educational computer simulation environments is introduced. The model implies that educational computer simulation environments are specified, constructed, tested, implemented and revised by teams of experts. The main domains of expertise required for the production of an educational computer simulation environment are: curriculum, didactics, system analyses, modelling, software engineering and graphic design. The production model implies that simulation environments are defined formally in respect to all these domains of expertise. For each domain of expertise the definitions are stored in a separate knowledge component. In principle, each expert specifies, develops, tests and revises the knowledge components of his own domain of expertise. The experts meet at discussion sessions, which are organized at regular intervals. During these sessions the activities of the experts are coordinated and the integrated prototype is evaluated. The model is called 'the simultaneous production model', since the knowledge components of the simulation environment are produced simultaneously. Important phenomena of the model are: dynamic specifications, the use of fast-prototyping techniques, object-oriented coding techniques and adaptability. Four production stages are described: the initial specification phase, the prototyping cycle, the field-test cycle and the implementation cycle. Empirical data about the production of 32 educational computer simulation program have been gathered during previous studies (Van Schaick Zillesen, 1990, 1991a, 1992, Van Schaick Zillesen et al., 1992). The simultaneous production model is based on the results of these studies. The model is described on a hardware and software independent level.

Introduction

The major purpose of an educational computer simulation environment is to provide students with a representation of a part of reality. The students are able to manipulate this representation, e.g. by changing the properties of the representation or by changing the conditions under which the representation operates. The behaviour of the representation as a result of these changes is similar to that of the represented part of reality. Students learn about the real system by transferring the knowledge gained through studying and manipulating the representation of the simulated system that behaves in a way analogous to the real system (Van Schaick Zillesen, 1990). Dynamic simulation enables students to acquire knowledge in an active, individualized way by means of exploration. The importance of this method of knowledge acquisition has been stressed by many authors (e.g., Kolb and Goldman, 1973, Bork, 1981, Romiszowski, 1981 and Wedekind, 1985).

BEST COPY AVAILABLE

ED 353 904

Simulations can be implemented in the curriculum to reach many didactic goals. Some examples of these goals are: enhancement of the students' insight, training problem solving, teaching simulation techniques, training procedures and visualization of abstract concepts (Doerr, 1979; Dekkers & Donatti, 1981; Moore and Thomas, 1983; Hebenstreit, 1988; Van Schaick Zillesen, 1991b; Van Schaick Zillesen et al., 1991, 1992). In this respect much emphasis has been put on the use of simulation programs to facilitate the learning of principles. Often the use of simulation programs is superior to any other teaching method that can be applied for this purpose.

Van Schaick Zillesen and Min (1990) introduced a general design method for the systematic design of educational computer simulation programs. Van Schaick Zillesen (1990, 1991a) tested the design method in respect of four aspects: domain independence, student population independence, performance and effectiveness. Twenty-three educational simulation environments were produced to perform the tests. The prototypes made use of advanced features like windowing man-machine interfaces, high resolution graphical screens and hypergraphics. The design method proved to be domain independent and student population independent. Furthermore, the method allowed for fast prototyping. Many of the simulation environments have been field-tested or are in use in everyday education. The results of the field-tests and the users' experiences both showed that the environments were adequate educational tools. Between 1990 and 1992 the method was applied for the production of a new series of nine simulation environments (published by Van Schaick Zillesen, 1991b, 1992 and by Van Schaick Zillesen et al., 1991, 1992).

Based on the experience gained by producing all (32) simulation environments the following conclusions are drawn:

- The development of an educational computer simulation environment requires expertise in many domains. The main domains are: curriculum, didactics, system analyses, mathematical modelling, software engineering and graphical design. In principle high quality educational computer simulation software can be created by teams in which experts in all these fields participate.
- In each domain other concepts, symbol systems and criteria are used for specifying the simulation environment. Some of examples of symbol systems used in the context of simulation environments are: lists of subjects from the subject matter (used by curriculum specialists), relational diagrams, flow charts (used by system analysts), mathematical formulae (used by modelling specialists), lists of didactic functions (used by specialists in didactics) and program flow diagrams (used by software engineers).
- Most experts are unfamiliar to the concepts and symbol systems originating from other fields but their own.

Communication between the members of the team is often problematic because of this unfamiliarity to each others concepts and symbol systems. Due to this lack of common knowledge within the team there may be obstructions for the adequate realization of a high quality software product.

Some experts may not realize the relevance of the arguments put forward by others. The quality of the final product may be poor due to this undervaluation. Especially in teams with a hierarchical structure this underexploitation of knowledge present in the team may occur.

Furthermore, some experts will not understand the specifications developed by others. The quality of prototypes generated during the process of software development may not be optimal due to such misunderstandings. The costs of the production will increase since more prototypes must be constructed before the final product is obtained.

In this paper a method will be discussed for designing and developing educational computer simulation programs. The problems mentioned above may be overcome, when the method is applied. The paper will deal with the following topics:

- a description of domains of the expertise needed for specifying educational computer simulation programs.
- a description of a model for realizing educational simulations. The main phenomena of the realization model are: the application of fast prototyping techniques, the use of a structured software construction environment and the use of an easy-to-learn, object-oriented, language for software specification.
- A strategy for implementing the production model.

Domains of expertise

The knowledge structures required for the production of educational simulation environments are presented in table 1. For the production of these structures expertise is required in the following domains: curriculum, didactics, system analyses, modelling, graphic design and software engineering. The knowledge structures are mutual dependent; for the creation of most knowledge structures, other knowledge structures are required. Consequently, the production of educational simulation environments can only be done by teams of experts. The activities of the experts in the team are described below. However, in practice the team may consist of less than six experts since individuals may be expert in more than one domain.

