UNIVERSITY OF LYON - Claude Bernard
DOCTORAL SCHOOL InfoMaths
Informatique et Mathematique

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Lyon 1 - Claude Bernard
Specialty : Computer Science

Defended by

Aarij Mahmood Hussaan

Generation of Adaptive Pedagogical Scenarios in Serious Games

defended on December 19, 2012

Jury :

Reviewers :	Pascal Estraillier, Professor, Université de La Rochelle
	Jean-marc labat, Professor, Université Pierre et Marie Curie
Examinateurs :	Daniel Burgos, Professor, International University of La Rioja (UNIR) -
	Phillipe Revy, Speech Therapist, Société GERIP
Directors :	Alain Mille, Professor, Université Claude Bernard Lyon 1 (LIRIS)
	Karim Sehaba, Assistant Professor, Université Lumière Lyon 2 (LIRIS) -

Acknowledgements

I start by thanking the Almighty God for giving me the opportunity, the courage and the patience to successfully complete my doctoral studies away from my home.

I would also like to graciously and sincerely thank all the members of jury for accepting to referee my thesis defense. It was an honor for me to have people of such excellence as jury members in my thesis defense. Thanks to Prof. Jean-Marc Labat and Prof. Pascal Estrallier to have reported my thesis. Their invaluable feedback on my research gave me precious perspectives for my future work. I also would like to heartily thank Prof. Daniel Burgos for accepting to be a part of my jury despite being located in Spain. His remarks, questions and encouragements helped me getting further more motivated to continue my line of research. Special thanks to Mr. Phillipe Revy for being part of my research, he provided us with the necessary tools that were required to test my research on a real-world platform. His remarks on my work will always have a special place with me.

I would also like to express my gratitude to my supervisor Mr. Karim Sehaba. I sincerely thank him for accepting me to do a PHD under his supervision. It would have been hard for me to find a better supervisor than him. His dedication to my research will always have a special place in my heart. His encouragements and motivations provided me to successfully pass a very difficult period of time. At times he acted more like an elder brother than a supervisor. I know that I am not the easiest person in the world to work with, and for that I thank Mr. Karim Sehaba for persisting patiently with me. I can hardly imagine completing my research without him.

One of the most memorable experiences for me in France was working with my thesis's director Prof. Alain Mille. His profound knowledge about Artificial Intelligence was a constant source of inspiration for me. I also interacted with Prof. Alain Mille outside the scope of my research and always found him to be most helpful and humble. His constant and invaluable feedback on my research allowed me to complete my degree. I can safely say that he played a very big part in teaching me how to do research. I shall remain forever indebted to him for his contribution.

My research was also aided by the outstanding environment I was working in. This included my research colleagues. I would like to thank my friends Saleheddine, Olivier, Patrice, Amaury, Lemya, Charlotte, and all other colleagues who helped me in feeling at home in a foreign environment.

Last but definitely not the least I thank my parents. Everything that I have in my life I owe to them. They always sacrificed their needs to provide me with the best they possibly could. It was my parents who motivated me to go to a foreign country to do research. The one regret I will always have is that my late father and my mother were not with me at my PHD defense. My parents are the greatest gift to my on earth by God.

Abstract

A serious game is a game whose principal objective is other than only entertainment. In this thesis, we are interested in a particular type of serious games: the learning games. These games make the learning process more attractive and amusing through fun-based challenges that increase the motivation and engagement of learners. In this context, this thesis focuses on the problem of the automatic generation of pedagogical scenarios in the learning games. It is thus a question of apprehending the integration of a pedagogical scenario with computer games within the context of learning games. By pedagogical scenario, we mean a suite of pedagogical activities, integrated in a learning game, allowing a learner to achieve one or more pedagogical objectives. The objective of our research is to define representation and reasoning models allowing the generation of adaptive pedagogical scenarios which can be used in serious games, in particular the learning games. The generated scenarios should take into account the user's profile, pedagogical goals and also his interaction traces. The traces get used to update the user profile and to evolve the domain knowledge.

The proposed **knowledge representation model** allows organizing the domain knowledge in three-layer architecture: the domain concepts layer, the pedagogical resources layer and the game resources layer. For each of these layers, we have proposed an adapted formalization. The generic organization of knowledge allows evolving the elements of a layer without changing or affecting the elements of other layers. Similarly, it allows putting into relation the same domain knowledge with different games.

As for the **scenario generation model**, it comprises of three successive steps. Firstly, starting from the user profile and his pedagogical objectives, it generates a conceptual scenario. This consists in selecting a certain number of concepts, among the domain concepts of the first layer, allowing satisfying the targeted concepts. These targeted concepts represent the pedagogical objectives of the user. The conceptual scenario is then transformed into the pedagogical scenario. For this, it requires to select for each concept in the conceptual scenario one or many pedagogical resources in relation with the concept in question. This selection takes into account the presentation model and the adaptation knowledge. The former allows structuring the pedagogical resources according to their type. The adaptation knowledge allows setting the difficulty level for each pedagogical resource in the pedagogical scenario. The third and final step consists in putting into relation the pedagogical resources of the pedagogical scenario with the game resources keeping into account the game model.

On the basis of the proposed models of representation and reasoning, we have developed the **platform GOALS** (Generator Of Adaptive Learning Scenarios). It is a platform, generic and accessible on-line, allowing the generation of adaptive pedagogical scenarios. This platform has been used in the context of a serious game for the evaluation and reeducation of cognitive troubles within the context of the FUI project CLES (Cognitive Linguistic Elements Stimulations). To validate our contribution, we have conducted several evaluations in the context of project CLES.

The objective of these evaluations is two-fold; firstly, to validate the scenario generator models, secondly, to study the impact of the scenarios generated by GOALS on the learning of users. For these two objectives, we have proposed **two evaluation protocols.** These protocols have been put into practice in the context of **two field experiments**.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Research Context: Project CLES . . . 3
- 1.3 Research Objectives . . . 6
- 1.4 Research Questions . . . 6
- 1.5 Characteristics . . . 8
  - 1.5.1 Generic Nature of the Scenario Generator . . . 8
  - 1.5.2 Continuous Knowledge Acquisition . . . 9
- 1.6 Summary . . . 9

2 Related Work **13**
- 2.1 Introduction . . . 14
- 2.2 Definitions . . . 15
  - 2.2.1 Adaptive and Adaptable System . . . 15
  - 2.2.2 Pedagogical Scenario . . . 15
  - 2.2.3 Serious Game . . . 17
- 2.3 Scenario Generation in Serious Games . . . 18
  - 2.3.1 Authoring Tools . . . 20
  - 2.3.2 Game Based Learning . . . 23
  - 2.3.3 Dynamic Difficulty Adjustment . . . 24
  - 2.3.4 Summary . . . 26
- 2.4 Scenario Generation in AEHS . . . 28
  - 2.4.1 Course Sequencers . . . 29
  - 2.4.2 Course Generators . . . 33
  - 2.4.3 Summary . . . 39

3 Contributions **41**
- 3.1 Introduction . . . 42
- 3.2 Knowledge Modelling . . . 42
  - 3.2.1 Three Layer Architecture . . . 44
  - 3.2.2 Domain Concept . . . 46
  - 3.2.3 Pedagogical Resource . . . 50
  - 3.2.4 Game Resource . . . 53
  - 3.2.5 Learner Profile . . . 54
  - 3.2.6 Presentation Model . . . 57
  - 3.2.7 Adaptation Knowledge . . . 59
- 3.3 Scenario Generator . . . 60
- 3.4 Scenario Generation Algorithms . . . 63
  - 3.4.1 Concept Selector . . . 63
  - 3.4.2 Pedagogical Resource Selector . . . 68

