Lessons learned from

a University-Industry Cooperative Project in Tutoring Systems

CLAUDE FRASSON ESMA AIMEUR

Université de Montréal Département d'informatique et de recherche opérationnelle 2920 Chemin de la Tour Montréal, H3C 3J7, Québec, Canada E-mail : {frasson, aimeur} @iro.umontreal.ca Tel : 1-514-343 7019 Fax : 1-514-343 5834

Abstract: this paper presents the lessons learned from a project developed in cooperation between universities and industries. In the present final phase, we can show positive results but also highlight some problems which have to be avoided to maintain a coherent and progressive development approach. The project aims at developing various components of Intelligent Tutoring Systems (ITS) in a discipline involving multiple knowledge-based systems. They include several components such as pedagogical expertise, learner model, subject matter, but also various methodological aspects. After reviewing the main advantages of ITS ry describe the most important components of the architecture and the progression of the project. We illustrate, through the development of the different prototypes, the difficulties to overcome and the solutions retained.

1. Introduction

Cooperation between University and Industry is a necessity as technology transfer represents a big challenge for the next few years. Communication between research environments and real applications production centers must be improved in order that they both benefit from complementary knowledge or mutual know-how.

This article presents an example of such a cooperation in a first priority domain : training in industry. Training industrial employees has become of crucial importance, particularly in period of economic competitiveness. The increasingly rapid change of advanced technologies, the transformation of responsibilities of employees that have often to acquire, in their environment, new and multiple skills, perhaps even new roles, make training an essential key of the economic evolution. New knowledge on more complex environments have to be acquired in a shorter time by employees less specialized. However, current *industrial training* suffers from several gaps

- Training session are very rare : recent studies (Industry Science and Technology Canada, 1991) have shown that the budget and the proportion of times granted to training by enterprises were clearly insufficient, compared to several countries and notably to Japan (6,7 hours of annual training in Canadian enterprises Vs 200 hours to Japan).
- The quality of training is questionable : just attending a prescriptive course is considered as a guarantee of good result. No real evaluation of the course and its contribution to the enterprise is really undertaken.
- The update of existing courses is sometimes long and expensive, and current systems do not allow to maintain employees at the fine point of knowledge, but this becomes increasingly important in an economic competitiveness context (Stahmer, 1991, McIlraith, 1991).
- It leads to stop production tasks, pulling the employee out of the workplace; in addition it is rarely well adapted to the environment in which the employee will be situated leading to out of context training.
- Employees are sometimes dispersed between branches and the capacity of training departments is often insufficient, entailing differences in the implementation of the new competencies.
- The characteristics of the learner, learning style, conceptions and misconceptions, are not taken into consideration.
- It is very difficult to detect misconceptions acquired by the employee during the training session and it is especially difficult to eliminate them. These misconceptions are susceptible to bring important prejudice to the enterprise. For example, the maintenance of US Air Forces planes provided by insufficiently trained technicians has entailed costs superior to the breakdowns themselves.

Several recent approaches highlighted the convergence of different disciplines to improve ways and methods of training, evolving from experiences in Computer-Assisted Instruction (CAI) to Intelligent Tutoring Systems (ITS). Thus, works in artificial intelligence, education, cognitive sciences, and multimedia, aimed to tackle the problems mentioned above and contributed to the development of ITS.

ITS aim to provide a teaching that could adapt dynamically to the learner, his/her learning style, preferences, rhythm, and especially the knowledge level. They provide individualized training, realized in the environment of work of the employee, on distant

sites, according to various forms of learning able to strengthen the development of new skills currently sought-after in enterprises. Measures made at Carnegie-Mellon university (Anderson, 1988) have shown that the intelligent tutor usage allowed an improvement of quality of learning (43%) and a reduction of length of learning (30%). Others more spectacular experiences have been undertaken on tutoring systems for the detection of plane breakdowns (SHERLOCK system, developed by Lesgold, Lajoie et al, 1990). Results obtained have shown that an apprenticeship of about *twenty hours* allowed employees to acquire a comparable expertise to that of employees having 4 years of experience.