CURRICULUM

The production of a new simulation environment starts by a careful analysis of the curriculum in which the final product will be implemented. Based on the results of this analysis, the objectives of the simulation have to be specified in an explicit way. Educational simulations can be used to reach several objectives such as:

- Teaching the principles of the behavior of the simulated system. Alessi and Trollip (1985) used the concept "physical simulations" for this type of educational simulations. Physical simulations facilitate the creation of a mental model of the functioning of the system, based on causal relationships between its entities.
- Training the procedures that are needed to control the system. Alessi and Trollip used the concept "procedural simulations" for this type of educational simulations. Procedural simulations are used to train the control of the system under normal conditions. Furthermore, the diagnosis and correction of disturbances in the system (trouble shooting) can be practiced by means of procedural simulations.
- Teaching the facts in respect to the progression of a system under various conditions. Alessi and Trollip used the concept "process simulations" for this type of simulations. Process simulations are used to enable the students to experience how the system is influenced by changes in the values of its external parameters.

Based on the specified objectives, the boundary between the simulated system and the environment must be specified. This boundary is usually called "the system boundary".

The system boundary is closely related to the learning objectives. For example, the objective of a simulation of a chemical plant introduced in the process engineering curriculum might be the teaching of the principles of the complex interacting chemical reactions within the plant. In this case a physical simulation should be produced. The simulated system consists of the chemical reactions within the plant and the configuration of the plant itself. The system boundary should include this system, as well as all relevant factors, influencing the system and their results (e.g., taps controlling the flow of chemicals

Table 1. KNOWLEDGE STRUCTURES REQUIRED FOR THE PRODUCTION OF EDUCATIONAL SIMULATIONS

knowledge structure	*domain of expertise*	*required knowledge*
simulation objectives	curriculum	course objectives student model
system boundary	curriculum	simulation objectives subject matter
learning method	didactics	student model simulation objectives
didactic functions	didactics	student model system boundary simulation objectives learning method
specifications for resources	didactics	student model didactic functions
formal model	system analyses	system boundary subject matter
mathematical model	modelling	subject matter (relations) formal model
model code	modelling	mathematical model subject matter (data)
graphic resources (on-line)	graphic design	specifications for resources
off-line materials	graphic design	specification for resources
hardware specifications	software engineering	model code didactic functions graphic resources hardware performance
software specifications	software engineering	didactic functions model code graphic resources hardware specifications

through the plant, the input of chemicals and the characteristics of the product).

However, when a simulation of the same chemical plant is to be applied for training the process operators controlling the plant, another simulation environment should be developed. Here a procedural simulation should be produced. This procedural simulation should consist of the task performed by the operators and all relevant factors influencing this task (e.g., alarm messages indicating faults in the plant, reports of reparations and the quality of the product).

DIDACTICS

One of the most important activities during the specification phase of an educational simulation environment is the selection of a learning method to realize the specified objectives. Min(1987) distinguishes five learning methods, which can be applied in the combination with educational computer simulation programs:

- (free) discovery learning. This method can be used for teaching principles. The student is allowed to experiment at his own initiative. This learning method puts severe demands on the students motivation and requires the presence of well developed higher order cognitive strategies.
- learning by doing exercises. The student is instructed to predict the behavior of the simulated system on consequence of future interventions. The predictions are tested by performing the interventions and observing the results. This method can be used for teaching principles.
- guided (coached) (discovery) learning. This method is an extension of the previous one. During the exercises, the computer software generates feedback messages that are presented to the students. The generation of the messages is based on the progress of the student. The application of this learning method in simulation environments is still in the experimental stage.
- problem oriented (discovery) learning. Abnormal behaving systems (simulation cases; e.g., patients in medical simulation) are presented to the students. The student has to establish the cause of the disturbance (diagnosis) based on an analysis of the behavior of the system. This method can be applied for training the procedures, needed to control the simulated system.
- learning by executing (scientific) experiments. When this method is applied, the relation is studied between two entities. The student repeatedly changes the value of one entity while the value of the other is registered. This method is usually applied in combination with process simulations.

After the selection of the learning method, the didactic functions, that should be performed in the simulation environment are listed. Several types of didactic functions may be performed such as:

- visualization (e.g., the presentation of a semi-abstract diagram of the simulated system on the computer screen).
- activation (e.g., the presentation of an exercise to the student).
- experimentation (e.g., enabling the student to test his hypothesis about the behavior of the system by manipulating the entities that make part of the system).
- feedback (e.g., presenting the resulting behavior of the system to the student).
- evaluation (e.g., asking the student to explain the difference between his hypothesis about the systems behavior and the actual behavior of the system).
- explanation (e.g., presenting information to the student to ease the explanation of the observed results).

Besides the didactic functions, the resources that are used in the simulation environment are specified as well. Important resources of a simulation environment are:

- The visualization of the state of the simulated system (e.g., an animation, semi-abstract diagram or a table).
- The presentation of the output of the behavior of the system (e.g., dynamic graphs or animations).
- The control structures of the simulation (e.g., windows, buttons and menubars).
- The instructions used in combination with the simulation (e.g., text or diagrams).
- Information that can be presented to the student (e.g., a help-system based on hypergraphics)
- Feedback messages

