Developing pedagogical simulations: Generic and specific authoring approaches

Viviane GUERAUD, Jean-Philippe PERNIN
Laboratoire CLIPS-IMAG- - Equipe Arcade -
BP 53 - 38041 GRENOBLE Cedex 9, France
Jean-Philippe.Pernin@imag.fr

Abstract:
Our team is interested by the production of pedagogical software based on simulation. These last years, we have worked in two quite different contexts: technical training in the industrial world, and general training in the academic world. The output of this research is a set of theoretical results, the development of authoring environments, and a group of experiments.

In this paper, we present two different approaches that we proposed and experimented in two different contexts. The first approach is generic and covers a wide variety of situations by providing authors with a set of general-purpose formalisms. The second approach is specific; it provides authors with a set of tools specially tailored to cover a well-defined class of problems. We describe the results of these experiments and attempt to evaluate the two approaches according to their different contexts of use.

Keywords:
Interactive Computer Based Learning Environments, Pedagogical Simulation, Authoring Approach, Experimentation.

1. Introduction: Develop operational solutions
These last years, many researchers on Interactive Computer Based Learning Environments have attempted to provide sophisticated solutions based on a cognitive approach and to study interactions between system and learners [BAL 97]. This approach requires reasoning models of the domain expertise, of the pedagogical expertise and of the learner behaviour [NIC 88]; it often meets problems due to the complexity of these models and frequently necessitates a close and lengthy collaboration between specialists (pedagogical experts and computer scientists). Actual efforts by normalisation groups (IEEE LTSC [IEE 98]) aim to harmonise the different approaches and to propose computer environments that would simplify authors' tasks. Even if they seem promising, these solutions do not make it possible yet to solve pedagogical needs and to satisfy the economical and practical constraints faced by business and industrial organisations, and by the academic world.

Our team, ARCADE (Ateliers de Réalisation et de Conception d’Applications Destinées à l’Education), has been working since 1985 in the domain of strongly interactive instructional applications based on manipulation, discovery, and simulation. We developed a computer science training application for university students [CAG 90]. This reinforced our conviction that this kind of applications should not be seen as a self-sufficient tool, but rather as an original and efficient add-on to the training process. We next worked on the definition of methods and tools for the production of similar applications, but dedicated to other scientific and technical fields. From the beginning, we intended to provide a set of operational solutions, applicable not only in academic, but also in industrial, contexts.

At first, we collaborated with CORYS, a software company specialised in the development of pedagogical simulations in the energy and transportation fields. The main objective was to study how to rationalise
production and decrease the development costs. Our analysis of the development process indicated that many actors (domain specialist, modeller, pedagogical expert, interface specialist, project leader, etc.) were involved and required different formalisms, methods and tools. The integration of these efforts was difficult. Furthermore, there was a clear need to produce simulations that would be modular and easy to modify.

Our main contribution consisted in the definition of a conceptual framework: MARS (Model-Associations-Representation-Scenario) [PER 96a,b] which provides a theoretical basis to structure the internal architecture of a pedagogical simulation. MARS distinguishes four independent components:

- the Model where the system behaviour is described;
- the Representation which deals only with the simulation interface and with pedagogical interactions;
- the Associations which links the two previous components;
- the Scenario which regroups abstract pedagogical aspects (list of exercises, pedagogical controls of the learner’s activities).

Building on this conceptual framework, we developed different integrated environments which offer to authors different levels of visibility (workspaces) to the basic components according to production context, authors' capabilities, technical domain studied, etc. In order to conciliate these different constraints, compromises had to be found between (a) structuring authors' approach by clearly separating the different features and (b) simplifying their activity.

In this paper we present two different approaches we have used and the experiments we made in different, one industrial and one academic, contexts.

The generic approach consists in covering a wide variety of situations by providing explicitly authors with a set of generic formalisms (for example, mathematical formalisms, event-condition-action rules, Petri networks, object-oriented approach); this kind of approach is also used in other authoring environments [DEJ 94], [MAR 97], [MUN 95], [ROS 94], [TOW 95], [VAN 96].

The specific approach consists in providing authors with a set of tools specially developed to cover a well-defined class of problems, by assisting them very closely and by generating automatically a large part of the application.

Finally, we attempt to describe what we learned from this work and to stress the main lessons we may learn from these experiments.