However numerous difficulties remain with ITS construction. ITS generally use several knowledge-based systems that are complex to structure and load: knowledge on the subject matter, knowledge on the pedagogical approach and knowledge on the learner's knowledge. In this paper we will describe a project that aimed at developing a complete ITS environment in cooperation with industry. We will comment on the approach used to develop and integrate all these knowledge systems, the problems we were faced, the solutions we have retained and the recommendations we can extract from this experiment. We first situate the development constraints in which we were placed, then we develop all the problems that we have addressed : the organization of the curriculum which represents the subject matter knowledge, the pedagogical approach adopted in term of multiple strategies to be applied according to the learner, the characteristics of the learner model with the different aspects of knowledge to consider. We also give a short description of the architecture and we comment on the problems during the design process. Finally, we present the results obtained in terms of prototypes and development environment produced during the project. We conclude with the lessons learned from this experiment.

2. Context of the project

SAFARI is a project under the auspices of Synergie, a programme sponsored by the Ministry of Industry, Trade, Science and Technology of the Government of Québec. The main objectives of Synergie are (1) to enhance cooperative research and development between universities and industry, (2) to accelerate the product development cycle, (3) to facilitate the transfer of knowledge between research establishments and the industry, and (4) to educate highly qualified professionals in the domain. The main objective of SAFARI is to develop a *methodology* and an *environment* for the *creation of tutoring systems to be used in professional formation*. The focus is on teaching mostly procedural knowledge concerning the operation of devices such as medical instrumentation, consumer appliances and aeronautical instruments.

SAFARI involves four Québec universities, two private enterprises and a government agency. The industrial partners are **Virtual Prototypes** Inc., providing a simulation software package VAPS, and **Novasys** Inc. VAPS (Virtual Applications Prototyping

System) is a high-quality commercial interface-building and simulation system, used in many areas (such as airline cockpit design. The SAFARI team includes eight professors as researchers on a part-time basis : C. Frasson (project leader), E. Aimeur, G. Gauthier, I. Gecsei, G. Imbeau, M. Kaltenbach, S. Lajoie, B. Lefebvre ; about 20 M.Sc and Ph.D students, two full-time programmers and two engineers among the industrial participants.

The main constraints on the project concern the two ways of work of the different partners that must be combined and respected.

- First, *academic research* must be realized by producing advanced results, scientific articles and training of highly qualified personnel (Ph.D. and M Sc students) who can be transferred to industry. Often, academic research pursue long term objectives but this project, however, has to be linked with realistic objectives that must be reached and validated within reasonable periods (6 months).
- *Industrial research* obeys to the principle of deliverable which must be respected by all the students and researchers, with fixed deadlines for reports or prototypes delivery. This constraint is not easy to adapt with students having to do their courses and, sometimes, with no real experience in software engineering. In addition, students research subjects must be adapted both in term of objectives and deliverables within their studies. Once the different steps of the project are fixed, research objectives and schedules have to be fixed according to the available resources.

Another difficulty is to maintain a cohesion of the team which is, in that case, a multidisciplinary team with different figures of work. Expertise is complementary (Kaltenbach & Frasson, 1994) and needs to be adapted to the demand of the project, respecting the delay. This means that a researcher cannot continue on a research topic if some results (indicating that the track is promising) have not been reached within a period of 6 months and so, has to accept to change his/her research target.

3. Problems addressed

3.1 Curriculum organization

The dynamic adaptation to the learner requires a deep knowledge of his behavior in learning situation and more particularly during the resolution of problems. For this purpose it is useful to know the different states through which the reasoning of the user passes (Frasson et de la Passardiere, 1990). Problems are how to detect the reasoning of the learner and obtain a model ? How to facilitate the production of correct reasoning, help the learner in problems solving situations and improve learning ?