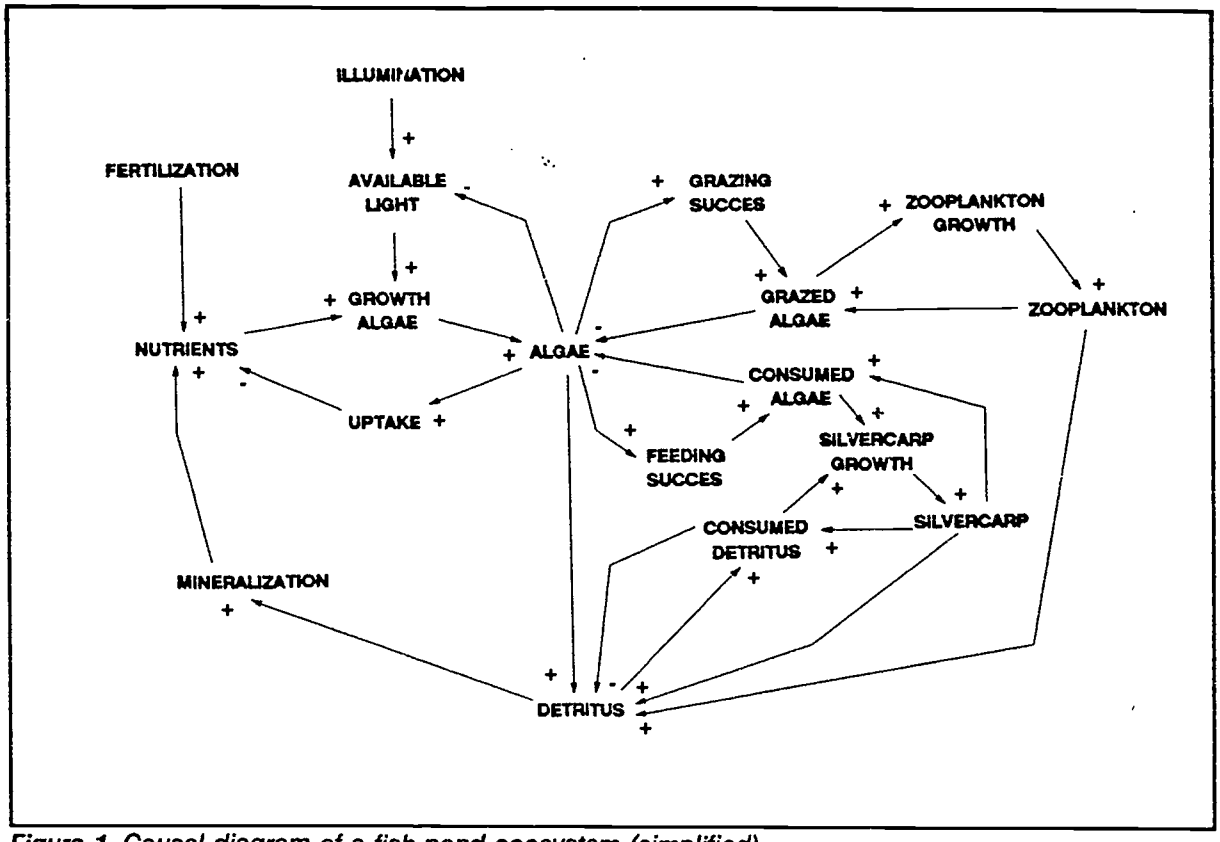

Figure 1. Causal diagram of a fish pond ecosystem (simplified).

SYSTEM ANALYSES

A system has to be described in terms of formal concepts and descriptions of the relations between these concepts to allow computer simulation. A description of a system conforming to these conditions is generally called a formal model (according to Forrester, 1968). In the formal model all relevant objects within the system (called state variables) must be identified and listed. Furthermore, the changes of these state variables in time (called transitions) should be specified in quantitative terms. The structure of the formal model on which an educational simulation is based depends strongly on the specified didactic functionality.

Two types of diagrams are very helpful during the process of system analyses: the causal diagram and the flow diagram. The objective of the construction of these diagrams is to organize and simplify the system boundary specified by the curriculum expert. The diagrams function as an intermediate layer between this concrete description and the abstract computer model that has to be derived by the modelling expert

The main purpose of the creation of a causal diagram (Fig 1) is the identification of feed-back loops. A feed-back loop is a path within the system, coupling cause and effect, returning to the same cause point. The identification of feedback-loops is extremely important for the prediction of the systems' behavior.

Compared with causal diagrams, flow diagrams (Fig 2) describe the system in more elaborately. Forrester (1968) introduced a standard for the construction of flow diagrams. This standard is generally accepted. However, in certain domains more domain-specific diagrams may be used instead of flow diagrams (e.g., analogous diagrams used in electronics).