2. An industrial context: Hewlett-Packard

2.1 TPEC Needs

The MELISA project (Methodology and Environment for developing Learning, Instruction and Simulation Applications) took place in 1994-1997. Its goal was to provide Hewlett-Packard T.P.E.C. technical trainers with a set of methods and tools they could use to develop, by themselves, simulations of equipment manufactured by the company. The need for simulations was clearly motivated by the deficiencies of their current practices.

The first deficiencies were pedagogical. Know-how training, when based on direct manipulation of equipment, frequently faces very difficult problems: unavailability of equipment, difficulty to reproduce real phenomena, necessity to degrade the equipment to reproduce failure situations, etc. There is often also a significant distortion between what the technician is supposed to have learned during training and his actual behaviour in the field.

The other deficiencies were organisational. Many TPEC training sessions are intended for a public of engineers and technicians who may have to handle a large variety of equipment. These sessions must satisfy
many organisational constraints: trainer availability; sufficient number of trainees; interval between sessions, … The training of a field technician may thus happen to be outdated or inadequate.

TPEC decided to use simulation to complement the current methods, with the following expected benefits:

- Complement the training sessions with manipulations of simulated equipment;
- Enrich the self-training resources provided to technicians wishing to update their skills when new models are introduced;
- Certify outside technicians as capable to maintain Hewlett-Packard equipment.

The decision to use simulation was easy to make, but its application appeared rather complex. Technical and economic aspects of the development of pedagogical materials are under the responsibility of an engineer. He understands the technical and pedagogical aspects of the product. However, neither he, nor his team, is trained to develop software.

To solve this problem, TPEC attempted first to sub-contract the simulation production to specialised software companies. This approach was discontinued since it took too much time and was much too expensive. Furthermore, the delivered applications were not easy to reuse or modify.

TPEC next evaluated if the trainers themselves could develop their own simulations, with authoring systems currently available on the market. No system was found adequate; they were all too complex to use; and they did not offer sufficient functionalities for pedagogical control.

TPEC finally decided to develop solutions adapted to their specific needs. The provided environments had to be usable by trainers, who were regular computer users, but who had no software development experience. The resulting simulations had to be easily adaptable to well-defined pedagogical contexts (initial training, self-training and certification). Simulation development time and cost had to be similar to what was required by traditional training. A newly developed simulation had to be available in less than three months to follow the release schedules of new products.

2.2 Our propositions

Current TPEC training is done in three-day sessions that include lectures, paper documents, videos, and hands-on manipulation of real machines. TPEC wanted to replace equipment manipulation by simulations. These simulations were to illustrate (1) how to diagnose a malfunction and (2) how to repair a machine by tuning or replacing faulty components.

After a first analysis phase, we were able to remark that repair rules for different machines followed similar strategies, while diagnostic methods were specific to each kind of machine. We thus choose to propose two very distinct environments to develop these two aspects of simulation.

A specific "GeneSimu" tool was used to develop repair simulations. With this environment, the author simply fills a form to describe the machine characteristics: active components; assembling conditions; functional of mechanical dependencies between components; main known failures and their associated repair procedures. All components of the repair simulation (Model, Representation, Associations and Scenario) are then automatically generated from a set of predefined rules about the interface and the pedagogical control. The author can then very easily improve the simulation interface by replacing for instance basic predefined interface elements by photographs.
Fig 1: Simulation interface, before and after personalization
(plain rectangles have been replaced by photographs)

With the resulting simulation, the trainee can take apart, test, tune or replace components.

A generic "Melisa Core" environment was used to develop diagnostic simulations. With this environment, based on MARS conceptual framework, the author must:

- model the foreseeable behaviour of the machine for each fault situation. The formalisms are based on an object-oriented approach (definition of properties and methods; use of Harel statecharts for representing the dynamics of the simulation),
- specify the learner interface, by using object libraries,
- and finally define the relationships between the Model and the interface, each association being defined with event-condition-action rules.

Furthermore, we provided the author with a precise procedure allowing a structured use of the environment. With the resulting simulation, the trainee can diagnose the current condition of the simulated equipment.

Specific software mechanisms were provided to insure the coherence between the two parts of a simulation.

2.3 TPEC experimentation

Our experimentation took place in MediSchool, the division in charge of medical equipment training. During the year 1997, four simulations of medical equipment (defibrillator, electrocardiograph…) were developed by MediSchool trainers and tested in real training sessions.

The results were globally encouraging. A more detailed analysis is given in [PER 98]. According to MediSchool, it would have been impossible to produce these simulations without the provided tools. Today, only one or two months are required to obtain a complete (diagnostic and repair) simulation, 2/3 of this time being spent on a precise analysis of the problem, and 1/3 on the simulation effective development.