3.4.3 Serious Resource Selector . . . 69
3.5 Learner Profile Updating Through Interaction Traces . . . 70
3.6 Formal Validation . . . 72
3.7 Summary . . . 74

4 GOALS: Generator Of Adaptive Learning Scenarios 75
4.1 Objectives of GOALS . . . 76
4.2 Different Types of Users . . . 77
4.2.1 System Administrator . . . 78
4.2.2 Domain Expert . . . 78
4.2.3 Learner . . . 78
4.3 Configuration of GOALS by the expert . . . 79
4.3.1 Projects Management . . . 79
4.3.2 Learners Management . . . 80
4.3.3 Knowledge Editor . . . 82
4.3.4 Presentation Model . . . 88
4.3.5 Learner Profile . . . 89
4.3.6 Scenario Generator . . . 90
4.4 Scenario Generation in GOALS by the learner . . . 91
4.5 Illustrative Example . . . 92
4.6 Technical Architecture . . . 94
4.6.1 Presentation Layer . . . 95
4.6.2 Business Layer . . . 98
4.6.3 Data Access Layer and Resource Layer . . . 99

5 Application Context : Project CLES 103
5.1 Context and objectives of the CLES Project . . . 104
5.2 Partners . . . 105
5.2.1 GERIP . . . 105
5.2.2 Laboratory EMC . . . 105
5.2.3 Laboratory LUTIN . . . 106
5.2.4 Laboratory LIRIS - SILEX Team . . . 106
5.2.5 Targeted Cognitive Functions . . . 107
5.3 Serious Game: Tom O'Connor . . . 109
5.4 Mini-Games . . . 109
5.4.1 Identify Intermixed Objects (Objets entérmélés à identifier) . . . 110
5.4.2 Memorize and Recall Objects (Mémoire et rappel d'objets) . . . 111
5.4.3 Point of View (Point de vue) . . . 112
5.4.4 Complete the Series (Séries logiques à compléter) . . . 113
5.5 CLES Modelling . . . 113
5.5.1 Main Concept Modelling . . . 114
5.5.2 Sub-Concepts Modelling . . . 116
5.6 Using GOALS for CLES . . . 119

6 Evaluations **121**
6.1 Introduction . . . 122
6.2 State-Of-The-Art . . . 122
6.3 Evaluation of Generator scenarios . . . 126
6.3.1 Evaluation Protocol . . . 127
6.3.2 Experiment and results . . . 131
6.4 Study of the impact of serious games on learners . . . 135
6.4.1 Evaluation Protocol . . . 135
6.4.2 Experiment and results . . . 136

7 Conclusions and Perspectives **141**
7.1 Perspectives . . . 143

Bibliography **145**

Chapter 1

Introduction

Contents

1.1 Motivation **2**
1.2 Research Context: Project CLES **3**
1.3 Research Objectives **6**
1.4 Research Questions **6**
1.5 Characteristics **8**
1.5.1 Generic Nature of the Scenario Generator 8
1.5.2 Continuous Knowledge Acquisition 9
1.6 Summary **9**

The lack of adaptation of the pedagogical scenarios in serious games motivates the work in this thesis. We start this chapter by showing the manifestation of this problem in the Project CLES that focuses on games for the evaluation and re-education of cognitive abilities. This follows by the presentation of the objectives of the research work. Next, we present the scientific research questions and the principle of the responses we give to answer these questions. We have also identified the characteristics that are necessary to make our approach generic and allow it to adapt by continuous acquisition of knowledge. The chapter ends with a plan for the rest of the thesis.

1.1 Motivation

The motivation of this research work can be explained through a scenario. Suppose a student named Jack. Jack wants to study a topic Division. This topic has several sub-topics. Furthermore, the topic Division can only be understood by the help of some other topic(s); these topics(s) can be either one or many. For example, let us assume that it's necessary to learn the topics Addition and Multiplication before learning Division. There are various teaching resources associated with each of the topics Division, Multiplication and Addition. These resources are necessary to teach the topics, such as documents explaining the concepts of topics, some examples, and exercises to test the competence of different topics like MathSnack[1], etc.

Now suppose that Jack is using a serious game to study Division. This game makes use of the different educational resources associated with Division and its related topics, in this case Addition and Division.

Lets consider that the serious game, in question, provides a One-Size-Fits-ALL solution i.e. the game does not provide any adaptation. Then we can outline some possible cases:

1. Jack is aware of the topics Addition and Multiplication, and the educational resources provided to Jack are exactly according to his level of understanding.
2. Jack does not know either the topic Addition and/or Multiplication.
3. Jack knows the topics Addition and Multiplication, but the educational resource is not according to his profile (competencies, preferences, abilities, etc.)

In the first case, the education provided by the serious game will satisfy Jack. In the second case, Jack needs to first understand the topics Addition and/or Multiplication, and afterwards, he can learn Division, and if this is not the case then he will have difficulties with the educational resources conveyed by the serious game. In the third case, Jack will either find the resource too facile or too difficult. In this last case, Jack could be frustrated by the difficulty of the game and consequently, lose interest in the game.

Any serious game designed in a non-adaptive manner faces these kinds of cases. If the serious-game designers assume that all the target users are similar to that of Jack in case 1, then this assumption is too optimistic, and it is hard to find in a real-world environment an example of this kind. In order to cater to different user needs, in cases 2 and 3 some adaptation of the educational resources is necessary. The adaptation, for the learners like Jack in case 2, can take the form of making sure that the learners has all the required competencies to progress in learning by playing the game. The adaptation, for the learners like Jack in case 3, could adapt the educational resource according to the competencies and preferences of the learner. This adaptation ensures that the learners are neither finding the game too difficult nor too easy to play. Similarly, the game should also be dynamically

[1] http://mathsnacks.com/

adapted according to the performance of the learner. This dynamic adaptation will keep the learner interested in the game and make sure that the learner achieves the maximum educational benefits. The need for adaptation increases when learners are in the **situation of a physical or cognitive disabilities**. Furthermore, if the game is on-line, where the number of learners is large, the need of adaptation increases, because its difficult to provide manual adaptation for each learner. Hence, the adaptation has to be provided in an automatic manner.

All this discourse can be summarized by saying that to make sure that a learner benefits the maximum, while playing a serious game, the pedagogical content should necessarily be personalized to the learner. This personalization should take into account the learner's profile and his pedagogical goals. Furthermore, the game should also take into account the pedagogical properties of a topic, i.e. whether the topic needs other topics to be understood, the topic can be decomposed into other sub-topics, etc.

1.2 Research Context: Project CLES

Computer games are providing entertainment to users almost since the inception of computers themselves. In addition of providing entertainment to their users, computer games, if used in moderation, can be beneficial in many other ways, as well. Researchers have observed higher level of hand-eye coordination, and visuo-motor skills in computer gamers [Enochsson 2004]. Computer games have also the capability to keep their players absorbed, engaged and motivated [Rieber 1996], all these traits can help increase the attention span of the player. Furthermore, games can help in the development of analytical and spatial skills, strategic skills and insight, learning and recollection capabilities, psycho-motor skills, visual selective attention [Mitchell 2004]; spatial modelling, design composition, and form creation [Coyne 2003, Radford 2000]. Mental rotation can be improved by playing games like TETRIS [De Lisia 2002]. Some other benefits includes: improved self-monitoring, problem recognition and problem solving, decision making, better short-term and long-term memory, and increased social skills such as collaboration, negotiation, and shared decision-making [ELSPA 2006, Mitchell 2004, Rieber 1996]. [Aldrich 2005, Tashiro 2009] present even some more advantages of games.

