SIMULATE : simulation authoring tools environment

Hijne, Hans; Jong, de, A.J.M.

Published: 01/01/1989

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author’s version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 17. Dec. 2018
SIMULATE:
SIMULATION AUTHORING TOOLS ENVIRONMENT

Hans Hijne\textsuperscript{a} and Ton de Jong\textsuperscript{b}

Paper presented at the
Third European Conference for Research on Learning and Instruction EARLI, Madrid, September 1989

\textsuperscript{a} Courseware Europe bv, Knowledge Engineering, Ebbekout 1, 1507 EA Zaandam, The Netherlands. Email: hijne@coeur.UUCP

\textsuperscript{b} Eindhoven University of Technology, Department of Philosophy and Social Sciences, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Email: wmorton@eurtc3.UUCP and University of Amsterdam, Department of Social Science Informatics, Herengracht 196, 1016 BS Amsterdam, The Netherlands. Email: tondj@swi.psy.uva.nl.UUCP

Department of Philosophy and Social Sciences
Eindhoven University of Technology

OCTO-report 89/02
SIMULATE: Simulation Authoring Tools Environment

An authoring workbench for simulation integrated learning environments.

Hans Hijne and Ton de Jong

ABSTRACT

Computer-based simulations can provide a powerful form of instruction that is both dynamic and interactive. This paper focuses on optimalisation of learning by simulations by combining the simulation with an intelligent learning environment. An Intelligent Simulation Learning Environment (ISLE) provides the learner with an interactive environment in which s/he can learn about a simulated reality in an active and exploring way, and a tutoring environment providing support facilities (help and explanations) tailored to the instructional needs of the learner. The SIMULATE project investigates the functionality of an ISLE and the requirements for a workbench of the construction of ISLEs in a variety of domains.

1. INTRODUCTION

This paper presents the research currently being carried out within the project SIMULATE, as part of the overall project SAFE (Standard Authoring Facilities Environment). SAFE is a project within the DELTA programme of the EC.

DELTA

The DELTA programme is one of the research and development programmes initiated by the European Community. DELTA (Development of European Learning through Technological Advance) studies the application of advanced information, communication and broadcasting technologies in education and training. Industries, broadcasters, producers of educational material and academic institutions from different member states collaborate in R&D projects and have 50% of their work funded from the EC budget. In March 1989, the first phase of the DELTA programme, the Exploratory Action, has started, which is scheduled for two years with a total budget of 20 MECU. Thirty projects participate in the Exploratory Action which may have a follow-up in a DELTA Main Phase.

1The authors are grateful to Roger J. Hartley and Kenneth A. Tait (Leeds University) for their input into the SIMULATE research proposal on which this paper is based. Other persons who contributed to the research proposal arc: Peter Goodyear and John Self (University of Lancaster), Jan van Bruggen (Courseware Europe), Wim Vaags (Eindhoven University of Technology), Robert de Hoog and Bob Wielinga (University of Amsterdam), and Willem Bulthuis (Philips TDS).
SAFE
SAFE is a R&D project partially funded by the CEC under contract D1014 within the Exploratory Action of the DELTA programme. In total 15 partners are involved in the SAFE consortium of which Philips TDS is the prime contractor.

The major objectives of the SAFE project are:
- to define the architecture of a standardised workbench for the development of technology based learning material,
- to explore the potentials of specific types of open learning material and to define the tools that are required to develop open learning material,
- to explore and define tools that may support all phases of the development of learning material.

SIMULATE
Within SAFE the subproject SIMULATE concentrates on simulation based learning material. Based on an exploration of the requirements and functionalities of simulation based learning environments, the SIMULATE project aims at designing of integrated set of authoring tools for the construction of simulations embedded in an intelligent learning environment.

The SAFE partners directly involved in SIMULATE are: Philips TDS (Germany), University of Leeds, University of Lancaster (UK), TIFSA (Spain), University of Amsterdam, Eindhoven University of Technology, and Courseware Europe (The Netherlands).

The following section first provides an overview of the theoretical underpinnings of SIMULATE. Thereafter the objectives in the long term are described, and finally a summary of the workplan of the first 2 years will be given.

