SAPIENCE: A Simulator of Interactive Pedagogical Activities for Distance Learning Environment

Tahar Bouhadada and Mohamed-Tayeb Laskri
Laboratory Research on Computing (GRIA/LRI), University of Annaba, Algeria

Abstract: *Several works on engineering Learning Active Situations (LAS) gave birth to tools for distance learning scriptwriting based on collaboration and interactivity. This paper describes a distance-learning environment that integrates a simulator for interactive pedagogical activities dedicated to the teaching of the compilation. The achieved prototype, SAPIENCE, is a framework based on controlled simulation. The device software uses the notion of interactivity with its different kinds in order to conceive LAS founded on the pedagogical script forms and the script monitor concepts. The teacher possesses tools enabling him/her to work out some LAS according to the learner's profile, to adapt the pedagogical strategies, to supervise the different registered learners in the different sessions in a personalized way, or in a group; and so, to control the learner's activities and to provide the most adapted pedagogical assistance.*

Keywords: *Pedagogical simulator, Learning Active Situation (LAS), pedagogical script, controlled simulation.*

Received February 18, 2006; accepted April 23, 2006

1. Introduction

The implementation of training devices on the Internet requires the use of a computer framework. This one can be standardize, conceived and merchandised by software specialized or non-professional companies, developed within the institution. The standard computer frameworks are reproducible software for a professional use in an industrialization process of computer software.

The inconvenience of these frameworks resides in their manipulation which makes difficult, perhaps impossible, the externalization of the content, which requires the use of supplementary software. The other disadvantage is that there are few frameworks using artificial intelligence techniques and automatic treatment of natural language [17].

Today, there are many works concern the exploitation of virtual classrooms, dematerialized spaces, where learners and tutors interact to bring the individual or collective training tasks to a successful conclusion. In the first instance, attention is concentrated on the use of frameworks of open and distance training Learning Management Systems (LMS). These technical systems provide the different training actors (learners, tutors, pedagogical designers, managers,...) with a set of functions that facilitate the setting up, and the exploitation of these new distance training situations [5, 11]. These environments enable the execution of training programs while integrating various pedagogical resources, the scheduling of the learner's activities, and the management of communications between learners and teachers. The present generation of distance-learning frameworks integrates a set of tools from a technical point of view, but does not place them explicitly in a pedagogical setting [12]. The present LMS are not adapted to all situations but implicitly encourage a determined type of training concentrating further on the proposed resources and tools than on the learner's activities [13].

Several authors [2, 12, 16] underline encountered difficulties to develop uses that differ from *standard* use characterized by:

1. A pedagogical model based on the transmission of knowledge.
2. Communication limited to relations between a tutor and an isolated learner.
3. An asynchronous mode of operation and exploitation.

Other types of training, based on constructive or socio-constructive approaches, are yet conceivable. The theory of activity and the theory of situated action respectively place the instrument and the group as supplementary poles to the classic triad learner-teacher-knowledge [1, 4]. Learning cannot only be conceived as a simple transmission of knowledge but must answer these two following requirements:

1. Learning must require a social interaction and collaboration.
2. Knowledge must be presented in an authentic context (i. e., requiring the implementation of this knowledge).

It hence seems interesting to apply these approaches to the distance-learning context mediatized by computer and to be able to propose learning situations that respect these two features [10].

Because of their difficulties, the current research works in this domain generally give priority to one or the other of these requirements. Tele-learning approaches by projects and more extensively, Computer-Supported Collaborative Learning (CSCL), give preference to collective learning, according to the generic meaning defined by George [7].

2. Active Learning Situation

Another research perspective aims to set Active Learning Situations or Learning Active Situations (LAS) in which the learner interacts closely with a device, which induces him/her to a discovery-construction of knowledge [9]. Such approaches correspond to the use of tele-manipulations [13], of micro-worlds [12], of simulations [1, 17], or of Interactive Learning Environments (ILE) in autonomy, as they are for example exercised in the learning of languages [11]. By the generic term of Interactive Pedagogical Object (IPO) Guéraud designates the software device support of this type of learning [10]. As opposed to an expositive pedagogical object, i.e. dedicated to knowledge transmission, an IPO includes a computing model able to modify its internal state and its representation according to the learner's manipulations. To encourage the setting-up of learning active situations, we must assist the teachers in their different tasks.

The first difficulty consists of the creation of the LAS. Indeed, if it is relatively simple for the teacher to propose expositive documents (completed by an automated evaluation as QCM type), it is more complex to conceive and to implement active situations based on the IPO. The second difficulty concerns the exploitation of the learning active situations. It is about providing their implementation for the learner and to facilitate the corresponding tutoring. In general, today, the importance of tutoring in the success of distance learning is recognized, in particular because it enables to break the isolation of the learner and to maintain their motivation. If we want to encourage the implementation of distance learning, efforts must therefore be accomplished to facilitate the tutor's task without decreasing the follow-up quality.