Researches have tried to make education appealing to learners by providing it through computer games. In the beginning, the idea of using games in lesson time was not hugely appealing to many, especially parents and teachers [Kirriemuir 2004, Klopfer 2009, Law 2008]. However, with the passage of time and the increasing interest of researchers in this domain, the idea of teaching via games has found its place and rightly so. As a result of these research, more and more sophisticated games for education came into existence. Researchers have also studied in considerable detail the impact on learning by games in comparison to computer-based teaching methods [Wong 2007, Papastergiou 2009]. In some contexts, the result from these studies has been positive [Papastergiou 2009]; especially in the

case of young children. Consequently, educational games found general acceptance with teachers and parents. In fact, a recent report of Entertainment Software Association (ESA) [2] showed that:

> *Parents also see several benefits of entertainment software, with 52 percent saying video games are a positive part of their child's life. Sixty-six percent of parents believe that game play provides mental stimulation or education, 61 percent believe games en-courage their family to spend to time together, and 59 percent believe that game play helps their children connect with their friends.*

In the literature, researchers use the term *Serious Game* to describe a game for education. The application domain of serious games is quite large, for example, health, medicine, training, military, business, advertising, etc, [Susi 2007]. In this thesis, we are the focused on serious games for the rehabilitation of **persons with cognitive disabilities**. In this context, researchers have developed many games for the evaluation and re-education of cognitive abilities. These games cover many different cognitive functions like treating visual-attention [Green 2003], memory [Ferguson 2007], visual-spatial [Enochsson 2004, Drivera 1991], attention [Castel 2005, Manly 2001], perception [Green 2010, Mody 1997], etc. Other systems use virtual reality to treat claustrophobic fear [Botella 2000], attention enhancement [Cho 2002], etc. These systems have the advantage of being more flexible and accessible. They can also store the traces of their users, which allow practitioners to monitor achievements and the progress of their patients [Sehaba 2005a]. However, most of these systems do not adapt to the characteristics and needs of each person. This adaptation is particularly relevant because different persons have different skills, abilities or preferences.

In the context of serious games and treating cognitive disabilities, we present the project CLES[3]. CLES acronym of Cognitive Linguistic Elements Stimulation is funded by the French industrial-ministry and supported by the business cluster Imaginove. The **objective of the project CLES is to develop an adaptive serious game for rehabilitation and cognitive stimulation** of persons with cognitive disabilities. This game, **available on-line**, aims at children and adolescents. Many research laboratories and enterprises have collaborated for this project. Among them is the team SILEX[4] of the LIRIS laboratory, the society GERIP[5], specializing in the development of edutainment for the rehabilitation of cognitive and linguistic functions, EMC[6] & LUTIN[7] laboratories, specializing in the study of cognitive mechanisms and the study of use of digital information technology respectively. This project is particularly interested in the following eight cognitive

[2] 2012 Essential Facts About the Computer and Video Game Industry http://www.theesa.com/facts/pdfs/ESA_EF_2012.pdf

[3] http://liris.cnrs.fr/cles/

[4] http://liris.cnrs.fr/silex

[5] http://www.gerip.com/

[6] http://recherche.univ-lyon2.fr/emc/

[7] http://www.lutin-userlab.fr/site/_pages/english/

functions [Hussaan 2011]: perception, attention, memory, oral language, written language, logical reasoning, the visio-spatial and transverse skills.

The serious game developed in this project is *Tom O'Connor and the sacred statue*. This is an adventure game. The protagonist of this game is a character named Tom, whose task is to find a sacred statue in a mansion. Based on different pedagogical sessions, this character finds himself in one of the several rooms, in the mansion. As shown in the figure 1.1, each room contains several objects (chair, desk, screen, etc.). Behind some of these objects, there are hidden challenges in the form of *mini-games*. The user has to interact with these objects in order to launch these mini-games. The player has to launch all the mini-games in the room to access other parts and advance in the game.



Figure 1.1: A room of Tom O'Connor's Mansion



Figure 1.2: Example of a mini-game related to Memory

Figure 1.2 shows the interface of a mini-game related to memory. As this figure shows, the game shows a series of images that the player must memorize. After a time period, the images disappear, the mini-game asks the player to select them among the several proposals. This game has several parameters: the number of im-

ages to be memorized and their complexity, the duration of display of these images, the number of proposals and the response time of the player. These parameters allow to adjust the level of difficulty of the game according to the abilities and needs of each player.

Thus, for each of the eight cognitive functions, there are a dozen games, and for each game, there are nine levels of difficulty. In the project CLES, the number of users exceeds 13 200, and since CLES is an on-line gaming environment; it will be difficult for the expert to interact individually with all of them. Therefore, it is of utmost necessity, to automatize the process of personalizing the player's path through the game and the activities according to the user's handicap, competencies and skills. Thereby, the role of the generator is to, on the one hand, select the mini-games and adjust their level of difficulty based on the player's profile, the traces of interaction, and the therapeutic goals of the session, on the other hand, put them in relation with the objects from different parts of the mansion.

1.3 Research Objectives

In the context of adapting educational content, in serious games, **the aim of this research is to propose models and processes to allow the generation of pedagogical scenarios that can be used in serious games**. These scenarios should be adapted to every learner. This adaptation should be done according to the learner's background knowledge, competencies, physical and cognitive skills, abilities, and pedagogical goals. Furthermore, the generation of scenarios will also take into account the traces of the learner's interaction.

We have not properly introduced some concepts, which we have used in the above paragraph like *pedagogical scenario*, *generated scenarios* and *interaction traces*. We have presented a detailed description of these three concepts in the following chapters, but let us describe them briefly. By *pedagogical scenarios*, we mean all the educational content (topics and resources) required to teach a topic to a learner. In order to generate such a pedagogical scenario, we need an automatic process that uses information about the educational content and the learner's pedagogical objectives to generate an adapted sequence of educational resources. This process can be referred to as *scenario generation*. The accuracy of the scenarios, generated by the generator, increase proportionally with the amount of information available to the generator. Therefore, we propose to use the *traces*, left behind as a result of the learner's interaction with the system/game, as knowledge sources in the scenario generation process.

1.4 Research Questions

In order to achieve the research objectives, there are some research questions that have to be addressed. Furthermore, these questions show the scientific founding of this work.

The first thing that we have taken into consideration, while designing an adaptive generator of learning scenario in serious game, is to identify the knowledge necessary for the required adaptation and, also models that represent and organize this knowledge. This identification process is necessary for the system for effectively providing the required adaptation. In this context of knowledge recognition, representation, and utilization, the first question is:

> **Question 1: What is the personalization knowledge to get for supporting the generation of adaptive pedagogical scenarios in a serious game environment? How to represent this knowledge?**

To answer this question we have identified and modelled different types of knowledge, including domain knowledge, serious game knowledge, and learner knowledge. The chapter 3 presents these models.

Next we look at the question of comprehending the use of this knowledge for the generation of adapted pedagogical scenarios in serious game. In this research's case, it is the use of the modelled knowledge to propose the learner with appropriate pedagogical scenarios. So the second question, is:

> **Question 2: What is the inference process for exploiting properly the personalization knowledge?**

To answer this question, we have proposed a model of a pedagogical scenario generator. This generator uses the knowledge, identified in the first question, to generate scenarios adapted to a user according to its objectives. These scenarios can be used in a serious game environment. The chapter 3 presents the pedagogical scenario generator.