Building a curriculum which can be used efficiently by an ITS is also complex as the objects in question are mainly knowledge elements (or networks of knowledge) which cannot be easily decomposed in terms of functionalities due to concepts related to

pedagogy and didactic consideration. Modelling this kind of knowledge remains difficult to realize and concerns the following problems : knowledge acquisition, distribution according to didactic objects, concepts, examples, exercises, scenarios, that can be reusable. Cognitive approaches for structuring knowledge seem to be the most efficient in terms of results of the provided apprenticeship (Mohan & al, 1991). They are also the most complex and necessitate the installation of a methodology of development. Very few ITS have been developed for industry. Generally they have been oriented to mathematics, programming, or troubleshooting detection (Lesgold, Lajoie, 1990). The project described in this paper is an example of ITS development for industrial use with realistic objectives (Gecsei & Frasson, 1994).

According to our approach, a curriculum is a set of five structures :

- the model of **capabilities** which represents the organization of the knowledge,

- the model of **objectives** which represents the learning objectives,

- the model of **resources** which represents the organization of the material used to support learning,

- the pedagogical model which is an intersection of the two first models and exhibits the links between objectives and capabilities,

- the didactical model which is an association between objectives, capabilities and resources.

Figure 1 represents a pedagogical model. Capabilities C1, C2 and C3 are prerequisites to objective O1 which contributes to the acquisition of capabilities C4 and C5. C1, C2 and C3 are the input capabilities for objective O1, and C4, C5 are the resulting capabilities. To each prerequisite link (input link) is associated an input level which is the level required for capability C to meet the conditions of objective O. To each contribution link is associated a weight which precise the contribution of reaching objective O to the acquisition of capability C.





Figure 1. The pedagogical model

A course is a training program which aims at reaching several objectives. A course can be described by a list of capabilities that the learner should master or a list of objectives to reach.

3.2 Pedagogical approach

We do not aim at deeply intelligent tutoring; this would be somewhat unrealistic, given the limited success and the problems such systems have even within narrow application domains. Instead, SAFARI provides a combination of some existing tutoring tools, enabling to build easily and rapidly a tutoring system for a given device or a problem solving situation.

We use a tutoring cycle derived from the experiments resulting from the literature (Gagne, 1984). In fact, we observed that a **natural cycle** in which most people acquire a given skill is by first *observing* someone's demonstration of the skill, then freely *experimenting* with the device in question (given the availability of the device, and that such experimenting is not hazardous), then *executing* precise tasks (assignments) in terms of the device functionalities under the guidance of an expert, and finally by *communicating* the learned skill to another person. So we distinguish the following four *tutoring modes:* demonstration, free exploration, coaching and explaining modes. What is particularly interesting, for a question of modularity, is that the functionalities attached to each mode can be organized in a layered architecture (Frasson et *al*, 1996).

Indeed, in SAFARI, it appeared that a same activity could be presented according to different modes of teaching, with different responsibility between the learner and the

system. Thus presently, five modes have been envisaged which are based on simulation of activities :

- **demonstration** : the system presents a simulation of various tasks to the learner. The realization of the tasks incumbs totally to the system without intervention of the learner (for example, a situation where the learner observes the computer solving a problem),
- **free exploration** : the learner can navigate into a simulation system which reacts to his actions without intervention or guidance of the system. The learner controls his activity (for example, the resolution of a task) and this mode can be compared with free navigation within a hypermedia document,
- **advice** : the learner is in a problem solving situation and can benefit from advices of an adviser who continually watches the tasks and can correct the actions with in depth explanations. Various types of guidance (on demand, automatic, with multiple explanations,...) can be obtained,
- **critic** : the learner gives the right solutions within an activity and is requested for explaining part or all the elements of the solution. This will force the learner to structure his knowledge and adapt the explanations according to the different steps of the solution, providing the advantage to stengthen his own knowledge acquisition (Chi et VanLehn, 1991).
- **curriculum**: the learner enters in a learning session, through a complete course with problems, exercises and evaluation of different activities. The course is given using a variety of learning strategies that can be selected according to the learner's model.

These modes can be carried out by different *agents* that can be simulated. In fact the approach we have selected is to turn to a distributed pedagogical knowledge system, using tutoring agents that can play different roles according to the conditions of learning. These conditions are determined by the learner's actions which are analyzed in the learner's model.