The objective of the present work is to define a software environment that enables the implementation of LAS based on simulation in a virtual classroom context. The virtual classroom term here designates a group of people registered on a distance-learning program, and who interact with each other and/or with their teacher during their learning [8].

In this work, we privilege the teacher's point of view. Assisting the teacher for designing the learning active situations is possible because of models and tools provided by previous works [10]. The pedagogical script concept is a key element for them. The pedagogical scripts specified by the teachers enable the implementation corresponding learning situations in a virtual classroom. They also enable to assist the teachers in their task of supervising group activities. The follow-up is considered not only in a synchronous but also in an asynchronous manner. The achieved prototype, designated by SAPIENCE (Simulator of Interactive Pedagogical Activities for a Distance-learning Environment dedicated to the Compilation), is a software environment enabling the implementation of Learning Active Situations based on controlled simulation. The obtained framework is achieved around a simulator of interactive pedagogical activities adapted to the resolution of controlled works dedicated to the learning of syntactic analysis. The simulator uses artificial intelligence techniques and calls forth the notion of interactivity during the resolution process; by which every learner has the possibility to validate every stage before he/she goes to the next one.

By simulator of pedagogical activity we designate the device software that uses the notion of interactivity with its different forms, in order to get a Learning Active Situation based on concepts of pedagogical script and script monitor.

3. Learning Approaches and Pedagogical Simulation

An interesting classification consists in taking into consideration the learning logic proposed by pedagogical software. Thus, two main approaches will be distinguished as shown in Figure 1.

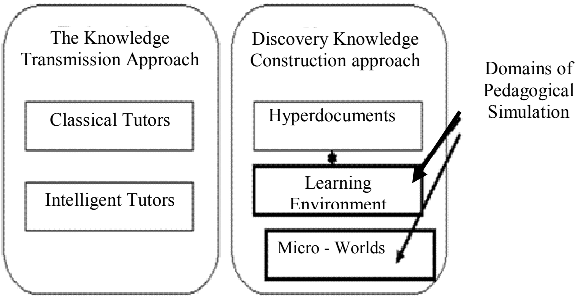

The Knowledge Transmission Approach
Discovery Knowledge Construction approach
Domains of Pedagogical Simulation
Classical Tutors
Hyperdocuments
Learning Environment
Intelligent Tutors
Micro - Worlds

Figure 1. Learning approaches and pedagogical simulation.

The first approach, called knowledge transmission, regroups the tutors of classical tutoring systems and Intelligent Tutoring Systems (ITS). The first is based on a decomposition of knowledge in elementary units whose linking is pre-established, on an evaluation of acquisition after the presentation of every unit, and on a progression conditioned by the given answer. The second, the ITS try to adapt the transmission of knowledge to the learner's features. This adaptation is achieved by explicitly integrating in the software, different expertises respectively concerning the domain, the pedagogical strategies, the learner and the human-machine interfaces [17]. The second approach, called discovery-knowledge construction regroups the hypermedias, the learning environments and the micro-worlds.

The hypermedia concept (or hyperdocument) allows the immersion in a sea of information where the learner navigates according to an associative process [17]. This type of software captures a remarkable interest at the present time, particularly because of the web expansion. The pedagogical use of hypertdocuments generally induces learning by discovery: the learner thus has the task of giving meaning to the path he/she creates through the proposed knowledge. The micro-world concept, has first, based its approach on a pedagogical innovation processes. Indeed, Papert proposes to offer an open learning environment in which the learner can develop his/her own models of the real world in order to experiment them by simulation. Micro-worlds thus allow the learners to develop and to correct their own theories [10]. The notion of learning environment is more based on the simulation of a model than on its construction. The learners acquire knowledge while modifying parameters and while observing the after-effects of their actions in the simulated environment [17].

These different types of environments can be proposed conjointly. It is indeed interesting to imagine that the knowledge proposed in a hyperdocument is reinforced by simulations when it is pertinent.

Inversely, learners who manipulate a simulation and perceive that they perhaps do not control the necessary knowledge will advantageously be able to search for the knowledge they lack inside a hyperdocument [17]. Simulation thus brings forth a very important contribution to the training domain. However, a simulator is not always appreciated especially if it threatens the role of the teacher. Indeed, as learning is taken over by the simulator and that this one evaluates the learners, teachers feel themselves dispossessed of their role. They must reorganize their way of work. The simulator reveales the mistakes committed by learners in continuous training, therefore, the simulator is an indispensable training tool.