2. THEORETICAL BACKGROUND

2.1 Learning strategies, learning goals and instructional environments

Nowadays, prevalent information processing theories state that in the learning process the learner is actively involved in constructing and reconstructing his/her knowledge base (Anderson, 1983; Rumelhart & Norman, 1977; Rumelhart, 1980). This view is reflected in currently used teaching approaches. In contrary to older ways of teaching in which the learner was viewed as an 'empty box' into which knowledge could be poured, new approaches stress the active role of the learner and the importance of his/her foreknowledge. This change in teaching approach includes a change in learning goals from reproduction of knowledge to deep understanding of a domain and to transferrable knowledge. Michalski (1987) discusses several fundamental learning strategies, ranging from rote learning, to learning by observation and discovery. He concludes that an increased activity level of the learner goes together with an increasing complexity in the inference processes involved, and thus poses additional problems for the learner.

Some forms of Computer Assisted Instruction are well suited for this teaching approach, at which a high level of initiative and activity is expected from the learner. For example, at hypertext-like systems, learners are encouraged to explore a domain. A second example of CAI that supports active learner behaviour, is simulation based learning (Hall & Layman, 1987; Miller, 1984; Rivers & Vockell, 1987, Woodward,
Carnine & Gertsen, 1988). Here, learners also actively explore a domain, but now by even changing input values for the domain. The advantages of simulation based learning are recognized by authors as was shown in a recent Dutch survey (de Jong, Pilot & van Andel, 1988). This study showed that of the programs used in higher education in the Netherlands about 47% employed simulations.

2.2 The use of simulations in educational settings

2.2.1 Simulations: characteristics and functions

By computer-based simulation is meant that a phenomenon, a process, a system or an apparatus (or whatever it is that is being simulated) is formalized into a model. This model may have a qualitative character, or a quantitative one (or both). For a quantitative simulation, independent variables, dependent variables and parameters are combined into a numerical model, for qualitative simulations the model will not be purely numerical, but the components of the model and the relations between them will be represented symbolically or structurally. Both kinds of models can be implemented as computer programs. Simulations may differ on a number of dimensions. To mention just a few:

a. The number of parameters involved in the underlying model.
b. The presence of interactions between variables, and/or between parameters.
c. Static vs. dynamic simulations (time dependency).
d. In dynamic simulations: static vs. dynamic parameters.

Functions of simulations

A simulation can be presented to a learner with different objectives:

a. To aid the learner in learning about the underlying (dynamic or static) model.
b. To aid the learner in learning a process (procedure, skill) associated with an underlying model (e.g. learning a skill as in flight simulators).
c. To help the learner develop higher order (knowledge acquisition) skills (such as experimenting).

Of course there are other ways of using simulations in education e.g. to enable the learner to develop experience in using the model as a tool for solving other problems (e.g. construction problems), but these do not fall within the scope of the SIMULATE project.

Fundamentally, there are two ways in which a learner may work with a simulation:

a. The learner can vary the values of the independent variable(s) and observe the results of his/her actions as a value of the dependent variable(s), or, when he/she varies the values (qualitative or quantitative) of the simulation parameters, as a relation between dependent and independent variables. It is expected that by these actions stimulated

\footnote{For a full discussion of characteristics of simulations for learning, see van Joolingen, de Jong, & Camacho Molina (1989).}
and guided by certain cognitive process (in the educational literature generally labelled as 'discovery learning') the learner will gain an understanding of the model.

b. Another way in which the learner may manipulate a simulation is called 'modelling'. Here the learner may not only vary the values of variables and parameters, but may also interfere with the properties of the model (Ross, 1986; Goodyear, 1987). The learner may add, delete or modify the relations of variables and parameters in the model. In modelling the learner task may be to make the simulation work in such a way that for a specified set of input the model gives output that resemble those of the simulated reality.

This last method of model building is more closely related to the way simulations are utilized in scientific research; in the first way, however, the model remains as it is and the learner has to learn this model.

2.2.2 Rational for simulations as an instructional device

Having described what simulations are, and how learners may use a simulation, it is important to state clearly their advantages for learning. There are two categories of reasons for using simulations as an instructional device.