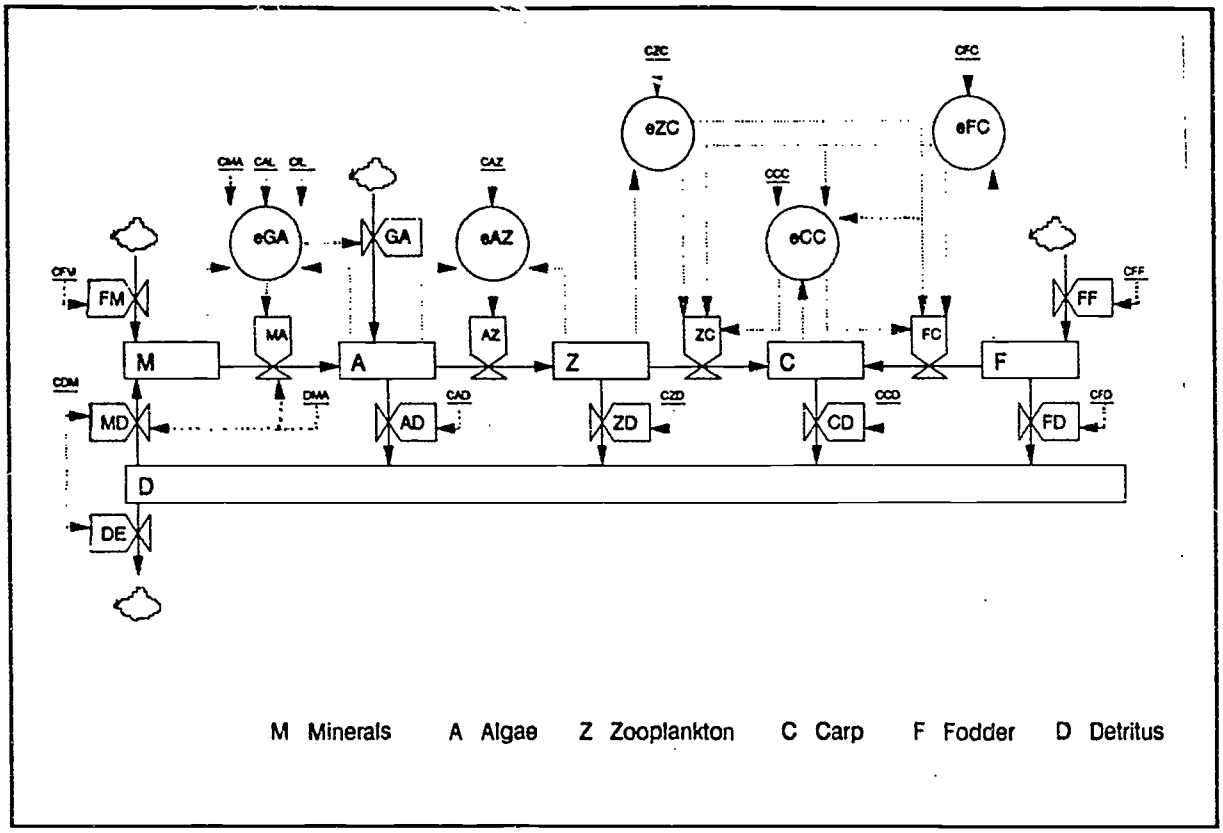

Figure 2. Flow diagram of a fish pond ecosystem (simplified).
The following entities are indicated by means of distinct symbols: state variables (rectangles), material flow between state variables (arrows), source of material flow from outside the system boundary and disappearance of material (cloud symbols), rates controlling the flows (valve symbols), auxiliary equations (circles), external parameters (underlined) and causation (dotted arrows).

MODELLING

The "heart" of a simulation environment is formed by the model code. This model code is the technical representation of the simulated system in the computer. By means of this code, the computer is able to simulate the behavior of the simulated system. The modelling expert derives this code from the formal model specified by the system annalist. Some tasks of the modelling expert are:

- the expression of the formal model in mathematical terms (usually in terms of differential equations). The resulting formulae are called "the mathematical model".
- the selection of a method for approximating the course of the values of the level variables of the mathematical model. On a digital computer this approximation is usually made by means of an iteration process (e.g., the Runge-Kutta algorithm).
- sorting the equations of the mathematical model to eliminate time difference between the level variables at the end of an iteration. Forrester (1968) introduced a method for solving this problem.
- validation of the mathematical model in respect to reliability (does the behavior of the model correspond with that of the original system?). The results of the simulations under various conditions have to be compared with sets of test data, gathered by means of scientific research. Advanced statistical skills are required for this validation.

An example of an elaborate mathematical model of a fish pond ecosystem has been published by Svirhezev et al. (1984).

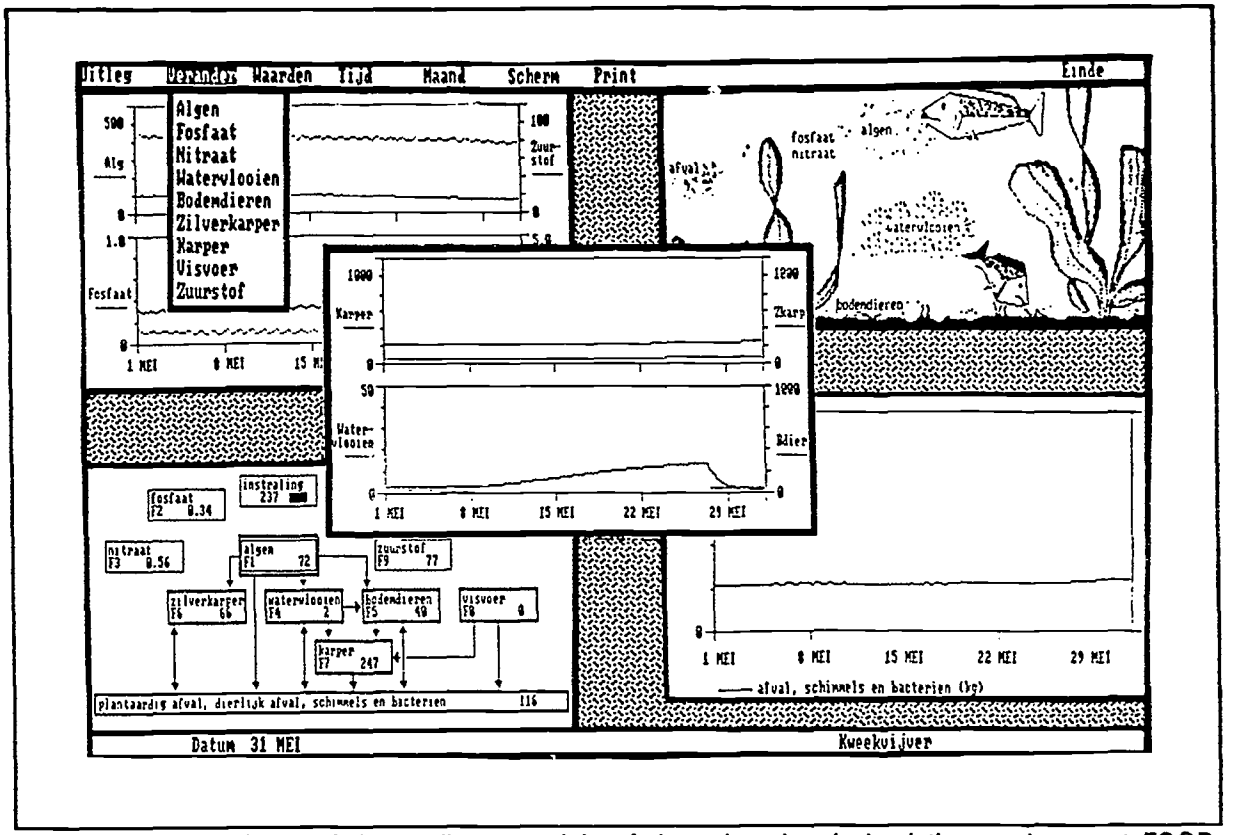

Figure 3. Screen dump of the on-line materials of the educational simulation environment FOOD-CHAIN. In this environment a fish pond ecosystem is simulated.

GRAPHIC DESIGN

In the simulation environment two types of materials are presented to the student: on-line materials (presented on the computer screen) and off-line materials (presented on paper). In Fig. 3 an example of the on-line materials of the educational simulation environment FOOD-CHAIN is presented. The off-line materials of this environment have been published by Hartsuijker et al. (1989). The graphic design of both types of materials is derived from the resources, specified by the didactics specialist. Sometimes draft materials are produced by the didactics specialist to ease the work of the graphic designer. Some examples of materials designed by the graphic designer are:

- lay out of windows and pages of text.
- diagrams.
- pictures.
- model-driven animations.