4. The Organization of the Simulator

The simulator essentially includes four phases as shown in Figure 2. We can say that in computer science, simulation can be used to understand, to create or to learn. We focused our attention on simulations whose ostensible goal is learning.

- *Initial Phase*: Fix a goal or determine the objective of the activity.
- *Analyze*: Before acting concretely and to be able to do it, the learner needs to understand the proposed problems, to acquire the adequate theoretical notions (to consult the course), etc.
- *Act*: The learner achieves his/her tasks.
- *Evaluate:* The learners evaluate the actions that they did, and see if they are enabled to reach the attainment of the objective, otherwise they go back again to Phase 2. This evaluation phase is reinforced by a final global evaluation of their work, which is proposed to them at the end of their activities (assessment of activities).

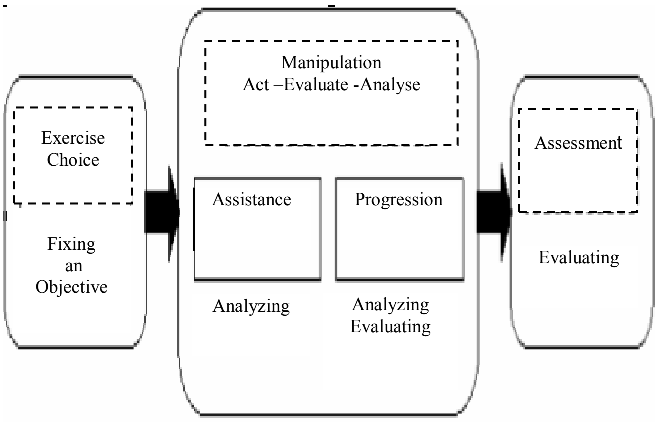


Figure 2. The general organization of a pedagogical simulator.

5. Learning Active Situations (LAS)

Pedagogical simulations are softwares that propose LAS based on simulation. In the present work, we are interested in a distance-learning context to implement learning active situations (the student's controlled works) exploiting a simulator of interactive pedagogical activities. It consists in giving teachers the tools to propose and to control the learner's activity, and also to provide, the different actors, with means of communication and exchange.

5.1. The Conception of LAS

For the development of LAS, we adopted the Model, Association, Script, Representation model (M.A.R.S) [17]. This model is a general conceptual setting that guides the designer in his/her process of defining the pedagogical simulation application. Its object is to make palpable the notion of distinct work spaces in which intervenes every actor, provided of well listed expertise. Its goal is, therefore, to serve as a reference in organizing of the software development of an application [10].

M.A.R.S model decomposes a pedagogical simulation into four workspaces as shown Figure 3:

- *Model Space (M)*: Describes the simulated system.
- *Representation Space (R)*: Defines the interfaces manipulated by the user.
- *Association Space (A)*: Permits the integration of the Model and the Representation in order to get a simulation.

- *Pedagogical Script Space (S)*: A Script is what directs, during the time of a session, the control of the learner, so we call it controlled simulation.

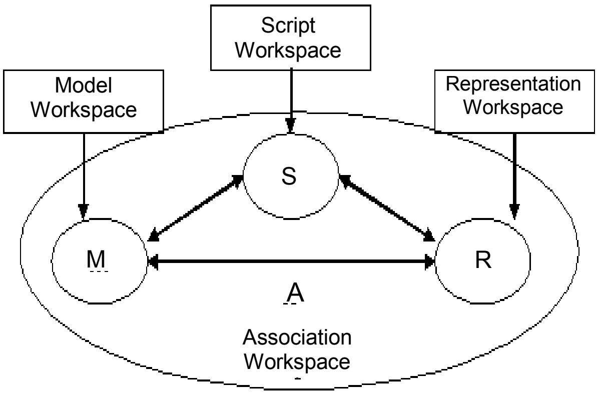


Figure 3. Workspaces in M.A.R.S model

5.2. The Progress of a Controlled Simulation

To construct a real learning situation, we introduce the generic term controlled simulation to designate the device software of this type of situation. This simulator must be centered on simulation and interactivity, and must be integrated in a Human Learning Computer Environment (HLCE), which takes into account the interactions between the different actors (teachers, learners, pedagogical designers, administrators, ...).

SAPIENCE, which is based on the M.A.R.S model, rests on two concepts to get a controlled simulation as shown in Figure 4.

- A pedagogical simulator.
- The pedagogical script that permits the specification of the proposed activity on the simulator.

The script monitor is an execution supervisor placed on every learner's station. It is responsible for exploiting the simulator's properties to get information about the learner's progression; it is designed to be in charge of verifying the learner's progression with regard to script's stages; and it is again in charge of transmitting the follow-up information to the teacher's station.