A first conclusion is that the development of repair simulations with the specific Genesimu authoring tool was considered as easier than the development of diagnostic simulations with the generic tool. The use of Melisa Core tool requires a more structured approach; more time is necessary to specify each component; authors have difficulty to translate their initial problem into the proposed formalisms and to describe completely the required simulation.

The simulations were used in 5 different real training sessions with about 50 trainees (HP technicians, as well as customer engineers). Trainees' reactions were very favourable, their most frequent critic concerning the lack of additional pedagogical documents (such as hypertexts). It is planned to extend the use of these simulations in the context of synchronous or asynchronous distant learning.
3. An academic context: the ARIADNE Project

3.1 Needs

The ARIADNE project (Alliance of Remote Instructional Authoring and Distribution Networks for Europe, 1996-1998) is part of the Telematics Applications European program [FOR 97]. Its objective is to provide teachers and trainers with:

- a set of coherent tools to define effective training curricula based on a variety of pedagogical resources (hypertexts, multiple-choice question evaluation, simulations, etc.). These resources, available in a centralised knowledge database, can be combined to define coherent pedagogical units usable in traditional training, as well as in self or distant training situations.
- a set of tools to develop new resources that could be later incorporated into the database. These tools should reduce production times. In consequence, reuse and adaptation of resources should be made as easy as possible.

3.2 Our propositions

Our contribution to the ARIADNE project is the OASIS tool (Operative Authoring tool for Scenarios and Interactive Simulations) [COR 97]. Our goal was to give teachers the possibility to easily adapt previously developed simulations, and also to create new simulations. Our task was more complex than in the TPEC situation because no simplifying hypotheses could be made on the kind of simulations, on the domains involved, on the pedagogical controls of the learner's activities, and even on authors' capability.

We therefore had to provide a "generalist" environment suitable for a variety of simulations: it will structure author's approaches and offer convenient visual tools. Based on the same MARS model than "Melisa Core", OASIS offers four specialised workspaces, devoted to the following tasks: (1) creation of an abstract Model of the simulation, (2) creation of the learner's interface, (3) creation of a "free simulation" by establishing links between the Model and the interface, and (4) creation of a "controlled simulation" by adding a set of exercises.

The formalisms used within the three first workspaces (Model, Associations, and Representation) are the same than in Melisa Core (Cf. §2.2).

The fourth workspace (Scenario) proposes an original way to specify exercises. The exercises supported by our tool are based on a relatively simple problem-solving approach: the learner has to reach a given goal, possibly by reaching first a set of intermediate sub-goals; controls can be specified to check for incorrect actions. When a learner works on an exercise, OASIS monitors his activities, detects sub-goals reached and incorrect actions. Defining a scenario requires two steps. In the first step, the author demonstrates how to solve the exercise. For this, he manipulates the free simulation, exactly as a learner would. He places the simulation in desired situations (i.e. the initial state, intermediate sub-goal(s), the final goal, etc.). When the author reaches a wanted situation, he records automatically all the relevant data, simply by clicking on a "photography" button. In a second step, the author may use a specialised editor to modify the automatically generated scenario. For instance, he may replace a photographed value by a range of acceptable values, in order to make the exercise less constraining. The author may also customise the pedagogical feedback given to the learner.

Depending on his capabilities, a teacher may use OASIS at different levels [COR 98]:

- The first level requires minimal technical abilities. It concerns the definition of exercises to complement an already existing simulation;
- The second level requires a more structured approach. It concerns the creation of new free simulations from existing simulation models. The author has to create a new learner's interface adapted to his specific
public, and to link this interface with the model;

- The last level concerns the design of abstract simulation models. In most cases, this activity requires a strong methodological competence, or even real modelling experiences.

### 3.3 CAFIM experiments

The CAFIM (centre for self-training and multimedia innovation of the Grenoble Joseph Fourier University) provides students with a set of self-training and self-evaluation activities for scientific disciplines. We experimented OASIS in this context with seven teachers of various fields (Physics, Chemistry, Geology, and Automation). They volunteered to participate in the development of simulations in their own speciality. Four of them had no programming competence, two were used to develop CAT applications, and the last one had some methodological competence (object modelling). After a 12-hour preliminary training on OASIS, they each specified a simulation. In order to validate the different access levels, we asked each teacher to develop all or only a part of his application, depending upon his capacities, the remaining part being done by our team.