For the graphic design of educational simulation environments, advanced skills in the field of computer graphics and computer animations may be required.

The on-line materials of an educational computer simulation environment consist of three layers or shells (Van Schaick Zillesen & Min, 1990).

During the simulation sessions the students interact with the outer shell. This shell consists of four parts: a presentation of the state of the simulated system, control structures enabling the student to interact with the simulated system, instructions and feedback. The state of the simulated system and the control structures are presented on a computer screen. Often, the instructions are presented on the computer screen. However, instructions also may be provided on paper. Various forms of feedback may be given (by the teacher, by other students or by means of messages on the computer screen).

The inner shell of the on-line materials consists of three parts: the model code, a formal specification of the didactic functions and the graphic resources.

Between the outer shell and the inner shell an intermediate software layer is present. On the one hand, this layer is decisive for the way information is presented to the student, based on the information stored in the inner shell. On the other hand, the actions of the student controlling the on-line materials are translated to the inner shell by means of this software layer. Examples of actions the student may take are: starting the simulation, manipulating the state of the simulated system and asking for information. Furthermore, the software in the intermediate layer may transfer data between the simulation environment and other computer programs.

The software engineer specifies and constructs the intermediate software layer. Commonl y the construction of this layer is a time-consuming process. The lack of software tools for constructing this layer has been a problem for many years (Wedekind, 1986, already stated that these tools were missing). Consequently, the construction of the intermediate software layer has been an expensive and time-consuming activity in the production process of many educational simulations. Van Schaick Zillesen (1990) showed, that this time can be greatly reduced by using so called design systems for educational simulations. Examples of design systems for educational simulations are: MacTHESIS (Min et al., 1986), THESIS (Van Schaick Zillesen, 1990) and COOP (this paper). Important facilities of these systems are:

- facilities supporting the design of student-controlled software with a parallel structure.
- facilities supporting the construction of the model code or the exchange of data with an external model code.
- facilities supporting the use of resources created by means of general purpose packages.

However, design systems for educational simulations are not yet commercially available or have disappeared from the market. Other types of educational software (e.g., tutorials, drill & practice programs), can be constructed in a fast way using so called authoring systems. Several authoring systems are commercially available. Nowadays, Authorware Professional is the most commonly used authoring system. Davies (1992) recently reviewed this system. However, reviews like the one of Davies show that none of these systems supports all of the facilities mentioned above. For this reason commercially available authoring systems cannot be used to construct the intermediate layer of simulation environments. Consequently the construction of simulation software remains a tedious activity, compared with the construction of other types of educational software.

The simultaneous production model

The knowledge structures required for the production of educational simulation environments are presented in table I. The table indicates, that the production process of educational simulation environments is driven by four external parameters:

- course objectives
- student model
- subject matter
- hardware performance

In many cases an educational simulation environment is used in several courses (e.g., the environment FOOD CHAIN, shown in Figure 3, is in use for lower vocational education, at several levels of secondary education and at a university course). Often, it is necessary to reuse a simulation environment in this way to finance the production costs. However, for each course a separate version of the simulation environment must be produced depending on the course objectives and student population. Van Schaick Zillesen (1990) published some guidelines for adapting the simulation environments based on data gathered by field-testing FOOD CHAIN. Often, an increase of the population of courses in which a simulation environment is used occurs during the first phases of its life-cycle. However, during the last phase of the life-cycle, this population may decrease again. Furthermore, the objectives of a course may change during the life-cycle of a simulation environment. Consequently, course objectives and student model are unstable parameters.

Often, the development of the required mathematical model puts high demands on the costs of the production process of an educational simulation environment (Van Schaick Zillesen, 1990). In many cases, these costs may obstruct the realization of the environment. However, often the costs of the modelling activities can be greatly reduced by adapting already existing models, closely matching the intended simulations, instead of developing new models. Many educational simulation environments (e.g., FOOD CHAIN) are based on models that were originally designed to serve quite other purposes than education (e.g., scientific research, management or as an operator tool). Modelling benefited enormously from the rapid evolution of electronic computers during the past 40 years. Due to this process a gigantic number of calculations can be computed in a limited time. Furthermore, this process stimulated the development of mathematic techniques and methods of system analyses. The number of available mathematical models quickly increases due to this development. Consequently, the number of educational simulation environments, which can be realized at reasonable costs rapidly increases. This means that subject matter, as far as modelling aspects are concerned, is an unstable parameter.

At present, hardware performance is the most unstable parameter of the external parameters that drive the production process of educational simulation environments. A new generation of microcomputers appears every two years, showing a dramatically improved performance. An average courseware production process takes about the same period. Consequently, hardware and software specified at the start of the production process will be obsolete at the moment of implementation or shortly thereafter.

The specification of the didactic functions and the resources are significant activities during the production process of an educational simulation environment. This activity implies the taking of critical design decisions such as:

- the way the simulated system is represented
- the way the results of the simulation are shown
- the way the simulation is controlled
- the way students are instructed during the simulation

However, there is a lack of research-based criteria to take these design decisions. Due to this lack of knowledge, many of these important design decisions can only be taken on a purely ad-hoc (or purely theoretical) basis. Often, results of tests of educational simulation environments show, that some of these decisions should be corrected. Furthermore, revisions will be needed caused by problems in respect to the communication within the team of experts that produces the simulation environment. This problem has already been mentioned in the introduction.

In summary, there are three phenomena obstructing the specification of educational simulation environments:

- The external parameters that drive the production process are not stable. These parameters are: course objectives, student model, subject matter and hardware performance.
- Well founded criteria to take design decisions are lacking.
- Communication between the members of the production team is not optimal due to a lack of common knowledge.