The following table (Figure 2) resumes the different tutoring agents which can be applied in a tutoring mode. We successively distinguish the *tutor* who gives a course according to a prescriptive approach, the *co-learner* who is a simulated learner with approximately the same level of knowledge than the learner, the *companion*, a simulated learner who can give advices, the *inverted tutor* played by the tutor who is waiting for explanations from the learner, the *troublemaker*, a particular companion who can randomly give true of false advices.

The originality of our approach in building a generic ITS is to allow the use of multiple *tutoring strategies*, a strategy being a mode associated with a tutoring agent.

3.2.1 One-on-one learning (Tutor)

This approach preceded the co-operative systems and consisted in simulating the computer as an intelligent tutor who can understand the learner and provide adaptive tutoring. The learner receives knowledge directly from the tutor who communicates and acts according to a prescriptive behavior. Most of traditional ITS adopt this approach with adaptive features more or less marked according to the complexity of the student model used to provide feedback. In this case the teacher's knowledge is without doubt higher then that of the learner.

As alternatives to one-on-one strategy, co-operative strategies comprise an additional element, namely *peer interaction*. Co-operative learning systems, called also social learning systems, adopt a constructive approach based on the use of the computer more as a partner then as a teacher in the process of knowledge transfer. Multiple agents that are either computer simulated or real human beings can work on the same computer or share a computer network.

3.2.2 The companion

The idea of introducing a *co-learner* in the learning process arose with the perception that knowledge should result more in a building process than in a transmission process (Gilmore & Self, 1988). In this scope, the learner could co-operate with a co-learner having quite similar objectives and level of knowledge. A learner is inclined to more easily understand explanations given by a co-learner, who has understood, knows what to do and to answer, than the teacher. The co-learner is supposed to have recently passed through the same understanding problem and so is more aware of the level of explanation and detail to give to solve the problem. The knowledge level of the co-learner is slightly higher than the learner. Chan and Baskin proposed a three-agent learning situation (Chan & Baskin 1990) which consists in a co-operation between a human learner and a simulated learning companion who learn together under the guidance of the teacher.

The *companion* and the learner perform the same task and exchange ideas on the problem. The learner and the co-learner (the companion) work together and ask the teacher for help only if they cannot find a solution. The role of the teacher is then to alternatively present problems and critiques of the learner's solution. The process is gradual in the sense that each learner produces a solution then checks the other's solution. Finally the teacher checks the solutions which are submitted to him in order to correct any remaining error. The companion and the learner have quite similar knowledge levels, while the tutor has a higher knowledge level.

Modes	Exploration	Demonstration	Problem solving	Problem solving
Tutoring agents			Advice	Critic
Tutor		The learner observes the demonstration realized by the tutor.	The learner works, the tutor gives advices.	After its work, the learner observes the judgment of the tutor.
Co-Learner		The learner observes the demonstration realized by the co- learner	The co-learner gives advices to the learner during the session	The co-learner criticizes the learner a posteriori
Companion		The learner observes the demonstration realized by the companion or the tutor.	The companion, under the control of the tutor, gives advices to the learner during the session.	The companion or the tutor can criticize the learner's activity
Inverted tutor	The learner explores and explains his activity to the tutor			
Troublemaker		Under the control of the tutor the troublemaker gives (right or not) the demonstration.	The troublemaker gives advices (true or false).	The troublemaker criticizes the learner to test his self-confidence a posteriori.

Figure 2 : Tutoring agents and tutoring modes

3.2.3 Inverted tutor

An additional form derived from the learning companion was also proposed by Chan & Baskin (1990). The idea was to encourage the human learner to teach the companion, by providing examples, explaining why the solution given by the companion is not adequate. The approach is called learning by teaching and has been further elaborated by other

studies (Palthepu et al, 1991, Van Lehn et al, 1994). Explanation of this approach can be found in the learning theory of Gagné (Gagné, 1984) who shows that a strong knowledge acquisition is achieved when a learner is able to fully explain the solution of a task using his own inference mechanism and this last exercise is in itself a knowledge acquisition method (Chi & Vanlehn, 1991).