A script includes various stage descriptions: the objective to be reached (or the problem to be solved), a sequence of resolution stages, each of them with its objective, its initial state, its expected final state and its intermediate controls, and information about the feedback to the learner (feedback in case of success or in case of failure).

During its execution, the script permits the observations of results concerning the learner's behavior. The LAS, thus specified, are implemented for learners because of the script monitor.

In SAPIENCE, the simulator is the software on which the learners have to work for achieving the different stages of an exercise in order to accomplish their objective. It contains a calculation model capable of modifying its internal state and its representation according to the learners' interventions.

The simulator is characterized by the following properties:

- *Inspectability*: we must be able to consult some variables of the simulator from outside. This allows the script monitor to detect some particular states and to validate the different stages of the resolution process.
- *Scriptability*: we must be able to change some variables from outside. This allows the script monitor to place the simulator in an initial state defined beforehand.

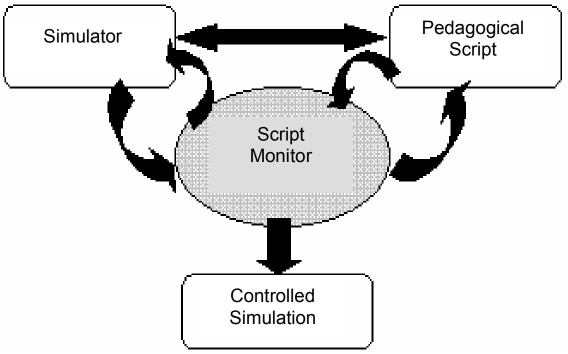


Figure 4 . The progress of a controlled simulation.

6. The Software Architecture of SAPIENCE

The system has been conceived according to the three levels Customer/Server architecture (3-tiers Architecture): customer level, data and applications server level and web server level as shown in Figure 5.

- *Customer level*: It represents the different services required by a customer, whether learner or teacher.
- *Web Server level:* It constitutes the interface between the customer and the application server while transmitting the customer's requests toward the application server, and the service achieved by the latter toward the customer.

It is an interface of communication between users and SAPIENCE, between learners and teachers and finally between the simulator and the script monitor.

- *Data and applications server level*: It represents the different services of data management offered to customers (teachers, learners). To be able to answer the different requests expressed by different customers, the data server has at its disposal a set of databases:

- A *Learner' base* that contains personal information about the learners.
- A *Teachers' base* that contains information concerning the teachers.
- A *Profiles 'base that contains the historical account of different learners' behavior* during the different sessions.

- The *Courses base* whose structure is hypertext, which contains whole courses, and structured in chapters.
- An *Exercises base* that contains the list of exercises for every session.
- A *Sessions base* that contains the list of different sessions of directed works according to the three classes of grammar: SLR (0), LR (1) and LALR (1).

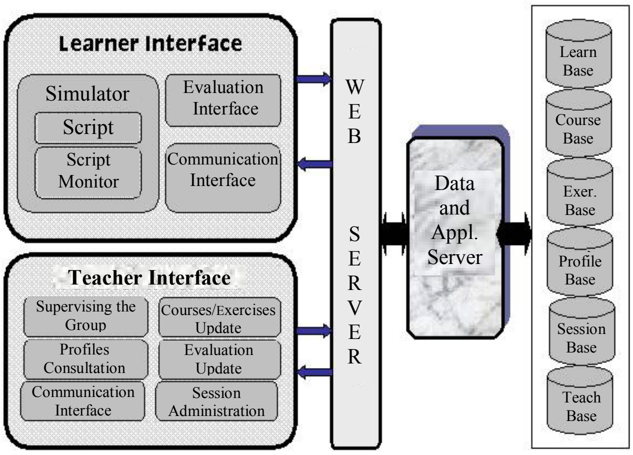


Figure 5. The structure of SAPIENCE.

7. Teaching Material

The simulator is dedicated to teaching the compilation. We first considered the syntactic analysis, taking into account the pedagogical strategy with which it is taught, and the quality of interactivity that enables the learner to intervene during the different resolution stages of the proposed exercise.

The system courses base is organized in chapters; each of them represents a state of knowledge acquired by the learner, and a set of concepts.

Courses are structured as a hyperdocument; every course incorporates definitions and examples that make explicit the different concepts. In the present version, the courses' base includes five courses distributed in five chapters as shown in Table 1.

Table 1. Courses contents.