It is impossible to specify a simulation environment completely and correctly at the start of the production process because of these phenomena. Consequently important characteristics of a production model for educational simulation environments must be:

- Dynamic specification. It must be possible to revise the specifications during the course of the production process.
- An Evaluation and Revision cycle. The unknown parameters of the production process can be tuned by means of an iteration process. The importance of this cycle for the development of prototypes of advanced educational software has been stressed by van Schaick Zillesen (1990 and 1991b) and by Moonen (1991).
- Fast prototyping. It must be possible to revise the prototype according to the changed specifications in a limited time. Fast prototyping techniques are required in order to do this.
- Separate development of functional parts. Each expert must specify, develop, test and revise his own part of the simulation environment. For this reason the knowledge components for each domain of expertise must be stored separately. Consequently, each expert is able to use both his own symbol system and the tools he is used to work with. Communication problems within the production team are avoided by this approach.
- Adaptability. The course objectives and student population of a simulation environment is usually not stable. For this reason it is necessary that the product is organized in a way that allows teachers to adapt the product afterwards.
- Object-orientated software development techniques. These techniques facilitate both separated development of functional parts and adaptability.

Recently several models for courseware production have been published. Examples of such models have been described by Koper (1992) and by Courseware Midden Nederland (Courseware Midden Nederland, 1991). However, none of the models published so far allows dynamic specification. Furthermore, in all models a sequential production process takes place. In none of these models much emphasis has been put on the aspect of repetitive software evaluation and revision. Moreover, this activity is usually restricted to the last phase of the production process. Consequently, production models published so far are not suited for the production of educational simulation environments.

The simultaneous production model was designed according to the characteristics mentioned above. Educational simulation environments can be produced in an extremely flexible way, when the model is applied.

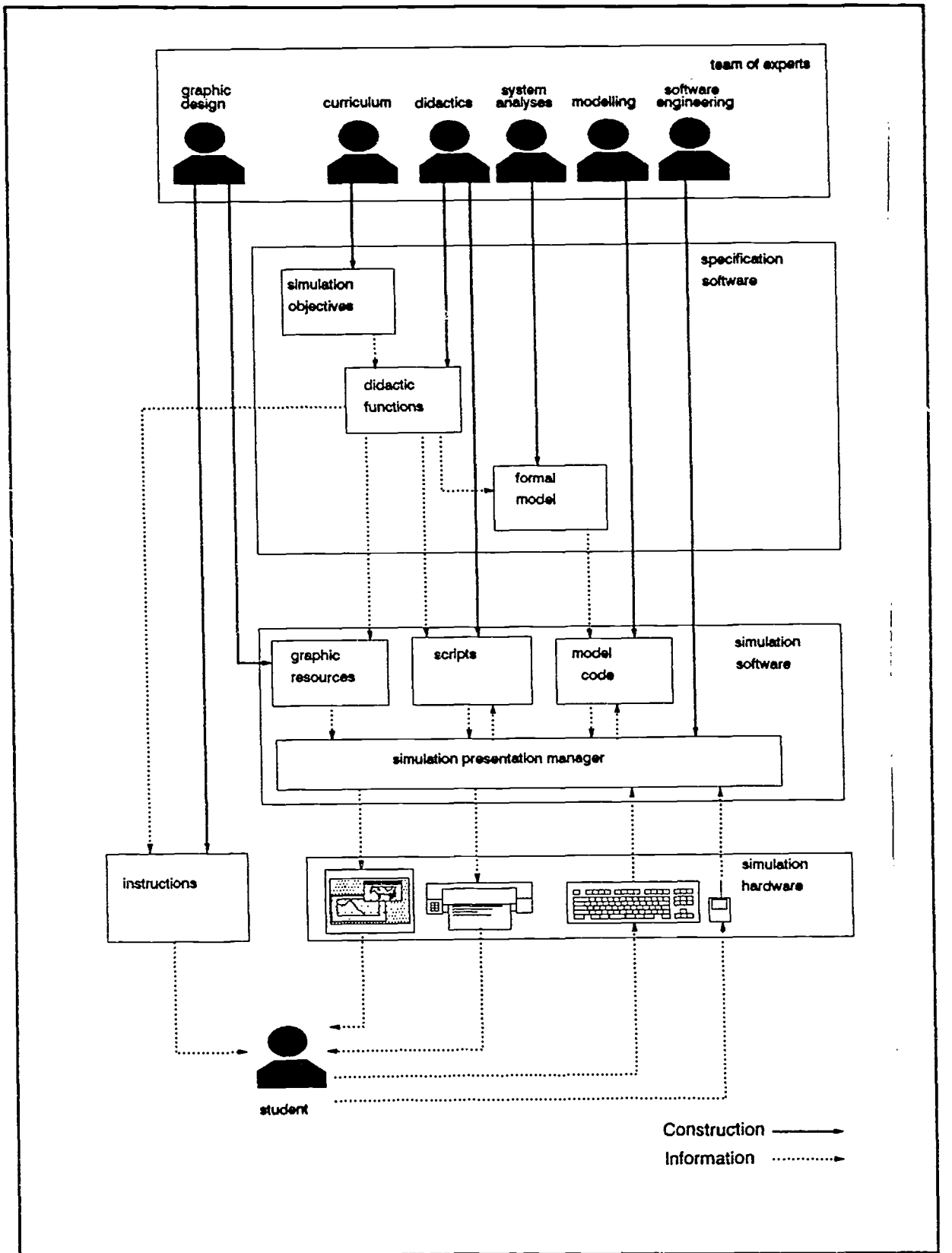

Figure 4. Structure of the production of a simulation environment following the simultaneous production model.

The simultaneous production model consists of three production phases: the Initial Specification phase, the Prototyping, Evaluation and Revision cycle and the Implementation cycle. The stages are completed in a sequential order. Both the structure of the products designed following the simultaneous production model and the production stages are described below.