3.2.4 Learning by disturbing

This learning form suggests that the computer can be simulated as two agents : a teacher and a troublemaker (Aimeur & Frasson, 1996). The competence level of the troublemaker is superior to the learner in order to provide a reasonable competition with the learner. A problem is submitted both to the learner and the troublemaker. The troublemaker can have different behaviors : give a wrong answer to the problem in order to force the learner to react and propose the right solution, wait for the solution of the learner and give a wrong suggestion or solution or a counter-example, or sometimes give the right solution. The learner explains the troublemaker under the control of the teacher. If the learner is unable to give a correct solution the teacher gives him the right solution. This method forces the learner to take self-confidence in his/her actions or conclusions and distinguish between wrong and correct solutions. In addition, it strengthens the knowledge acquisition process. The learner confronts the troublemaker and is faced to his own knowledge and needs to prove that he has assimilated a right knowledge. Ultimately, he would feel a certain pleasure to give proof of his capacity in front of the troublemaker.

3.3 Learner model

The learner model deals with the knowledge on the learner's knowledge. It is an overlay of the curriculum of the three networks represented by the objectives, the capabilities and the resources.

The learner model is made up of three parts :

- The cognitive part is in fact an overlay of the capabilities of the learner compared to the capabilities in the curriculum. They concern general knowledge, capabilities of memorization, learning by association, speed of problem solving, domain knowledge, inductive reasoning.
- The affective part concerns several parameters such like attention, rapidity, anxiety, motivation, confidence, self-confidence, good or bad self-appreciation.
- The conative part allows to classify a learner within the following categories : thoughtful, impulsive, holistic, serial, spatial, verbal, exploratory, independent, conformist.

Initialization of the learner model is realized by submitting the learner to a set of tests. Then, the model is updated each time an action is performed by the learner in problem solving situation.

3.4 Development process

The project was developed according to different phases with precise tasks and schedule. They respect the pedagogical approach mentioned above in 3.2. and so, we successively developed the modules able to provide demonstration, exploration, execution of a task, and course consultation. Figure 3 shows the main components that have been developed to realize this development process.

• The first phase aims to develop a prototype of a *demonstration* generator capable of illustrating the functioning (tasks) and the capabilities of a device. Present experimentation is achieved in medical domain with a device used in intensive care unit, the Baxter pump. This pump provides liquid injection to patients. A *simulation* of the device has been realized in VAPS (Virtual Application Prototyping System) and various editors (tasks, scenario, problems)) were developed both in VAPS and Smalltalk. This experiment required a strong cooperation between the research team and the General Hospital in Montreal (Azevedo et al, 1996).



Figure 3. Development process

• The second phase aims to develop an *exploration environment* in which the learner can directly use the simulated device to practice, solve problems and exercises. A complete *cognitive task analysis* was carried out in order to build an effective problem

space and a cognitive task editor was set up. Various *scenarios* were constructed and stored in a database of scenarios. The actions of the learner are intercepted and compared with the tasks in the database and advices can be given, depending of the *tutoring strategy* that is selected. At that time of the project several different algorithms were developed for the strategies.

- Phase 3 consists in building a complete *course* using the *curriculum* and all the tools attached with this component. A planner and the learner' model are used to guide the training.
- Phase 4 aims to develop intelligent features able to follow the actions of the learner in a problem solving situation and choose the best time to interrupt the learner and to give him/her an adapted advice. The *adviser* (Pachet et al, 1995) which resulted from this phase is able to explain when the actions of the learner are wrong and what solutions should be chosen.
- Phase 5 aims to develop a complete environment for *cooperative learning*, using cooperative tasks editors.

Presently we have realized the first 4 phases and about 15 prototypes have been developed.

3.5 Architecture retained

After completion of the different phases we finally reached a common decision on the architecture to adopt for future ITS development. This architecture (Figure 4) contains the following modules :

- the curriculum module with all the editors. This component is essential for providing a complete course. The description of a course include the different objectives to reach and all the didactic links useful for completing these objectives. An important feature of the curriculum is its ability to generate a course adapted to a target public description. In fact, using the objectives and supposed capabilities of this target public, the curriculum builds the course by selecting the appropriate nodes to present to the learner.
- the planner generates a lesson using the characteristics of the learner in the learner model and the available time. It also uses a course plan given by the curriculum and including a sequence of objectives for the learner, capabilities and resources involved.
- the learner model allows initialization, consultation and update of the information of the learner. The model also includes a propagation model able to control the evolution of knowledge on the learner.