Chapter Code	Chapter Title	Description of the Contents
Chapter 1	Introduction	Generalities
Chapter 2	SLR (0) Grammar	1. Definitions. 2. The Closing Operation 3. The Transition Operation 4. Construction of the Analysis Table. 5. The Items' Collection 6. The SLR Analysis Table 7. ……………………….
Chapter 3	LR (1) Grammar	1. Definitions 2. …………..
……..	………..	…………………………

An example of an exercise could be:

1. Find the collection LR (1).
2. Fill the analysis table by items found in step 1.
3. Launch the execution of the chain.

8. The Simulator of SAPIENCE

The learners must achieve with the simulator some tasks that are described in the script. It is an action interface which enables the learner to answer the directed works proposed in different sessions. To simplify the learner work, the simulator is reinforced by a pedagogical script and a script monitor to guide him/her in the learning process.

8.1. The Pedagogical Script

A script explicitly defines the different activities proposed to the learners on the simulator. It also specifies the control that will be made of the learner's progression during this activity; it determines the pedagogical assistance that will be provided to him/her automatically according to his progression [10].

For this effect, the interface of the simulator is provided with a panel including different descriptions of every stage of the work to be treated as shown in Figure 7.

A pedagogical script is defined by [10]:

- The initial situation and the objective to reach.
- Situations corresponding to the pertinent resolution stages.
- Particular situations to observe (constraints to respect, classical errors, potential dangers, ...).
- The reactivity that enables the assistance of learner according to his progression. It determines system reactions (feedback information, assistance, return to the first stage,...) corresponding to different controls (successful stage or not, particular reached situation, reached objective or not).

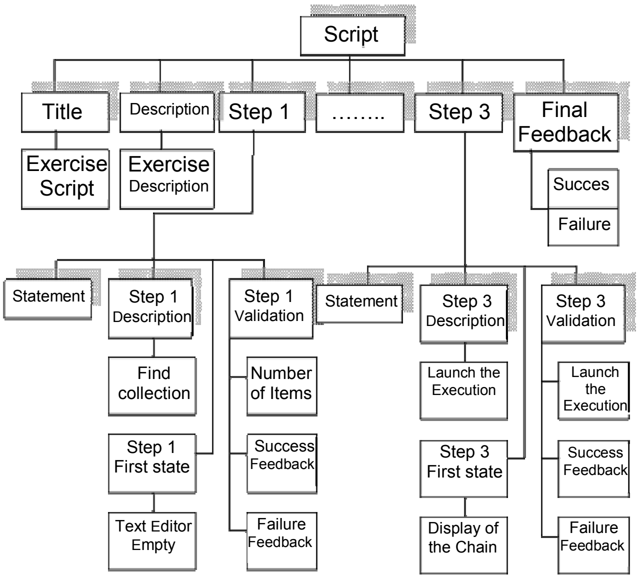


Figure.6. Arborescent representation of a script.

When it is executed, such a script does not force the learners to reach the objective according to a unique progression, but enables the validation of a set of solutions. Indeed, the teacher fixes the stages that he judges inescapable as well as their order; the system will verify if the situations corresponding to these stages will be achieved successively, but will not control the way that the learners proceed to reach the end of every stage. However, the system will be able to detect the particular situation reached by the learner if the teacher specified it as a situation to be observed.

During its execution, the script enables the observation of the learner' behavior results on the LAS. While marking a certain number of remarkable situations (clearing of stages, error situations,...), it automatically establishes a diagnosis of the learner activity. This diagnosis is located on a behavioral level. It describes behaviors without trying to discern the state of knowledge that the learner put into action [6].

Once defined by the teacher, the script enables the proposition of different work modes to the learner: "learning" mode or "evaluation" mode [19]. In the present work, we exploited the learning mode. In this mode, besides the global introduction, the learner receives the instruction of the firsst stage and can have access to an assistance provided for this eventually. When he/she thinks that he/she succeeded in this stage, the learner asks for the validation of his work and receives the corresponding evaluation. As soon as an intermediate stage is gone through with success, the order of the next stage is displayed. The learner can abandon a stage at any time and can go to the next (the system puts the simulation as if he/she had succeeded in the stage). If he/she reaches a situation "*to be* observed", the feedback information and the expected action are then launched.

The script includes various descriptions of stages described in an XML format. That pedagogical script model is defined by its title, a general statement, and the different resolving stages (one stage at least). The stage is considered like a sub-objective to reach and the learner must successively validate all the stages to succeed in the exercise. However, he/she can ask to abandon the current stage and go to the next. Feedback information is provided in case of global success of the exercise as well as in failure as shown in Figure 6 [8].