We have mentioned that we generate the adaptive pedagogical scenarios according to the user. We have used the term "according to the learner", but we have not defined what does "according to the learner" means? And how to measure it? By these questions, we mean to evaluate the generated scenarios. This evaluation is necessary to validate that knowledge in use and that scenarios are appropriate according to the representation of the learner. Consequently, we need to respond to the following questions:

> **Question 3 : How to validate the functioning of the scenario generator (the knowledge models and strategies used to generate the pedagogical scenarios)? and How to study the impact of the generated scenarios on the actual learning of the learner?**

To answer this question, we have developed the GOALS platform (Generator of Adaptive Learning Scenarios) in which we have implemented the proposed models. We have used GOALS to validate the scenario generator's functioning and the impact of the generated scenarios on learning. To conduct these validations, we have

defined two appropriate methods. The first method is used to evaluate the system of scenario generation. The second is used to study the impact of the generated scenarios by the system on the learners' learning..

For the first one, we have proposed an evaluation protocol, based on a Comparative Evaluation Strategy, that we have conducted with an expert therapist in the context of the project CLES. The principle of this protocol is to compare the pedagogical scenarios produced by the proposed scenario generator with the scenarios created by an expert, for the same input. This protocol helps us in following the evaluation process and also identify if something is not right and identify the problem.

For the second one, we have conducted an experimentation with learners to study the impact of the generated scenario on them. The idea is to compare the performance of two groups of learners, one group uses the proposed scenario generator to learn while the other uses the traditional means of learning. We have conducted a pre-test to evaluate the actual competence of the two groups. Next, we allowed one of the groups to use the scenario generator, while the other uses traditional methods. Afterwards, we conducted a post-test to find out whether the group, using the scenario generator, had have a learning gain or not.

1.5 Characteristics

The scientific contributions according to the research questions, posed in the previous section, have a broader view, i.e. they aim to be utilizable with many pedagogical domains. Therefore, there are some characteristics that the proposed models should adhere to.

- Generic nature of the scenario generator
- Continuous acquisition of knowledge

In the next two sections, a detailed explanation of these two characteristics is presented.

1.5.1 Generic Nature of the Scenario Generator

Recall that the objective of this research is to propose models and processes to allow the generation of pedagogical scenarios that can be used in serious games. This research deals with two distinct domains: Serious Games and Pedagogical Domains. The research contributions can be applied in many potential fields like in the pedagogical domains of physics, maths, etc. In this thesis, we have tested this research in the field of rehabilitation and re-education of cognitive functions in the project CLES. However, our contributions intended to be *generic*. This means that the contributions can be utilized with many pedagogical domains and serious games. Ideally speaking, the proposed scenario generator can be used to generate scenarios of any pedagogical domain, that can be modelled or represented using

pedagogical properties. These scenarios can be used with any serious game that has the possibility of getting parametrized with pedagogical resources.

Therefore, it is necessary to organize the knowledge structure in a way that any pedagogical domain can be studied through a variety of serious games and a serious game can be used to teach a variety of pedagogical domains. To achieve this characteristic, we propose to organize the knowledge in three-layer architecture:

- Domain concepts
- Pedagogical resources
- Serious Game resources

The first layer contains the pedagogical domain's concepts. The second layer contains the educational resource/pedagogical resources related to the domain topics. The third layer contains the serious game resources that are in relation with the pedagogical resources.

1.5.2 Continuous Knowledge Acquisition

A system that remains static, while the world around it evolves, becomes out of sync with the world. This means that some of the suppositions made by the system about the world are no longer valid. Therefore, it is of ut-most important to the system that it evolves along with the world around it. This will help it to be relevant to its users. The second characteristic targets to keep knowledge models evolving by continuously acquiring knowledge about the learner through his interactions.

When the learners interact with the pedagogical scenarios, generated by a generator, their knowledge can evolve. If the approach does not record these changes in the learners' knowledge, then the knowledge about the learner will go out of sync with the learners' competencies. As a consequence, the scenarios proposed by the generator to the learners will become less and less utile over time. In order to avoid this situation, we propose to keep track of the learners' interactions with the serious game and use them as knowledge sources to update the system's representation of the learners. In addition, we propose to analyse these learners' interaction traces to find trends in the learners' behaviour patterns. We use these traces to follow the evolution in the learners' knowledge. We propose to analyse these traces to update the pedagogical domain model, if necessary.

1.6 Summary

The rest of the thesis is organized as follows: the next chapter, chapter 2, presents a literature review of the domain concerning this research. This chapter starts with defining formally the *adaptive systems* and the term personalizing/adapting of systems. Then the definition of the terms *pedagogical scenario* and *serious game* are presented.

Afterwards, we present a review of the scenario generation in serious games employed in education. Then another review of the pedagogical scenario generation in Technology Enhanced Learning Systems (TEL) is presented. We present an analysis of the both the reviews according to our objectives and characteristics. This analysis positions the research work, i.e. point out what lacks in the existing approaches and where exactly this work is contributing.

After outlining the areas where the contribution will be made, in the chapter 3, we present the contributions. These contributions are our answers to the research questions that have been identified in this first chapter. The chapter 3 starts by answering the first question, "how and what to represent for the adaptation?" The knowledge models are presented, and their organization is also discussed. Next, the second question is addressed i.e. "how to use the knowledge to generate the scenarios?" Here, the models, that are used for scenario generation and the different strategies/algorithms that have been employed in the generation process, are presented. Furthermore, the formal validation of these models is also discussed. The chapter ends with a summary.

The manifestation of all the proposed theoretical models in a fully functional platform is presented in chapter 4. Thus in this chapter, we present the platform GOALS (Generator of Adaptive Learning Scenarios), which we have implemented to test the contributions. GOALS lets a domain expert to create and organize the knowledge related to a pedagogical domain and serious game. Furthermore, the information about the learner can also be managed by GOALS. The expert can keep track of the performance of the learners. GOALS can also be used by the learners to interact with the personalized pedagogical scenarios. We also present the technical details of GOALS.

Chapter 5 highlights the application context of our work. Here, the proposed models and GOALS are put to practice in a real-world project, Project CLES. The chapter starts by presenting the description of this project, which includes the objectives of the project, the partners and their contribution in the project and the cause of our motivation to participate in the project. Afterwards, we present the serious game developed in the project CLES "Tom 0'Connor", along with the mini-games or pedagogical exercises that are associated with the project. We also presents the modelling of the CLES's knowledge via the proposed knowledge models. This chapter finishes with a simple example.

We present the response to the third question, which is regarding the evaluation of the contributions in the chapter 6. This chapter shows our efforts in evaluating the scenario generation process of the generator and the study of actual learning as a result of using the generator. The chapter starts by presenting a literature review of the evaluating process of similar approaches. This follows an analysis of the review and how we conduct the evaluations. Afterwards, we present the protocols, which we propose to follow the evaluation process. These protocols not only helps in guiding the evaluations but also to identify the problems in-case something goes wrong. Furthermore, we present two experiments, which we have conducted in the context of the project CLES. The first experiment validates the working of

the scenario generator and answers the first part of the third research question, i.e. how to evaluate the scenario generation process of the generator? The second experimentation studies the impact the of the generated scenarios on learning, and answers the second part of the research question i.e. to study whether the generated scenarios helps in the learning process. To end the chapter, we present a general analysis of the.

The chapter 7 presents some concluding remarks and discussions on our work. We discuss some limitations, and the future works that we plan to do.