• the session manager is the core of the system. It takes all the tutoring decisions and the control of learning. The first action of the session manager is to ask the planner for a lesson. Then it can choose activities to undertake for each objective of the lesson. This leads to activate didactic resources that contains their own management system or call a pedagogical strategy which will use adidactic resources. The session manager builds a state of the learning advancement of the learner and dynamically updates the learner model.



Figure 4 Architecture of the system

- the didactic resources are tactical means to support a learning session. It can be a demonstration, an exercise, a problem, a hypermedia or multimedia document, a HTML document. They contain their own management system (including evaluation of the resource execution).
- the pedagogical actors are responsible to set up a pedagogical strategy using an advanced approach based on actors. Actors in SAFARI are intelligent agents able to react to a situation, to control it, but also to learn new ways to cope with new situations. The different actors that can consult each other are the tutor, the

companion, the inverted tutor, the trouble-maker. Actors use adidactic resources which do not contain their evaluation system. The evaluation of the resource activation is made by the actors in a given strategy and communicated to the session manager for subsequent decisions.

• the interface assume the communication between several modules (session manager, pedagogical strategies, didactic resources) and the learner in a learning environment.

3.6 Problems during the design process

Researchers were forced to design their own system architecture, implement all system components, develop knowledge representation strategies and reasoning mechanisms, and acquire and encode all relevant domain and instructional knowledge. While one or more of these tasks may indeed be focal research concerns, others are merely drudge tasks needed to erect enough infrastructure to allow work to proceed on the core of problem. To improve the transferability of the prototypes, a first line of developers was formed by the students using application manager, a utility program capable of structuring the different classes so that they can be reused by the team. The analysts of the team constituted the second line. They maintain the different versions of the prototypes and integrate the students works into a realistic prototype.

Implemented ITS systems typically are large programs designed under assumptions of plentiful computing resources. They may require special-purpose hardware, and/or are implemented in languages that may limit delivery to certain types of platforms. As desktop computers become interchangeable, ITS developers will need to respond with multiplatforms architectures and tools.

The two main programming languages used for the development of prototypes were VAPS and Smalltalk. Interpretation of the learner's actions and the different knowledge bases are done in Smalltalk. In particular, the knowledge base was first implemented in the Smalltalk file organization then moved to an object oriented database in Gemstone. We also have implemented our prototypes on SUN workstations (UNIX) and on PC workstations (Windows). We striked one problem with the simulation part, in VAPS. Its portability was not immediate and required some transformations.

Software tools exist both in the marketplace (word processors, spreadsheets, drawing tools, editors) and in research labs but these tools cannot be easily imported and used in ITSs. We have been obliged to develop specific editor to manipulate specific elements such as knowledge elements. However, we adopted a modular architecture in order to eventually include in the future standard software tools. The effective problem space was built after a cognitive task analysis which was made « by hand », resulting from interviews. The process is long to acquire.

We are presently experimenting new learning strategies (Aimeur et al, 1995), however it can be exceedingly difficult to ascertain the strengths and weaknesses of individual tutoring approaches, to identify all units across domains, and to share instructional materials and tutoring strategies. The process of measuring and testing the efficiency of new strategies is long and implies to compare fixed experiments but we consider that this approach is very important and that new strategies must be strongly tested and validated before including them in a real system (Aimeur & Frasson, 1996).

4. Results

Several prototypes have been developed (15) and presented to the industrial partners for validation. Several industries are already interested to acquire and use the SAFARI technology for their own training department. In fact it is possible to eventually use parts of the system due to its modularity.

More than one hundred publications were written and about 25 students (at Ph.D., postdoctoral and master level) have graduated. Several students have immediately found a job, sometimes a few months before having finished their master thesis.

As a side effect the experience of members of the group increased and is now widely recognized. They are now members of the Canadian Center of Excellence in Telelearning and they have organized the third international conference in Intelligent Tutoring Systems (Montreal, June 1996).