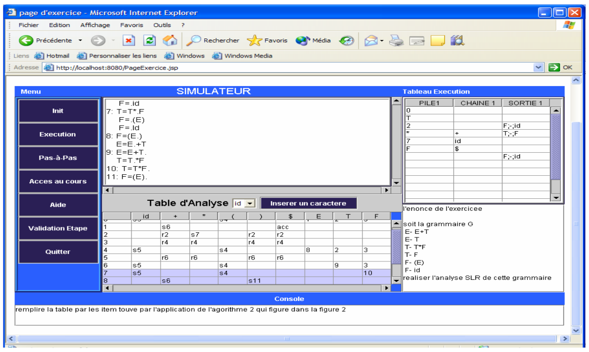
Every stage is characterized by:

- *A description*: Which briefly indicates the essential objective of the stage. It enables us to mark stages.
- *A statement*: Which constitutes the specific instruction of the stage.
- *An initial state*: It is used by the script monitor to initialize the simulator in the beginning of the session.
- *Features of the stage validation*: They contain a situation that enables us to validate the stage as well as the associated reactivity in case of success as in case of failure. When the learner asks to validate the stage, the script monitor may forward an evaluation evaluates if the current state of the simulator satisfies these conditions and thus determines if the stage is gone through successfully or not.

8.2. The script Monitor

The monitor enables the learner as shown Figure 7.

- *To Load an exercise*: loaded from an XML document.
- *To Initialize the simulator*: the simulator is set in the initial state of the first stage, because of its scriptability feature.
- *To Ask for a validation of an end of stage*: the monitor uses the inspectability feature of the simulator to compare the current state with the end stage indicated in the script. To avoid that the learner randomly falls on an end of stage, we oblige him/her to make an explicit validation request when he/she thinks that he/she has satisfied the objective of the stage. Success automatically provokes the passage to the next stage.



```
   F=.id
7: T=T*.F
   F=.(E)
   F=.Id
8: F=(E.)
   E=E.+T
9: E=E+T.
   T=T.*F
10: T=T*F.
11: F=(E).
```

	id	+	*	(	)	$	E	T	F
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				

PILE1	CHAINE 1	SORTIE 1
0		
T		
2		F;-;id
*	+	T;-;F
7	id	
F	$	
		F;-;id

l'enonce de l'exercicee

soit la grammaire G
E- E+T
E- T
T- T*F
T- F
F- (E)
F- id
realiser l'analyse SLR de cette grammaire

remplire la table par les item touve par l'application de l'agorithme 2 qui figure dans la figure 2

Figure 7. Screen "Initialisation of the simulator".

9. Implementation

The development of a distance-learning environment requires languages dedicated to the implementation of web applications. The realization of an environment according to the 3-tier architecture requires navigation, interpretation and very powerful communication tools.

The implementation of SAPIENCE required Customer-oriented languages and Server-oriented languages.

The system has been developed with Java (JBuilder 7) on the APACHE server's base and uses its PHP interpreter for the interpretation of different interactions. The generation of different pages is obtained with JSP. For the conception of courses, we used XML language, more adapted to the development of hyperdocument systems. Finally, for the management of the different databases, we opted for

MySQL, whose performances are especially indicated for this type of applications, with an access via JDBC.

10. The Human-Machine Interactions

The system includes an interface dedicated to the learner and an interface dedicated to the teacher (Figure.8)

The teacher's interface enables a teacher to manage the courses base, the exercises base, the questions base, to define the controlled work sessions, to consult the learner's profile, to follow the activities of a learner or of a group (Figure.9), to provide pedagogical assistance to the learner, to define the evaluation and to communicate with a learner or with a group.

The learner's interface enables the learner to consult a course, to choose a controlled work session (SLR (0), LR (1) or LALR (1)) as shown in Figure 10, to choose a session of tests, an evaluation or to communicate with another learner or with the teacher.

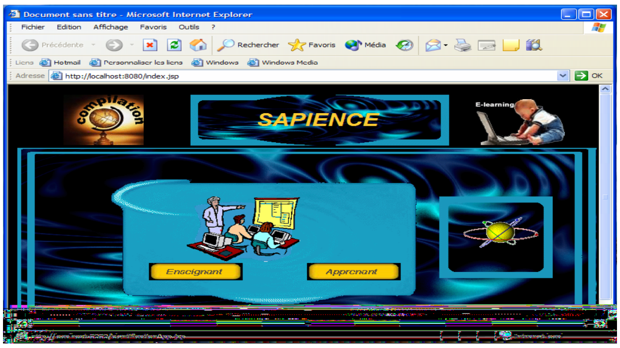
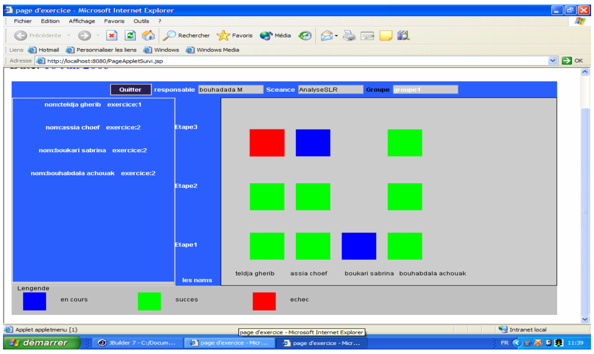