PRODUCT STRUCTURE

Educational simulation environments, produced following the simultaneous production model consist of four main components:

- simulation hardware
- instructions
- simulation software
- specification software

During the simulation, the students interact with the simulation hardware and the instructions. This part of the product corresponds with the outer shell described in the section 'software engineering' (page 9). Instructions can be provided in two ways: on paper and on the computer screen. The intermediate layer described on page 9 is formed by a program called 'simulation presentation manager'. The simulation presentation manager is re-usable; it is a general computer program, that can be used for the intermediate layer of any educational simulation environment. The inner shell is formed by three subcomponents: graphic resources, scripts and modelcode. The didactic functions are stored in the scripts subcomponent. Each subcomponent is maintained separately. Simulation environments differ in respect to the contents of these subcomponents. The graphic resources and the scripts have a modular structure. Modules from these subcomponents may be shared by several simulation environments. Standard graphical packages are used for the construction of the graphic resources. Any wordprocessor can be used to create the scripts subcomponent. The model code must be coded in a general computer language. Furthermore, in some simulation environments the use of a standard computer language is needed to extend the source of the simulation presentation manager (e.g., when model-driven animation techniques are used).
The specification software is used for the construction of the simulation software. The central part of this component is formed by three design documents: the simulation objectives, didactic functions and formal model. The documents are created by means of general purpose packages such as: wordprocessors, case tools and modelling packages.

INITIAL SPECIFICATION PHASE

At the start of this phase course objectives, that might be reached by means of computer simulation, should be identified. Nowadays, the objectives of new courses are usually described in formal documents. However, this is not always the case in existing courses. When a formal document does not exist, a list of course objectives may be gathered by consulting professors and analyzing the educational materials that are used for teaching. Based on the identified objectives, initial specifications for the system boundary, the didactic functions and the formal model are constructed. The development of a new mathematical model puts high demands on the total costs of the production process of an educational simulation environment. These costs can be greatly reduced by using adapted versions of already existing models in stead of new models. In many cases educational simulation environments can be based on adapted versions of mathematical models that have been originally developed for quite other purposes then education (e.g., scientific research or decision support). The selection of existing models also takes place in the initial specification phase.

PROTOTYPING, EVALUATION AND REVISION CYCLE
(in further text: PER-cycle)

The PER-cycle implies the iteration of three phases:

- **PROTOTYPING.** The construction of prototypes of knowledge structures for each domain of expertise based on the current specifications. The first time the specifications resulting from the initial specification phase are used for this purpose. Each expert specifies, develops and tests the knowledge structure of his own domain of expertise. The experts may use dummy knowledge structures for simulating the knowledge structures from other domains of expertise.
- **EVALUATION.** The evaluation of the complete prototype, consisting of all knowledge structures. The experts evaluate the prototype at discussion sessions, which are organized at regular intervals (in practice: about once a month). The experts must inspect the complete prototype before the start of the session. Parallel to this evaluation an second evaluation takes place. This evaluation is performed by external experts that do not take part in the production process. Two techniques may be used for these evaluations: product review or a checklist procedure. These methods have been discussed by Duchastel (1987).
  When the evaluation method of reviewing the product is applied, each expert evaluates the software, based on his own personal criteria. The results of the evaluation depends strongly on the personal characteristics of the experts.
  When the checklist procedure is applied, the software is evaluated based on a standardized evaluation instrument. The instruments consists of a lists of explicit criteria. Compared with the method of reviewing the product, the methods of executing a checklist procedure depends less on the personal characteristics of the evaluator because the system of standardized criteria puts a limit to this influence. However, most instruments for courseware evaluation published so far are not very useful for the evaluation of simulation environments since not enough emphasis has been put on the specific characteristics of this type of courseware. Nevertheless, Latzina and Wedekind (1986) published a checklist that can be used for this purpose. The PER-cycle is ended and the implementation cycle is started as soon as the results of all of the evaluations indicate, that the prototype is a suitable tool to reach the specified educational goals. Otherwise, the next phase of the PER-cycle is started.
- **REVISION.** In the last phase of the PER-cycle the specifications are revised After The production process enters the production phase as soon as the results of the evaluation phase indicate that the prototype is a suitable tool to reach the specified educational goals. However, if this is not the case, the specifications are revised according to the results of the evaluation. the production process enters the implementation phase.

Fast prototyping techniques are required for an adequate implementation of the PER-cycle. In practice, it must be possible to revise a prototype within a few weeks. Empirical data from previous research (Van Schaick Zillesen, 1990) indicate that 6 - 12 iterations of the PER-cycle are needed, before the implementation cycle can be started.

IMPLEMENTATION CYCLE

Just like the PER-cycle, the implementation cycle implies the iteration of three phases:

- **ADAPTATION.** Several problems may occur when a teacher tries to implement an educational simulation environment in a curriculum. The courseware may use an instructional strategy that does not conform to ,the teacher's educational philosophy. Furthermore, the language used by the courseware may differ from that used in other educational materials (e.g., books) that are used during the same course. Moreover, the courseware may present the simulated system in a way that does not appeal to the students (e.g., the presentation is too abstract or too concrete). In case of a well designed simulation environment the teacher may overcome these problems by adapting the courseware. For this reason, the production model implies the construction of simulation environment that are equipped with a teacher's environment. The concept of a teachers environment has been introduced by van Schaick Zillesen & Min (1987). In this environment the teacher is able to adapt all resources and scripts used by the simulation presentation manager. General purpose packages can be used to implement these adaptations. The importance of general purpose packages for adapting the design of educational software has been stressed by Chandra (1988).
- **USE.** After the completion of the adaptations, the environment is used for everyday education.
- **EVALUATION.** At the end of the course the environment is evaluated. During the use of the prototype data may have been gathered for the purpose of this evaluation. Several techniques for data collection may be used. One of these techniques is the direct observation of students using the environment. Furthermore, the student-computer interactions may be recorded automatically (software enhanced with facilities to make such records can be produced using the design systems THESIS and COOP). Additional information may be gathered by testing the students (preferable before and after the computer sessions) and by means of evaluation forms, filled in by the students. The implementation cycle and the life-cycle of the simulation environment end as soon as the results of this evaluation phase indicate, that the simulation environment is (no longer) the best tool available for reaching its goals. In other cases, the results of the evaluation may be used for improving the simulation environment.

References

Alessi, S.M. & Trollip, S.R. (1985) *Computer Based Instruction, Methods and Development.* Prentice-Hall. New York.