Chapter 2

Related Work

Contents

2.1 Introduction 14
2.2 Definitions 15
2.2.1 Adaptive and Adaptable System 15
2.2.2 Pedagogical Scenario 15
2.2.3 Serious Game 17
2.3 Scenario Generation in Serious Games 18
2.3.1 Authoring Tools 20
2.3.2 Game Based Learning 23
2.3.3 Dynamic Difficulty Adjustment 24
2.3.4 Summary 26
2.4 Scenario Generation in AEHS 28
2.4.1 Course Sequencers 29
2.4.2 Course Generators 33
2.4.3 Summary 39

The objectives of our work require investigating different domains for solutions. These domains include adaptive systems employed in learning, more precisely, the approaches which proposed to generate pedagogical scenarios according to a learner, and serious games. Despite there being numerous kinds of serious game, we have studied only learning-based serious games. We start the chapter by defining different terms, which we use in the context of this study. Afterwards, we present an analysis of the different domains. Finally, we conclude this chapter by providing a summary of all our analysis.

2.1 Introduction

Recall, that our objective is to generate pedagogical scenarios that can be used in serious games. As it happens, the problem of the generation of the pedagogical scenarios and the problem of using pedagogy in serious games have been studied by different approaches. For the former, a branch of systems called *Course Generators* are present, and for the latter, serious games for learning or learning games. An analysis of these two domains is necessary for, on the one hand, to find out how much the solutions offered, by the current state-of-the-art, are closer to our objectives. And on the other hand, to find out that if the current solutions are insufficient, then where can a contribution be made to meet our objectives.

In the next section (section 2.2), we present the different terms that we use in this study. These include, pedagogical scenarios, adaptable, adaptivity, adaptive, personalization, and serious games. In the next two sections that follow, first, we review the existing state-of-the-art of scenario generation in serious learning games (section 2.4). Second; we review the adaptive learning approaches i.e. course generators in Adaptive Educational Hypermedia Systems (AEHS) (section 3.3). Each review is followed our analysis of the reviewed approaches. The review of serious games and course generators has been done keeping into account the criterion defined below:

Domain independent Architecture: Since we are targeting our approach to be generic in nature therefore, the reviewed approach should have a general architecture independent of the pedagogical domain and serious game.

Flexible scenario structure: Being generic also means that the reviewed approach should use flexible scenario structures in-order to cater various pedagogical domains.

Step-By-Step learner guiding: Our objective is to help the learner achieve his pedagogical goals, therefore, the approach, under consideration, should also have the ability to guide the learner in a step-by-step fashion towards his pedagogical goals

Adaptation of Pedagogical Resources: In line with our objectives the reviewed approach should be able to adapt the scenarios according to a learner. Therefore, not only appropriate resources are to be selected but also provisions have to be provided to adapt the pedagogical resource, wherever it is possible, according to the learner.

Continuous Knowledge Acquisition: In order to achieve this characteristics, the reviewed approach should make use of interaction traces for the updating of learner profile and adaptation of the learning scenario.

Serious Game oriented: Since we also target to use the pedagogical scenarios with serious games, therefore, reviewed approach should also take into account the serious game specificities to generate the scenarios.

2.2 Definitions

In this section, first, we present that defines the adaptive nature of a system, then we analyse the different uses of the term pedagogical scenario and give a definition that we use. In the end, we define the notion of serious games.

2.2.1 Adaptive and Adaptable System

In this thesis, we have frequently used the terms like adaptivity, adaptable, personalization and adaptation. Different researchers have used these terms differently to describe different aspects of their respective approaches. Here, we attempt to find a proper definition of these terms and explicitly state what they refer to in this manuscript.

According to [Oppermann 1994], a system is said to be **adaptable**, "If a system provides the user with the tools that make it possible to change the systems characteristics." Furthermore, a system is said to be **adaptive**, "If a system is able to change its own characteristics automatically according to the user's need." Moreover, adaptivity, "is the form of an adaptive system based on the assumption that the system is able to adapt itself to the wishes and tasks of the user by an evaluation of user behaviour."

Another term that is often used is **Personalization**, is defined by [Germanakos 2006] as:

> Nevertheless, most of the definitions that have been given to **personalization** are converging to the objective that is expressed on the basis of delivering to a group of individuals' relevant information that is retrieved, transformed, and / or deduced from information sources in the format and layout as well as specified time intervals. More technically, it includes the modelling of Web objects (products, and pages) and subjects (users), their categorization, locating possible similarities between them and determining the required set of actions for **personalization**. On the other hand, many argue that for the actual meaning of personalization, not only personalized information needs but also emotional or mental needs, caused by external influences, should be taken into account.

In the literature, adaptation is much more general than personalization. Personalization is a specific case of adaptation i.e. when the system tries to fit the needs of a person (collective or individual one). However, some authors [Baldoni 2005] do not make this distinction in their research. The system in this research is adaptive in nature, thus providing automatic adaptation.

2.2.2 Pedagogical Scenario

One other term, which is at the centre of our research, is **pedagogical scenario**. Different researchers have tried to give their own views about the notion of the

pedagogical scenario. We give the the different point of views used, by different authors, to define what constitutes a **pedagogical scenario**, and then we give our definition of a pedagogical scenario.

- According to [Peter 2005], "One way to define the activities that will take place within a unit of study is to describe them in a **pedagogical scenario**. Such a scenario defines the activities which must be done by the learners and the tutors, the sequencing of these activities as well as the learning objects and tools that should be provided to the different actors. For instance, the emerging standard IMS-LD uses a theatrical metaphor where the activities take place in different acts that define the sequencing."

- [Schneider 2003] defines a pedagogical scenarios as "a sequence of phases within which students have tasks to do and specific roles to play. In other terms we advocate creative but flexible and open **story-boarding**."

- A pedagogical scenario presents a learning activity *turnkey*, initiated by a teacher to guide the learning of students (before, during and after the activity sheet with self-assessment and evaluation, implementation situations, educational resources, etc.). A pedagogical scenario presents an approach for achieving educational goals and skills related to general or specific to one or more disciplines under the terms and specifications of the curriculum. The scenario gives rise to a project, a particular learning activity, the realization uses the resources of the Internet and possibly also in print, audiovisual or multimedia. A standard form of pedagogical scenarios is a check-list that allows the sharing of resources between projects and teachers.[1] [Bibeau 2004]

- For [Guéraud 2006], the concept of a pedagogical scenario "a key element: scenarios are created by trainers (instructors) to propose a set of activities and goals on Interactive Learning Objects, such as simulations, micro-worlds; scenarios are further used to assist trainers in their task of monitoring the class activity."

- [Pernin 2006] defines a learning scenario as an object "which represents the description, carried out a priori or a posteriori, of the playing out of a learning situation or a unit of learning aimed at the acquisition of a precise body of knowledge through the specification of roles, and activities as well as knowledge handling resources tools, services and results associated with the implementation of the activities."

- [Tetchueng 2008] In their framework, the goal of scenarios is to describe the learning and tutoring activities to acquire some knowledge domain (for instance physics) and know-how to solve a particular problem. They claim that a scenario is defined from the following dimensions: the learning domain

[1] http://www.epi.asso.fr/revue/articles/a0409a.htm

(course topic), the learner, the learner know-how and knowledge levels, the tutor/teacher, the resources (documents, communication tools, technical tools, etc.), the pedagogical and /or didactic model, the learning *procedures* according to a particular school/institution/ university, classroom type, face to face or at distance,

- According to [Emin 2008], a *learning scenario* describes the organization and schedule of learning situations implying various actors (student, teacher, tutor, designer, etc.).