First, the basic advantage of simulations is that the learner is active in exploring (the content) of the model. The learner can set up his/her own hypotheses and test them by changing values in the simulation. This 'learning-by-doing' is generally accepted as an effective way to acquire knowledge, both procedural and conceptual. Hebenstreit (1987) adds that a computerized simulation creates a 'world' in between reality and some abstract model of it; this intermediate layer may help the learner to bridge the gap between reality and the model. Some characteristics of simulation are helpful in sustaining learning by doing:

a. In simulations the time-scale can be changed: real processes can be slowed down or speeded up to have an acceptable time of instruction;
b. In simulations a hypothetical reality can be created, e.g. a simplification of the reality.

Second, there may be economical and/or ethical objections or difficulties to learning in a real life situation; the following more 'practical' reasons may be given for using simulations as an instructional device:

c. Training on the job can be expensive or time-consuming;
d. Training on the job can be dangerous with respect to man, environment and/or material;
e. In simulations one can introduce rare or unusual events, like e.g. disasters so that the trainee can learn to react to them.
f. Sometimes, training on the job can be highly stress invoking.

Simulations thus seem to have many advantages for use as an instructional device. However, it is also clear that learning by means of simulations puts a high cognitive demand on the learner. As was shown in the present section simulations may range from very simple models to very complex ones. Learning these models by simply exploring alternatives may be very unfruitful. Learners may become involved in making changes randomly instead of purposefully manipulating variable and parameter values, and there
is a chance that especially weak learners may derive little benefit from the simulation (see for example Lavoie & Good, 1988, Bork & Robson, 1972) or may not take full advantage of the simulation environment (Njoo & de Jong, 1989).

2.3 Intelligent learning environments

It seems clear that in order to be effective as an instruction vehicle, simulations may need some facilities that support the student in learning from the simulation. This support and guidance may be provided by human tutors. An alternative way which corresponds more with the demands of Open Learning, is to incorporate the simulation into a computer learning environment. A simulation-based intelligent learning environment combines the features such as open-ended exploration in an interactive simulation with features of an intelligent tutoring environment like for example diagnostic feedback, explanation, coaching and sequencing problems based upon the student's evolving mental model. A key component in intelligent learning environments is a learner model. How the system should respond when errors are made, or questions asked, or coaching required, is dependent not only on the working contexts but also on the learner's requirements in relation to what they have knowledge of.

Although there are few knowledge based learning environments, there is clear evidence of the value, to motivation and learning, of providing intelligent support with educational and training simulations (Bork & Robson, 1972; Burton & Brown, 1979). Also the research performed on these systems, may provide some guidelines for the design of integrated simulation learning environments as envisaged in the SIMULATE project.

The program SOPHIE (Brown et al., 1975) simulated electronic troubleshooting and showed how learner's fault hypotheses could be tested by reprogramming the model to be in line with these hypotheses, and then matching the performance of the model against those of the actual faulted circuit. Feedback comments could then be generated. This technique was also used to prune and guide SOPHIE's own fault hypotheses when constructing hints for the learner.

STEAMER (Hollan et al., 1984) simulates the control of an engineering plant in large ships. The functional and structural properties of the model are used in providing causal explanations and qualitative reasoning for the learner, and STEAMER can also use different levels of abstraction in its coaching. The object orientated graphic design of the interface (Hollan et al, 1986) has high conceptual fidelity and can be directly manipulated by the learner when learning from the simulation, enabling easier transfer from the model to reality.

GUIDON (Clancey, 1987) - an expert system which can provide teaching on the diagnosis and treatment of bacterial infections - also gives particular attention to the tutoring strategies which guide question-answering and explanation. It proved necessary to make a distinction between rules which hold the domain knowledge and strategic meta-rules which control them; hence the system can explain concepts and rules as well as the ways they are used in diagnostic planning.

WEST (Burton & Brown, 1979), a learning environment which adopts a game-like format to teach mathematics, is able to make a distinction between valid responses of the student in the subject domain and strategically good moves. The former leads to coaching in the domain when errors are made, whereas the latter comments on the
strategic management of the simulation exercise. This distinction is particularly important in on-line help systems.