Document sans titre - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris Média
Liens Hotmail Personnaliser les liens Windows Windows Media
Adresse http://localhost:8080/index.jsp
compilation
SAPIENCE
E-learning
Enseignant
Apprenant

Figure 8. Screen "Home page".


page d'exercice - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris Média
Liens Hotmail Personnaliser les liens Windows Windows Media
Quitter
Applet appletmenu [1]
Intranet local
démarrer
JBuilder 7 - C:/Docum...
page d'exercice - Micr...
page d'exercice - Micr...
FR 11:39

Adresse http://localhost:8080/PageAppletSuivi.jsp

responsable bouhadada M Sceance AnalyseSLR Groupe groupe1

nom:teldja gherib exercice:1

nom:assia choef exercice:2

nom:boukari sabrina exercice:2

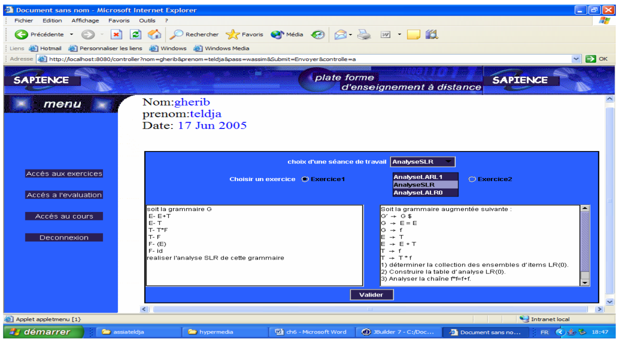
nom:bouhabdala achouak exercice:2

Etape3	🟥	🟦		🟩
Etape2	🟩	🟩		🟩
Etape1	🟩	🟩	🟦	🟩
les noms	teldja gherib	assia choef	boukari sabrina	bouhabdala achouak

Lengende

🟦 en cours 🟩 succes 🟥 echec

Figure 9. Screen "Supervision of a group of learners".



Adresse http://localhost:8080/controller?nom=gherib&prenom=teldja&pass=wassim&Submit=Envoyer&controlle=a

Nom:gherib
prenom:teldja
Date: 17 Jun 2005

choix d'une séance de travail AnalyseSLR

Choisir un exercice ● Exercice1 ○ Exercice2

AnalyseLARL1
AnalyseSLR
AnalyseLALR0

```
soit la grammaire G
 E- E+T
 E- T
 T- T*F
 T- F
 F- (E)
 F- id
realiser l'analyse SLR de cette grammaire
```

```
Soit la grammaire augmentée suivante :
G' → G $
G → E = E
G → f
E → T
E → E + T
T → f
T → T * f
1) déterminer la collection des ensembles d'items LR(0).
2) Construire la table d'analyse LR(0).
3) Analyser la chaîne f*f=f+f.
```

Figure 10. Screen " Loading an exercise".

11. Conclusion

We described a simulator of interactive pedagogical activities integrated in a distance-learning environment, and dedicated to teaching the compilation. The achieved system SAPIENCE adopts a collaborative pedagogical method that allows a constant solicitation of the learner, a permanent evaluation, a multiplication of paths, and multimedia tools that encourage the use of maximum learning channels implying a community of human and machine actors.

SAPIENCE, in its present version, offers tools for developing LAS based on controlled simulation, using a pedagogical activities simulator that employs the notion of interactivity and integral concepts of pedagogical script and script monitor.

For the teacher, these concepts represent a real asset to facilitate the exploitation of LAS. They enable him/her, in particular, to offer original functionalities of perception concerning the learner and group activity. The system adopts a 3-tiers Customer/Server architecture where human and software actors can communicate through the Internet network.

SAPIENCE, with its script controller, constitutes for the teacher an ideal tool to organize learner's activities in different sessions; every session is constituted of LAS, and enables the manipulation of pedagogical activities that are particularly reactive to learners' behaviors.

The ambition of the initiated project leads to an engineering of active learning situations to create helping tools to design scripts for distance-learning pedagogical resources based on collaboration. This requires complementary and pluridisciplinary contributions.

References

[1] De Jong T. and Van Joolingen W. R., "Scientific Discovery Learning with Computer Simulations of Conceptual Domains," *Review of Pedagogical Research*, vol. 68, no. 2, pp. 179-201, 1998.