We can conclude that:

> A pedagogical scenario is a suite of ordered activities, which have to be performed by different actors (learners, teachers, etc), in-order to achieve a pedagogical goal.

Generally, these definitions include the role of multiple actors in a pedagogical scenario. For our research, we are only interested in the interaction with one type of actor i.e. the learner. Furthermore, the activities have to be proposed to the learner according to the learner's profile. And the scenario should also take into account the specificities of the computer-based learning environment which for our case is serious games. Therefore, we use a definition similar to that used by [Ullrich 2009a, Tetchueng 2008], and define a pedagogical scenario as:

> *a suite of pedagogical activities generated by the systems for a learner keeping into account the learner's profile to achieve a pedagogical goal in a computer based learning environment (in our case, serious games).*

In the next section we define the notion of Serious Game.

2.2.3 Serious Game

Serious game has been defined differently by different authors. For instance, according to [Bergeron 2006], "An interactive computer application, with or without a significant hardware component, that has a challenging goal, is fun to play, incorporates some concept of scoring, and imparts to the user a skill, knowledge, or attitude that can be applied in the real world."

Whereas, [Michael 2005] defines a serious game as, "A serious game is a game in which education (in its various forms) is the primary goal rather than entertainment". They further add that, "Thus serious games are games that use artistic medium of games to deliver a message, teach a lesson, or provide an experience."

[Mikael 2009] define serious games as, "games that engage the user, and contribute to the achievement of a defined purpose other than pure entertainment (whether or not the user is consciously aware of it)."

The position on serious games taken by [Zyda 2005] is as follows, "Serious game: a mental contest, played with a computer in accordance with specific rules, that

uses entertainment to further government or corporate training, education, health, public policy, and strategic communication objectives."

One thing that is clear from all the different definitions is that serious games are games where the main objective is education rather than entertainment. It is Zyda that argues serious games have more than just software, story and art; it is the addition of **pedagogy** that makes a game serious. Zyda uses the figure 2.1 to demonstrate the elements of a serious game.

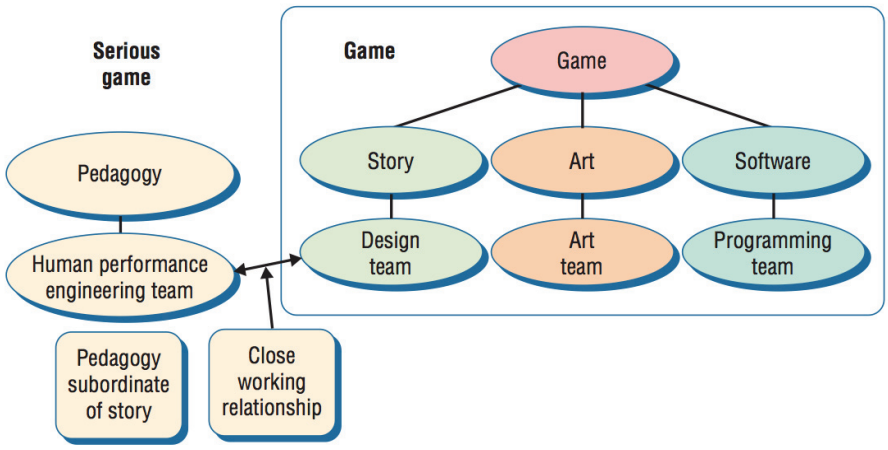


Figure 2.1: Zyda's definition of a serious game

The figure 2.1 shows the basic elements that are necessary to develop a video-game. Games are mostly story-driven, i.e. a gaming scenario can be defined for the purpose of keeping a player motivated and immersed in the game. This scenario gets defined by a design team. Not only, the scenario has to be defined but also the gaming world has to be designed. This world contains different forms of elements according to the gaming scenario, and then there is a programming team who makes the game playable by implementing the gaming scenarios. In order to make the game serious, the pedagogy has to be engineered. This gets done keeping into mind the pedagogical objectives that are to be achieved by the game. Though a close working relationship is desirable between the game scenario design team and the pedagogy engineering team, however, a carefully planned serious game can have the gaming elements usable by many different pedagogies.

2.3 Scenario Generation in Serious Games

Since we are primarily focused on the use of serious games to deliver educational content, therefore, we review this domain to search for solutions. Though serious games have been used in many domains [Susi 2007] for example, military games, government games, educational games, corporate games, healthcare games, and political, religious and art games, etc, [Michael, David R. And Chen 2005]. A classification of serious games is also shown in figure 2.2 [George 2010].

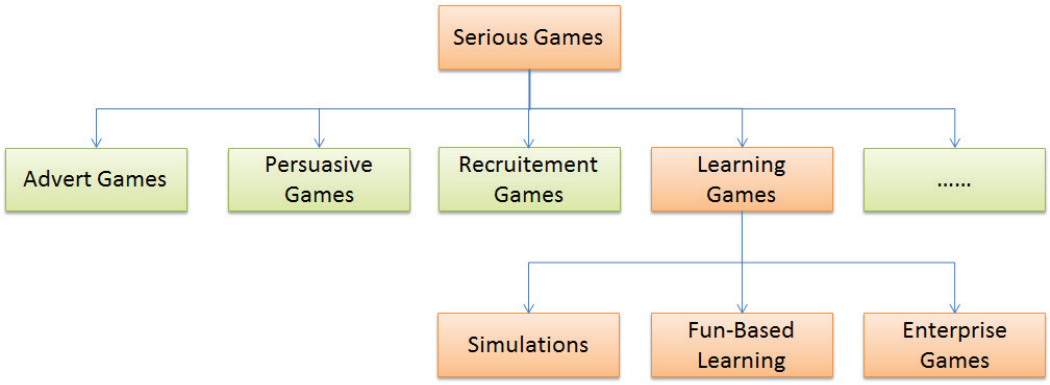


Figure 2.2: Classification of serious games [George 2010]

We are only considering those approaches that provide any learning, they can be called *Learning Games*. Learning games comprises of many kinds of approaches like: simulations, fun-based learning and enterprise level learning. Learning games try to transfer knowledge through the gaming experience [Fu 2009]. Simulation is an "acting out or mimicking an actual or probable real life condition, event, or situation to find a cause of a past occurrence (such as an accident), or to forecast future effects (outcomes) of assumed circumstances or factors"[2]. Fun-based learning tries to teach traditional concepts like maths (Math Blaster[3]), language [Amoia 2012], physics [Vanlehn 2007], etc, using fun elements of traditional digital games. There are also games aimed at providing training in an Enterprise. These games try to prepare different personnel in an enterprise with different aspects that can help the enterprise in improving its performance. Some examples of these games are: The Enterprise Game[4], Renault Academy[5], Gaining Leadership[6], etc.

This review encompasses different categories of approaches concerning Learning Games.

We have dedicated a section for each of these categories. Each section starts with an introduction to the category. This follows a review of multiple approaches. For each of the reviewed approaches, the review starts with an introduction of that approach followed by our analysis. Each section ends with a summary of all the approaches presented in each section. Finally, at the end of each section 2.3 we present an overall analysis of all the reviewed serious game approaches.

[2]http://www.businessdictionary.com/definition/simulation.html
[3]http://www.mathblaster.com/
[4]http://www.enterprisethegame.com/
[5]http://www.daesign.com/en/realisations/renault-academy.html
[6]http://www.ranj.com/content/werk/the-gaining-leadership-program

2.3.1 Authoring Tools

The main objective of these kinds of tools is to provide the user with the possibility to create a serious game, without delving into the technical details of serious game design. These tools could have the possibility to design a system, which can be used with multiple games and pedagogical domains, hence meeting our criterion.