One of the most extensive on-line help systems is EUROHELP (Breuker, 1988; Hartley and Smith, 1988), which provides support for the learning of any information processing package. The system has several features which are relevant to SIMULATE. First, techniques of error diagnosis and the inferring of learner’s task plans from responses leads not only to the development of a learner model, but to the construction of a local knowledge need (of the learner) which is employed to guide the question-answering, explanation-giving and coaching. Second, an extensive set of didactic strategies has been developed which takes account of user knowledge and experience, the type of concept (support or operational) which is the focus of the coaching, and the type of relations which link it to other concepts and procedures. Third, Eurohelp employs an utterance generator to assemble and cohere the teaching content into natural language (Stausholm, 1986).

In summary, research will provide some guidelines for the design of simulation-based learning environments. As no current system linked to simulations takes the overall perspective, the SIMULATE project would be a significant and useful advance.

2.4. Authoring for intelligent learning environments

The process of authoring for course delivery on a computer based learning station is both technically complex and demanding. Authors must be able to take subject matter expertise and convert it into learning activities taking full advantage of the technology. This applies to a high degree for computer-based simulations having a teaching/learning function, the program development of which is relatively time-consuming. Mostly they are designed as specific exercises and programmed in whatever language is judged most practical. Even when they have been programmed using authoring systems, simulations are limited since they must work with pre-stored textual material and response matching schemes, and are usually deficient in the graphics they can employ. Further, related simulations will probably have to be conceived and implemented as separate, independent projects.

During recent years, several software tools for model building have become more generally available. One of the most interesting from the educational point of view is STELLA, which provides tools for drawing and building dynamic models following a generic data-flow system metaphor. The functional interconnections between the data stores, which serve as the dependent variables of the model, can be specified mathematically or graphically, and the time-steps used in calculating output also can be easily modified. Running on the Apple Macintosh, full use is made of the interface for both the designer and the learner of the simulation.

Some designers of enhanced simulations have followed an object orientated approach: i.e. data-structures in the simulation systems are defined as objects capable of performing certain behaviours, which are invoked by message-passing. SMALLTALK is the archetypal language for this technique, and recent work has also developed software tools (implemented in SMALLTALK) for creating interactive animation within simulations. STEAMER, referred to in the previous section, also follows an object orientated approach. An advantage is that different presentation forms, for the objects and their interactions, can be used without altering the rest of the system.
Although progress has been made in model building tools, software for developing simulations into intelligent learning environments is much less advanced. Such teaching/learning systems are usually constructed with three main components: the domain knowledge to be learned (expressed as a conceptual network or a model-based simulation), the model of the learner, and tutoring knowledge which uses the learner model to decide what should be learned and in what way (Sleeman and Brown, 1982; Self, 1988b). The approach advocated by designers of intelligent teaching systems has the knowledge components separately specified, and this is clearly advantageous since new teaching systems can be created by exchanging a knowledge component or adapting existing ones. This is rarely possible with conventional courseware, where knowledge used in decision making is bound into its specific content (Cazala et al., 1987, Chapter V).

In summary, there is an increasing interest in the development of software tools which are able to support author-designers of intelligent learner environments, but such work is still in its initial and formative stages. SIMULATE would make a notable contribution through its concentration on simulation-based learning environments and the generic specification of a shell/software tools for the construction of the model, its interface and the tutoring components.

3. PROJECT OBJECTIVES: AN AUTHORING WORKBENCH FOR INSTRUCTIONAL EMBEDDED SIMULATIONS.3

We envisage a system, called SIMULATE (SIMulation Authoring Tools Environment), which is an integrated collection of authoring tools for constructing simulations embedded in an intelligent learning environment.

The result of applying SIMULATE will be an Intelligent Simulation Learning Environment (ISLE) containing a simulation together with advanced help, hints, explanations and tutoring facilities. This environment may offer the learner a 'guided tour' through the simulation tailored to his/her specific instructional needs.

Figure 1 introduces an overview of SIMULATE and ISLE and their interrelation. The architecture depicted here should, of course, be regarded as a tentative one.