The main contributions of the project are in the following aspects

- the development of the curriculum components (Nkambou et al, 1996) allowed to include various pedagogical expertise (such as Gagne, Merill (Li & Merill, 1990), Bloom (Bloom et al, 1956) points of view). In particular it is possible to automatically generate a course from a first curriculum and a list of objectives,
- various editors (physical, functional, operational, for the adviser and the critic, for the curriculum) have been completed and can be used separately or in combination,
- a multistrategy approach has been implemented and partly experimented. Experiments are under way to validate new strategies (Aimeur et al, 1995).

The cost of development was under control. There is no difference between the initial and actual budget. The objectives fixed initially in the planification were all reached. The evaluation of prototypes was realized by nurses and also a physician who was in the laboratory to test the different prototypes in medicine (Intensive care unit). A version of the prototype was also given to the industrial partners for validation.

The majority of components of the prototypes are reusable. However they should be recoded into C++ for more efficiency and portability. In fact the most important problem

of portability lies in the presence of two languages : first VAPS which is the simulation system and Smalltalk which is the development system. It is however possible to use other simulation packages connectable to Smalltalk using a C interface.

The other problem of portability is the availability of the prototypes under Windows 95 platforms. As Smalltalk presents the advantage to be easily portable on a UNIX platform, on a Macintosh or on a PC station, the majority of our prototypes are also available on a PC Windows station and they can communicate between a UNIX prototype and a Windows environment in a client server protocol.

5. Lessons Learned

To confront our prototypes and also the orientation of research we have organized a one day presentation with industry. About 60 enterprises attended this presentation. We received many positive comments but also we were surprised of the high rate of people requesting for the availability of SAFARI on low cost platforms such as PC 386. It seems difficult to include all the intelligent aspects of the prototypes (for instance reasoning about the learner's actions) on such an environment. It appears more feasible to put the intelligent components on a powerful workstation and run the simulation and interactive parts on a client that can be a low cost station.

The need for training tools is increasing but we have not yet evaluated the impact of such tools in the workplace. We do not know the possible alteration of skill but we are confident in the necessity of these training tools. A simulation environment with interactive capabilities could be a solution to catch the interactions, comments and improvements for later interpretation.

Co-operating with industry is an exciting challenge, forcing the research to be situated at two levels of effort : a long term effort (for PH.D. works for instance) and a short term with immediate goals and results. An enormous advantage of this approach is to implement first, test and validate a research work before publishing papers. We know now that some research tracks cannot be successful at large scale due to the experiments we have realized. The orientation of some academic papers on the same track (dead end) show us the importance to make experiments the closest to reality in order to distinguish which tool or approach is definitively acceptable.

We also have been faced with the need to persuade people (demonstrations for potential clients) of the interest of the prototypes. Even if this approach is generally used in the industry it is relatively new for academic people for which it can appear as a lost of time.

A difficult point concerns the choice of the development language. If the language is close to the implementation procedure (like C++) the prototyping process might be long. Developing with Smalltalk provides the facility to obtain a prototype in a reasonable time. The effort of adapting the prototype to a more efficient final version remains to the industrial partners. In addition to the development language is the problem of storage organization. Initially stored in Smalltalk files the knowledge bases were progressively redesigned for relational databases using Access.

The project in still in development for the last phase which tackles cooperative learning, new learning strategies and advanced tools for design and control.

Acknowledgments : we thank the Ministère de l'Industrie, Commerce, Science et Technologie du Gouvernement du Québec for his support of the project. We also thank all the members of the SAFARI project who contributed to the development of the different parts of the architecture.

References

Aïmeur, E., Frasson, C. « Analyzing a New Learning Strategy according to different knowledge levels », *Computer and Education, An International Journal*, 1996 (to appear).

Aïmeur, E., Frasson, C. & Alexe, C. (1995). Towards New Learning Strategies In Intelligent Tutoring Systems. Brazilian Conference of Artificial Intelligence SBIA'95, Springer Verlag.

Anderson, J. (1988). NSF principal Investigator's Meeting, Phoenix.