Recently, some of the researchers have tried to use Intelligent Tutoring Systems(ITS) with games. This would not only, bring together the **learning** advantages of an ITS with the **liking** of a game. However, also, theoretically, fill the gap between learning and liking, often faced by ITS developers. An investigation into this topic has been conducted by [McNamara 2010]. Another conceptual model for game-based ITS is presented in [Mills 2007], figure 2.3 shows this model. This model allows to exploit the adaptation features of an ITS in the game.

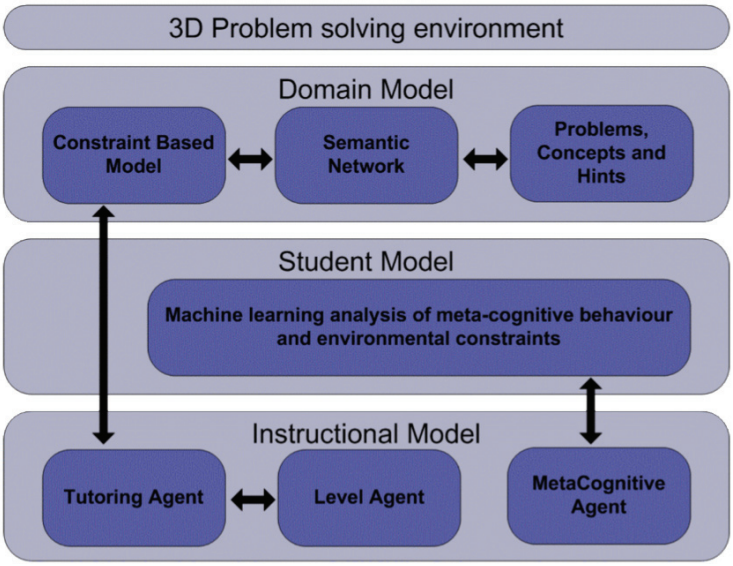


Figure 2.3: Mills' conceptual design for a game based ITS [Mills 2007]

In the figure 2.3 the domain concepts are linked with hints about modifying or using these concepts via a semantic network. The student model gets learned via machined learning techniques, and it contains the student's learning goals and his preferences. The agent based instructional model helps the student in the game, in his/her learning process.

However, according to the author, the modelling of domain knowledge should be tightly integrated with the game elements. This implies that a single game cannot be re-used with other pedagogical domains, which is a disadvantage of these kinds of models, from our point-of-view.

In this regard, [Moreno-Ger 2007b] presents a document-based approach to create games . The initiative is <**e-Game**>. The idea is to describe the gaming/pedagogical scenario in an XML-based language (<e-game> document), this

document gets entered in the <e-game> engine, which will render this scenario in a visual game by using the arts assets. The approach can be seen in the figure 2.4. They same authors proposed a similar language called **<e-adventure>** [Moreno-Ger 2007a, Moreno-Ger 2008a, Burgos 2008] and used it as an adaptive Units of Learning (UoLs) in LMSs. It allows the dynamic adaptation of the pedagogical scenarios. However, they define the adaptation by pre-defined paths. The figure 2.5 shows these paths, each path provides different gaming experience to the player. The path gets chosen based on a pre-test given to the player.

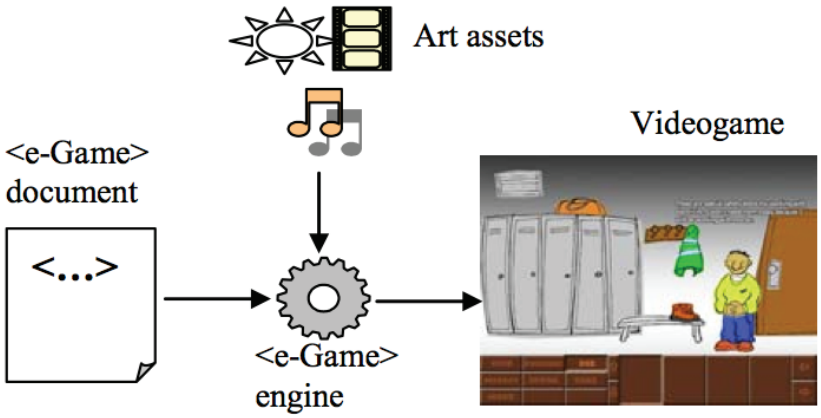


Figure 2.4: <e-Game> engine [Moreno-Ger 2007b]

These kinds of approaches divide the players/learners into large groups. Hence, the personalization aims at different groups rather than individuals. We know that each learner is different and should be dealt according to his competencies, preferences and performances, as [Carro 2006] proposes for adaptive educational games. Moreover, the scenarios, in <e-adventure> language, get defined manually by the expert, in the form of a tree and as the scenarios become complex in nature, the definition of the tree becomes a complex task.

[Carro 2006] presents a methodology for describing adaptive educational game environments and a model that supports the environment design process. These environments combine the advantages of educational games with those derived from the adaptation. The role of the system is to permit the Teacher to create an environment of exercises, for a user. These exercises can contain activities or games. These activities or games get selected on the fly according to the user.

However, once an activity gets selected it cannot be adapted to the user. They also did not discuss the modelling of the didactic domain. They also did not discuss the practical implementation of their approach.

[Bieliková 2008] presents a system S.M.I.L.E Smart Multi-purpose Interactive Learning Environment. S.M.I.L.E. gets used for generating three-dimensional interactive multimedia educational games. They also make provisions for handicapped persons in their systems. The teacher can create a game without any programming

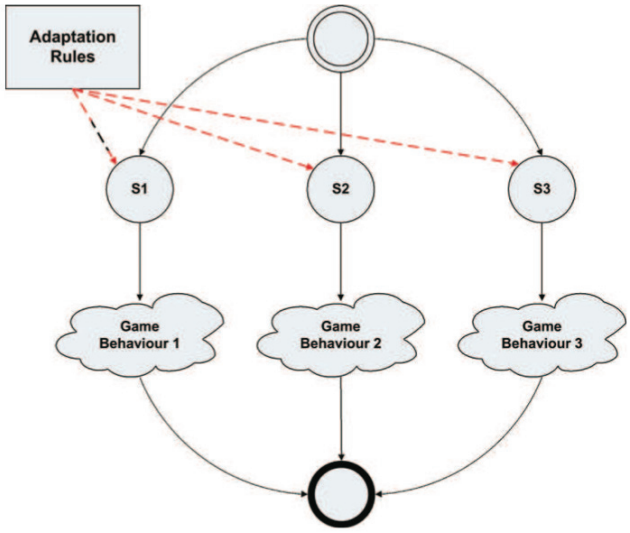


Figure 2.5: The dynamic adaptation in the <e-adventure> platform [Moreno-Ger 2007a]

knowledge; however, it requires an enormous amount of explicit modelling effort on teacher's part to model the learning and gaming objects.

The notion of a pedagogical scenario is not clearly present in this approach. The games are adaptable but not adaptive. The users can choose the preferences of display, but this gets done manually and not automatically. The relations get defined at the pedagogical resource level rather than the domain concept level which makes it difficult to add new resources.

These and other similar approaches for authoring serious games ([Dung 2010]) are present, but they fail to satisfy the criterion for our work.

To address these issues [Kickmeier-Rust 2006] has proposed the **ELEKTRA** project. The project aims to develop a methodology to create games where the gaming content, as well as, the pedagogical content could be easily adapted, dynamically, according to the performance, competencies and preferences of the learner. Though, a lot of research had been done and published in the context of ELEKTRA, including some games ([Steiner 2009], etc.), no concrete methodology has yet to come forward.