Bork, A. (1981) *Learning with computers.* Digital Press, Bredford

Chandra, P. (1988) *Adaptable interactive CBL design tools for education.* In: Lovis, F. & Tagg, E.D. (Eds.) Computers in Education.
543-547, Elsevier Science Publishers B.V. (North Holland)

Courseware Midden Nederland (1991) *Het Courseware Ontwikkelproces.* 10 pp. Courseware Midden Nederland, Utrecht [Dutch]

Davies, P. (1992) *Authorware Professional: multi-platform icon authoring.* The CTISS File 13 (April), 7-11.

Dekkers, J. & Donatti, S. (1981) *The integration of research studies on the use of simulation as an instructional strategy.* Journal of Educational Computing Research (74): 424-427.

Doerr, C. (1979) *Microcomputers and the 3 r's.* Haydn Book Company, New Jersey

Duchastel, P.C. (1987) *Structures and Methodologies for the Evaluation of Educational Software.* Studies in Educational Evaluation, 13(19): 111-117

Forrester, J.W. (1968) *Principles of systems.* MIT Press. Cambridge. England.

Hartsuijker, A. Hondebrink, J., Labordus, V. & Van Schaick Zillesen, P.G. (1989). *Een vijver in de computer; een integratievoorbeeld van Informatiekunde en Natuuwetenschappen.* National Institute for Curriculum Development (SLO), Enschede [Dutch]

Hebenstreit, J. (1988) *Computers and education: An encounter of the third kind.* In: Lovis, F. & Tagg, E.D. (Eds.) Computers in Education. 3-11, Elsevier Science Publishers B.V. (North Holland)

Koper, E.J.R. (1992) *Premises for the development of high quality educational software.* In: Plomp, T.J., Pieters, J.M. & Feteris, A.(Eds.) European Conference on Educational research, Book of Summaries, Vol 2, 634-637, University of Twente, Enschede

Kolb, D.A. & Goldman, M.B. (1973) *Toward a typology of Learning Style and Learning Environment.* Mass. Ins. Techn. Cambridge

Latzina, M. & Wedekind, J. (1986) *Simulationsprogramma: Systematische Beschreibung und Bewertung.* Log in (5/6), 35-41 [German]

Min, F.B.M., Renkema, M., Reimerink, B. & Van Schaick Zillesen, P.G. (1986) *MacTHESIS: A Design System for Educational Computer Simulation Programs.* In: J. Moonen and T. Plomp (eds.) Eurit'86: Developments in Educational Software and Courseware, 689-691, Pergamon Press, Oxford

Min, F.B.M. (1987) *Computersimulatie als leermiddel: Een inleiding in methoden en technieken.* Academic Service, Schoonhoven [Dutch]

Moonen, J. (1991) *Toegepast Onderwijskundigen: Architecten of Ingenieurs?* In: S. Dijkstra, H.P. Krammer & J.P. Pieters (Eds.), De onderwijskundig ontwerper., 47 - 60, Swets & Zeitlinger, Amsterdam [Dutch]

Moore, J.L. & Thomas, F.H. (1983) *Computersimulation of experiments: a valuable alternative to traditional lab work for secondary school teaching.* School Science review, 64(229), 641-655

Romiszowski, A.J. (1981) *Designing Instructional Systems. Decision making in courseware planning and curriculum design.* Kogan Page

Svirhezev, Yu. M., Krysanova, V.P. and Voinov, A.A. (1984) *Mathematical modelling of a fish pond ecosystem.* Ecol. Modelling, 21: 315 - 337

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.* PhD. thesis. University of Twente, Enschede, The Netherlands

Van Schaick Zillesen, P.G. (1991a) *Ontwerpen van Educatieve Computersimulaties: Een gezamenlijke taak voor onderwijskundigen, modelbouwers en docenten* In: Dijkstra, S., Krammer, H.P. & Pieters, J.M. (eds.) De onderwijskundig ontwerper; 187-198, Swets & Zeitlinger B.V., Amsterdam [Dutch]

Van Schaick Zillesen, P.G. (1991b) *Educatieve Computersimulaties en Natuurwetenschappen; Doelstellingen en Werkvormen.* SCOPE-Nieuws, nr 4/3, 4-7 [Dutch]

Van Schaick Zillesen, P.G. (1992) *SPIL-PROJECT: A Research and Development Approach to the Introduction of Educational Simulations in the Food and Bioprocess Engineering Curriculum.* Paper European Conference on Higher Education in Agriculture, Wageningen, The Netherlands, April 13-15, 1992

Van Schaick Zillesen, P.G. & Min, F.B.M. (1987) *MacTHESIS: a design system for educational computer simulation programs.* Wheels for the mind of europe, No 2, 23-33

Van Schaick Zillesen, P.G. & Min, F.B.M. (1990) *Towards an interactive design method for educational computer simulation programs.* Paper Eurit'90. Published by means of electronic media

Van Schaick Zillesen, P.G., Min, F.B.M., Gmelich Meijling, M.R. & Reimerink, B., (1991) *Computer Support of Operator Training: Constructing and testing a prototype of a CAL(Computer Aided Learning) supported simulation environment.* Paper International Conference for Corporate Training for Effective Performance (COTEP), 15 pp, University of Twente, Enschede, The Netherlands

Van Schaick Zillesen, P.G., Zwietering, M.H. & Van 't Riet, K. (1992) *Computer Support of Process Engineering Education.* In: Plomp, T.J., Pieters, J.M. & Feteris, A. (Eds.) European Conference on Educational Research; Book of Summaries (Vol 2); 493-497, University of Twente, Enschede, The Netherlands

Wedekind, J. (1985) *Einsatz von Mikrocomputern fuer Simulationszwecke im Unterricht.* In: Mandl/ Fischer (Hg), Lernen im Dialog mit dem Computer (210-217), Urban & Schwartzenberg [German]

Wedekind, J. (1986) *Software Tools for Teachers and Learners.* In: J. Moonen and T. Plomp (eds.) Eurit'86: Developments in Educational Software and Courseware, 243-247, Pergamon Press, Oxford