Azevedo, R., Lajoie, S. P., & Bret, P.M.J. (1996). Instructional scaffolding during medical decision making. the 26th International Congress of Psychology, Montreal.

Bloom, B. S. et al (1956). Taxonomy of educational objectives. Mew York : David McKay

Chan, T.W. & Baskin, A.B. (1990). Learning Companion Systems. In C. Frasson & G. Gauthier (Eds.) *Intelligent Tutoring Systems : At the Crossroads of Artificial Intelligence and Education*, Chapter 1, New Jersey : Ablex Publishing Corporation.

Chi, M. & VanLehn, K. (1991). « The content of physics self-explanation ». Journal of the Learning Science, 1(1), 69-106.

Frasson, C., Mengelle, T., Aïmeur, E., Gouardères, G. (1996). An Actor-based Architecture for Intelligent Tutoring Systems, ITS'96 Conference, Lecture Notes in Computer Science 1086, Springer Verlag, pp 57-65.

Frasson, C., Gauthier, G. (1994). A Gradual Software Environment for Developing Tutoring Systems ». Dans J.W. Brahan, G.E. Lasker (dir. publ.) Advances in Artificial Intelligence — Theory and Application II (7th International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Allemagne, 15-21 août 1994), pp. 73-78.

Frasson, C., de La Passardière, B. (1990). A student Model Based on Learning Context, International conference on Advanced Research on Computers in Education, IFIP, Tokyo.

Gagné R.M., (1984). The conditions of learning, 4 ed, Les éditions HRW Ltée, Montréal. Gecsei, J., Frasson, C. (1994). SAFARI : an Environment for Creating Tutoring Systems in Industrial Training. EdMedia, World Conference on Educational Multimedia and Hypermedia, Vancouver, pp. 25-30. Gilmore, D. & Self, J. (1988). The application of machine learning to intelligent tutoring systems. In J. Self. (Ed.) *Artificial Intelligence and Human Learning, Intelligent computer-assisted instruction*, New York : Chapman and Hall, pp. 179-196.

Kaltenbach, M., Frasson, C. (1994). Intelligent Tutoring Systems : A Multidisciplinary Approach, Encyclopedia of Computer Science and Technology, vol 30, S-15, 1994, pp 287-300.

Lesgold, A.M., Lajoie,S.P., Bunzo, M. & Eggan, G., (1990). A coached practice environment for an electronics troubleshooting job. In J. Larkin, R. Chabay & C. Sheftic (Eds), Computer assisted instruction and intelligent tutoring systems : establishing communication and collaboration, Hillsdale, Nj, Erlbaum.

Li, Z., Merrill, D. (1990). Transaction shells : A new approach to courseware authoring. Journal of Research on Computing in Education, 23(1), 72-86.

Mcllraith, S., McCracken, J., Jones, M. et al. (1991). A report on the potential for improving government operations by applying artificial intelligence to training Ottawa : Training and Development Canada.

Mohan, P., Greer, J. E., Jones, M. L. (1991). Adapting instruction to the cognitive needs of learners in intelligent tutoring systems. In L. Birnbaum (Ed.), Proceedings of the International Conference on the Learning Sciences, pp. 322-328, Association for the Advancement of Computing in Education.

R. Nkambou, B. Lefebvre, G. Gauthier. (1996) A Curriculum-Based Student Model for Intelligent Tutoring System » accepted for presentation at Fifth International Conference on User Modeling, Kailua-Kona, Hawaï USA.

Pachet, F., Djamen, J-Y., Frasson, C., Kaltenbach, M. (1995). Un mécanisme de production de conseils pertinents exploitant les relations de composition et de précédence temporelle dans un arbre de tâches. Sciences et Techniques Educatives, Ed. Hermès, Paris.

Palthepu, S., Greer, J., & McCalla, G. (1991). Learning by Teaching. *The Proceedings of the International Conference on the Learning Sciences*, AACE.

Van Lehn, K., Ohlsson, S. & Nason, R. (1994). Application of simulated students : an exploration. *Journal of artificial intelligence in education*, vol 5, no 2, pp.135-175.