[2] Després C., Modélisation et Conception d'un Environnement de Suivi Pédagogique Synchrone d'activités d'apprentissage à Distance," *PhD Thesis in Computer Science*, University of Maine, France, 2001.

[3] Engeström Y., *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research,* Orienta, Konsultit Oy, Helsinki, Finland, 1987.

[4] EIFEL Report, "Une Etude Européenne des Outils Pour la E-Formation," European Institute For E-Learning (EIFEL), France, 2002.

[5] Faerber R., "Accompagner les Apprentissages à Distance et Collaborer en Petits Groupes," *in Proceedings of the 4th AECSE Congres*,

Villeneuve d'Ascq, France, 2001.

[6] Galison A. and Nouveau J. S., “OASIF : Un outil Collaboratif d’aide à la Scénarisation de Modules de Formation Ouverte et à Distance,” *Télécom Paris*, Department of Educational Innovation, France, 2003.

[7] George S., “Apprentissage Collectif à Distance. SPLACH: Un Environnement Informatique Support d’une Pédagogie de Projet,” *PhD Thesis in Computer Science*, University of Maine, France, 2001.

[8] Gil P., *E-Formation NTIC et Reengineering de la Formation Professionnelle*, Dunod Editor, 2000.

[9] Guéraud V., “Pour une Ingénierie des Situations Actives d’Apprentissage,” *in Proceedings of the Environnement Informatique d’Apprentissage Humains (EIAH’03)*, Strasbourg, France, 2003.

[10] Guéraud V., Adam J. M., Pernin J. P., Calvary G., and David J. P., “L’exploitation d’Objets Pédagogiques Interactifs à Distance: le Projet FORMID,” *Sciences et Technologie de l’Information et de la Communication pour l’Education et la Formation,* STICEF, vol. 11, 2004.

[11] Hérino M. and Petitgirard J. Y., “Langues et Multimédia- de la Réflexion à la Pratique,” Repport of CRDP of Grenoble, France, 2002.

[12] Koper R., “*From Change to Renewal: Pedagogical Technology Foundations of Electronic Learning Environments*,” Open University of the Netherlands, 2001.

[13] Lelevé A., Meyer C., and Prevost P., “Télé-TP: Premiers pas vers Une Modélisation,” *in Proceedings of the TICE Symposium*, France, pp. 203-211, 2002.

[14] Leontiev A., “*Activité Conscience Personnalit*é,” French version, Progrès Editor Moscou, 1975.

[15] Ologeanu R., “Visioconférence Dans L’enseignement Supérieur: Le Processus D’innovation, Des Expérimentations Aux Usages,” *PhD Thesis in Information And Communication Sciences*, University of Grenoble 3, France, 2002.

[16] ORAVEP Report, “Etude Comparative Technique et Pédagogique des Plates-Formes Pour La Formation Ouverte et a Distance,” *Direction of Educational Technologies, Information and Communication (DT/SDTETIC)*, Ministry of National Education and Technology, France, 2000.

[17] Pernin J. P., “MARS: un Modèle Opérationnel de Conception de Simulations Pédagogiques,” *PhD Thesis in Computer Sciences*, Joseph Fourier University, Grenoble 1, France, 1996.

[18] Soury-Lavergne S., “Etayage et Explication Dans le Préceptorat Distant, le cas de TéléCabri,” *PhD Thesis*, Joseph Fourier University, Grenoble 1, France, 1998.

[19] Wenger E., *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann, Los Altos, 1987.

Tahar Bouhadada obtained his MSc degree in 1994 from the University of Annaba. Currently, he is a teacher-researcher in the Research Group. He has been working on the learning environment architectures and Intelligent Tutoring Systems (ITS) since 1990. He presented many papers about the Interactive Learning Environments (ILE), the Learning Companion Systems (LCS), and the learner modelling. He supervises many graduate works on learning environments, distance learning and web-based systems. He also teaches at the graduate level in computer science and data base systems. These last years, he has been working on simulator systems and the design of interactive pedagogical activities.

Mohamed-Tayeb Laskri obtained his PhD degree in 1995 from the University of Annaba. Currently, he is president of the University of Annaba. He is the chairman of the Research Group on Artificial Intelligence (LRI/GRIA). He supervises many works on learning environment, automatic treatment of the natural language, the users-case, human-machine interfaces, the coordination on multi-agents systems. He leads tasks at some national commissions on treatment of natural language, the distance learning frameworks and information technologies. He supervises many PhD thesis and several MSc students in knowledge management, distance learning environment. He also teaches modules at both graduate and MSc levels in computer science, expert systems, Intelligent Tutoring Systems (ITS) and artificial intelligence.