The Intelligent Simulation Learning Environment contains a domain representation (DR) that the author has created of the subject domain. This domain representation includes all the information necessary for simulations in the domain. Besides this domain representation, the Intelligent Simulation Learning Environment will encompass a number of modules all of which make use of the domain representation, and adapt the simulation to the needs of a particular learner. The modules will contain the following types of knowledge:

a. Knowledge about how to present a simulation depending on the characteristics of the underlying model (SMM: Simulation Manipulation Manager);

---

3In this paper a global and preliminary outline will be given of this system and the learning environment that can be created with it. During the DELTA exploratory action the specifications for such a system will be developed. A next phase in the DELTA programme might furnish the opportunity to develop the system.
Figure 1. SIMULATE and the resulting Intelligent Simulation Learning Environment.

b. Teaching knowledge in order to guide the learner and to explain the result of a learner action and to provide the correct feedback (TM: Teaching Manager);
c. Knowledge of prospective learners in order to be able to build a learner model (UMM: User Model Manager);
d. Diagnosing knowledge in order to be able to diagnose the faults and misconceptions of the learner (DM: Diagnosing Manager);
e. External representation, this means knowledge on what types of external representation of simulations can be given to learners (ERM: External Representation Manager);
f. Help facilities which reference:
   - Domain knowledge, e.g. simulation specific variables and parameters;
   - General simulation related concepts, such as parameters and variables;
   - Manipulation information, i.e. information that explains to the learner how to manipulate a particular application, for example how to input and change his/her data etc. The Help Manager (HM) contains this type of knowledge.

SIMULATE is the authoring workbench that will contain all the means (tools and methodologies) necessary to create an ISLE that can then be presented to a learner. Therefore, SIMULATE will consist of three main components:

I  Simulation Construction Tools (SCT).

II  Learning Environment Tools (LET).

III  Simulation Construction Methodology (SCM).
The Simulation Construction Tools (SCT) will enable the instructional designer to develop a simulation of any subject matter domain. It will allow the author to construct a domain representation and a control structure in order to produce a simulation.

The Learning Environment Tools (LET) will allow authors to create the environment around the simulation. In essence these tools concern learner modelling, the external representation of the simulation, and teaching strategies.

The Simulation Construction Methodology (SCM) describes the usage of the tools and will guide the authoring process.

4. METHODS AND TECHNIQUES (TECHNICAL APPROACH)

The workplan of the SIMULATE project in the DELTA Exploratory Action globally has three phases: (a) an inventory phase, (b) an experimenting and prototyping phase, and (c) an integration phase.

4.1 The inventory phase

The inventory activities are centred around four *design components* which represent the elements of the instructional situation where a learner learns by interacting with a simulation and which are directly related to the main components of the SIMULATE authoring workbench as depicted in Figure 1. These design components are:

a. the model
The basis of a simulation is some type of model of the reality or of an artifact. These models may differ on a variety of dimensions, e.g. the number of parameters, number and type of relations, static vs. dynamic, qualitative vs. quantitative relations (Gredler, 1986; Welham, 1986). A representation of the domain is strongly related to the model and its characteristics; this domain representation can be used not only to form the simulation but also to function as basis for the tutor component of the SIMULATE system.
Information gathered in relation to this topic is reflected in the Simulation Manipulation Manager and the domain representation from the ISLE in Figure 1.

b. the learner
Learners will display a number of cognitive and non cognitive characteristics that have significance for interacting with (tutorial embedded) simulations. These characteristics range from domain knowledge described in generic terms (de Jong & Ferguson-Hessler, 1986; Ferguson-Hessler & de Jong 1987), through learning processes used (Ferguson-Hessler & de Jong, 1988), to, for example, self-control. An Intelligent Simulation Learning Environment will need to adapt the instruction to individual differences in these characteristics of the learner, to his/her misconceptions, and to the knowledge base of the learner as it develops during the learning process. Knowledge of these learner aspects provides the basis for learner modelling (Self, 1987; 1988a) in an intelligent learning environment.
The User Model Manager and the Diagnosing Manager from the ISLE in Figure 1, will be based on this design component.
c. the tutor
An intelligent tutoring system needs knowledge on tutoring and help facilities. In the field of simulations several general tutoring approaches can be taken, e.g. coaching, learning by diagnosis (Woolf et al., 1987), or going from qualitative to quantitative models. Also knowledge on how and when to present help (to be distinguished from tutoring, see Carroll & McKendree 1987), and what type of help is required to suit particular local needs (EUROHELP) has to be present.
Information related to this design component, will be used in the Teaching Manager and Help Manager from the ISLE as depicted in Figure 1.