Our main conclusions are the following:

The strongly structuring approach imposed by OASIS (separate development of Model, interface and exercises) was diversely appreciated. While well accepted by the teachers who had no previous experience with application development, it appeared to others as a constraint and a source of difficulty. All, however, agreed that modifications were made much easier by this separation of the different aspects.

Designing abstract models appeared as the most complex activity. It requires a strong capacity of abstraction that can only be obtained through a specific training. In most cases, we had to define ourselves the abstract models in collaboration with the teachers. This difficulty has also been observed in other contexts [VAN 96].

We were more surprised to discover that it was difficult for teachers to imagine new exercises to complement an existing simulation. How to exploit the simulation with learners seems to be a difficult pedagogical question: what goals should they give to a learner? How to define a progression of exercises? Etc. In this occasion, we observed teachers reconsidering completely their own pedagogical methods. All agreed that, technically, developing exercises with OASIS was a very simple task.

Finally, we were able to validate the various levels of use. The first level (exercise creation) does not require any specific technical competence; the teacher can effectively concentrate his efforts on the pedagogical aspects. The second level (interface creation) is more difficult: this activity, often well liked by teachers, may require a long time and many iterations to improve presentation details. Finally, only the teacher with a previous modelling experience was able to operate correctly at the last level (model design).

### 4. Discussion

#### 4.1 Generic versus specific authoring tools

The two contexts described above are quite different. ARIADNE context is very general: we do not know what kind of simulations will be developed and we cannot make any hypothesis on authors’ capabilities. In Hewlett-Packard case, applications were of a precise type, authors had well-defined (homogenous) competencies, the pedagogical contexts of simulation usage were known.

We now have two available environments:

- A generic environment to develop various scientific or technical simulations. An author may use it, according to his needs and competencies, to add new exercises to an existing simulation, to create a free simulation based on existing models, or to define new models (this last activity being the most difficult).
• A specific environment dedicated to a particular kind of simulations. The author's task is much simpler and is reduced to a simple form filling activity.

We think that these two categories of environments are useful, but that if many simulations of the same type are required, authors greatly benefit from a specialised environment. Following our experience with the development of these environments all based on the MARS model, we now want to study how to produce, at low cost, specialised environments. We presently work on the definition of a framework [FAY 97] that will reuse our know-how at 3 levels: software architecture, design and implementation.

4.2 Pedagogical integration of simulations

In the two contexts, simulations are integrated differently to the pedagogical learning process. At Hewlett-Packard, simulations were defined as a response to precise problems. Thus the pedagogical objectives were clearly given and the pedagogical integration defined in advance. In CAFIM experiment, the volunteer teachers wanted to have simulations on particular topics, but they did not clearly imagine in advance, how they would use them. This explains the difficulties they had to define, afterwards, useful exercises. Their approach was more exploratory than guided by precise needs, and thus not very efficient.

ARIADNE tools, and especially the curriculum editor integrating tool, are now operational. Trainers may use this complete environment to insert simulations in a curriculum with clearly defined pedagogical objectives. Further experiments (ARIADNE II European project) should thus show that trainers pay more attention to the proper integration of the desired resources.

4.3 Exercises and pedagogical control

With our generic environment, authors can start with a free simulation and add exercises. Experimenters reported very favourably on the pedagogical interest of this kind of exercises and found them technically easy to define. This encourages the reuse of simulations already developed (with our tool) by facilitating their pedagogical adaptation.

To further augment reuse possibilities, we are now studying how to add such exercises and pedagogical controls to simulations not developed with our environment. We work on the definition of a minimal set of services that the simulation applications would have to provide. We also want to build reusable components, based on these services, that could be added to the applications in order to implement the desired level of pedagogical control.

Finally, we are interested by new pedagogical contexts, such as distant learning. We study tutoring and collaborative work, in both synchronous and asynchronous modes. The pedagogical controls we propose could facilitate tutoring, as well as collaborative work, by giving access to pertinent information on learners' activities.

Bibliography


[COR 97] Cortes, G., Guéraud, V. Helping The Teacher to Create Simulation Based Exercises, 4-th International Conference on Computer Aided Engineering Education (CAEE 97), Krakow, Poland, Sep. 1997


[PER 96a] Pernin, J-P., MARS : un modèle opérationnel de conception de simulations pédagogiques, Ph D., Université Joseph Fourier-Grenoble1, France, Jan. 96.