d. the learner interface
Learner interfaces have a double function. They have to be designed in such a way that the output of the system is understandable to the learner, and that the learner may easily transform the actions s/he has in mind into inputs for the system (Norman and Draper, 1986). Screen output for simulations may range from tables to full graphical output. Some of the man-machine interface aspects have a more general character that has to be specified to working with simulations (e.g. direct manipulations), while others are more specific for simulations (e.g. the fidelity of the interface, meaning the correspondence of the interface to reality). In this proposal we will mainly focus on those interface aspects that are specific for learning with simulations (see also Symons, 1985). The External Representation Manager from Figure 1, is related to this design component.

Moreover, the inventory activities are subdivided into two streams. The first stream provides, along the lines of subdivision into the above mentioned design components, a description of the knowledge and educational information relevant to the use of simulations in education and training. Accordingly, for each of the elements of the instructional situation with simulations, the second inventory stream describes the currently available technologies and construction tools for the development of simulations and learning environments.

Within both streams, the inventory will provide the basis for an assessment of the applicability of currently available information and technologies, to the use of simulations in education and training and to the development of a methodological and software system like SIMULATE, and for a formal description of each of the design components. This formalization plays a central role in the SIMULATE project. It serves the following goals:

- To have an implementation independent description of the components;
- To ensure that several components in the final SIMULATE system will be able to communicate with each other;
- To give the system the opportunity to communicate and reason about its content;
- To enable the rapid inclusion of new developments.

4.2 The experimenting and prototyping phase
The second phase of the project will involve experimental research on learning with simulations, since it is expected that the inventory activities will not provide all the information needed on this topic. The main reason for this is the rather young history of the use of simulations in learning, and hence the experimental research will service three
goals:

- To delineate the type of knowledge that is lacking;
- To try-out experimental pilot techniques that will be useful when more extensive research has to be undertaken in the future;
- To try-out and assess observational techniques that can be used when evaluating mock-ups or prototypes of SIMULATE and ISLE.

In this phase some mock-ups or prototypes of ISLE's will be built, in different domains and with different levels of learners.

4.3 The integration phase

The activities in the first two phases all have a descriptive character. The integration phase is meant to render information of a prescriptive nature in three ways. First, specific rules for the use of simulations in education will be derived. The functionality of integrated simulation environments will be identified, and hence the requirements for an ISLE are stated. For example: the prescriptive rules will give information on how to reach educational goals by presenting a type of model(s), with a certain interface, under a specific tutorial strategy for learners with identified characteristics. Based on the identified requirements for an ISLE, the second integration activity involves stating the requirements and functional specifications of the SIMULATE system. Related to this, the integration finally provides an outline of a methodology for the development of simulations embedded in intelligent learning environments.

Figure 2 depicts the organization of the SIMULATE project in a diagram. The nucleus of the project is formed by the four design components, with an integration surrounding them. This forms the knowledge that will be present in an ISLE. To create the ISLE several tools will be available in SIMULATE, with the integration of the tools represented by the outer circle.

5. CONCLUSIONS

The project is now on its way for a number of months, and we already can come to a few conclusions.

First, we found that the diversity of the use of the concept simulation to be very large. Simulations, even within a context of learning may differ widely. A delineation of our use of the term simulation will be given in de Jong (1989).

Second, the lack of research information on learning with computer simulations is even larger than we expected it to be. Especially detailed information on how people learn in exploratory learning environments and the problems they encounter is still an undiscovered territory.

Third, the dependencies between activities and workpackages as sketched above is bigger than we might have expected. Specific topics may fall within different design components. The decision where to incorporate them will be a delicate one.

The fourth, and last, conclusion might be that the field of learning in exploratory learning environments, is a promising one with many difficult, but highly interesting aspects. Some of these aspects will be discussed in the two first deliverables of the
project to appear at the end of 1989 (de Jong (Ed.) and Tait (Ed.)).

REFERENCES