The authoring tools for serious games are there to allow teachers/domain experts to create games without any experience in game design or programming languages. However, the games they produce are tightly-coupled with the pedagogy, this means that the games cannot be reused with other pedagogical domains. The teachers have to define manually the learning paths or the pedagogical scenarios. This manual definition of pedagogical scenarios is not feasible in the case of hundreds and thousands of learners. Sometimes, no feedback gets modelled by these authoring tools. The feedback is necessary to update or modify the pedagogical scenarios according to

the performance of the learner.

Next, We look at some of the games, which provide education, and review their process for delivering education.

2.3.2 Game Based Learning

In this section, we will review some of the systems, which mix pedagogy with games. Game based learning (GBL) is a branch of serious games that deals with applications that have pre-defined learning outcomes. Generally, they are designed in order to balance the subject matter with the gameplay and the ability of the player to retain and apply said subject matter to the real world.

[Bikovska 2007] presents an approach for scenario development methodology for planning and management of business simulation games. They propose to develop the scenario as trees. Furthermore, the scenarios are implemented in the form of a game. They do not separate pedagogical aspects from the gaming aspects. Moreover, the scenarios are static in nature.

[Carron 2007] proposes a learning environment based on a graphical representation of a course: a pedagogical dungeon. The author of the course defines a dungeon which is tightly coupled with the pedagogical scenario. The traces of the user are used by the teacher to monitor user performances, give the player hints if necessary and to initiate collaborative activities with other students. Pedagogy is tightly coupled with the game, which makes the re-use of this system difficult.

[Chang 2008] presents a serious game to teach C programming language. Bomberman game supports learning concept of C programming language and teachers can build a meaningful game environment to specify the sequence of topics for students to learn.

The tight coupling between the pedagogical scenario and the gaming interface deprives the above mentioned and similar ([Brown 2009], [Tashiro 2009]) approaches from re-usability.

[Hodhod 2009] developed a serious game to teach ethics. They proposed a model to implement an adaptive educational interactive narrative system (AEINS). AEINS is an inquiry-based edu-game to support teaching ethics. Their proposed architecture can be seen in the figure 2.6.

They present a concept of teaching moments. They are the moments where learning can take place. This paper addresses the issues of modelling of the didactic domain as well as integrating it with a story generator. However, the teaching moments have to be defined by the teacher in advance and are not generated automatically.

[Torrente 2009] proposes an approach for making a serious game. They proposed a HCT game. In their paper, they exemplify through a case study how the <e-Adventure> educational game platform addresses these issues, describing the development of a low-cost, adaptive and assessable game-like simulation in the field of Medicine education.

[Lo 2008] presents a design of a digital game-based e-learning system aiming at

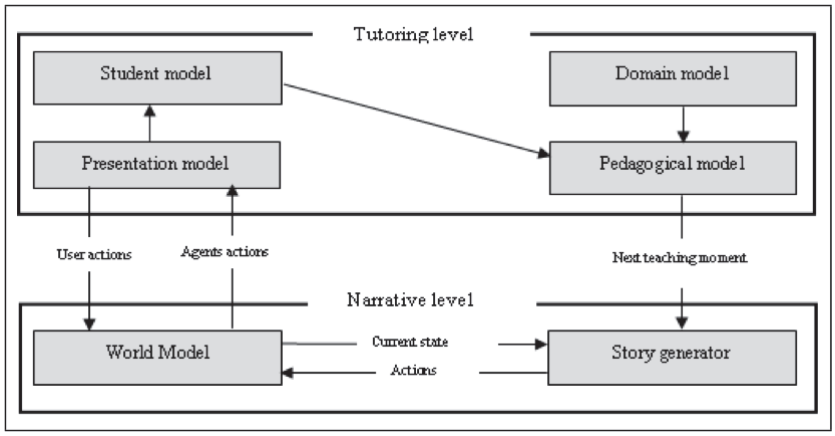


Figure 2.6: Architecture for AEINS [Hodhod 2009]

4-6 grades elementary students for ocean ecology learning. The main scenario of the game is centred on the life of a sea turtle. The scenario presented is non-linear. Though, the scenario, presented in their paper, is non-linear, and it is pre-defined and thus non adaptive.

Game based learning systems have a well-defined pedagogical objective. The above mentioned approaches serve the purposes in their own regard. However, these systems (and other similar systems) do not satisfy the criterion outlined for our approach. There are some games that have an explicit notion of a pedagogical scenario; however, most of the times, these scenarios are to be laid out manually by the designer. The manual definition of scenarios will make it difficult to personalize the game for a large number of learners. There are also some approaches, which allow some dynamic adaptation in a pedagogical scenario. They work by defining a tree of possible outcomes and the learner follows one path through the tree according to his performances in the game. However, in this case the learners get divided into large groups, and the personalization or adaptation gets provided to a group rather than an individual.

One other limitation of these approaches, from our point of view, is that the pedagogy gets embedded tightly in the gaming scenarios. This means that the games focus on one pedagogical domain. Consequently, these games cannot be easily used with other pedagogical domains; hence, no re-use is easily possible.

Since, we are also aiming towards the dynamic adaptation of scenarios; therefore, we also looked at some of the techniques employed in general games.

2.3.3 Dynamic Difficulty Adjustment

Many research approaches use the concept of Dynamic Difficulty Adjustment (DDA), for example, [Togelius 2007, Jennings-Teats 2010, Hunicke 2004, Yang 2007]. The idea behind DDA is to modify or adapt the levels of a game accord-

ing to the performance of the user [Jennings-Teats 2010]. Such that, the user does not feel bored (too easy) or frustrated (too difficult) while playing the game. This idea interests us as we are also trying to modify or adapt the pedagogical scenario of the serious game according to the performance of the user. Hence, this review will give us an idea about how this gets done in games and whether these games adapt only the game-level elements or both the pedagogical and game elements.

[Togelius 2007] uses evolutionary algorithms to adapt the racing tracks of a car racing game according to a player. Their approach targets commercial car racing games. They model the player in order to provide adaptation. They use supervised learning to associate the state of the car with the actions the human take given that car state.

Similarly, [Jennings-Teats 2010, Hunicke 2004, Yang 2007] all try to use DDA for different purposes, and all of them have the same limitation, from our point of view, they do not discuss pedagogy. The techniques provided by these approaches focus on the technical aspects of their respective systems and provide provisions for the adaptation of the gaming. Consequently, it is not easy to use them in a pedagogical context. However, some other researchers have proposed other techniques to be used in games for adaptation purposes.

[Bakkes 2008] discusses an alternative to existing approaches for the adaptive game AI for adapting rapidly and reliably to game circumstances. Their approach can be classified in the area of case-based reasoning. In their approach, domain knowledge is necessary to adapt a game. The circumstances get gathered automatically by the game AI, and get exploited immediately to evoke effective behaviour in a controlled manner, in on-line play.

[Ram 2007] proposes a case-Based reasoning approach for adaptive strategy of a game AI. The main idea behind their approach is the utilization of Expert's trace as a base to adapt to the player's trace i.e. they let the expert demonstrate how to resolve a problem. While the expert is solving the problem, they store the traces of his/her interaction as case in their case base. Next the expert annotates these traces. The purpose of this annotation is to describe which action gets performed to achieve which goal. In the next step, the CBR techniques re-use the expert's traces to adapt to the current goal of the player.

These approaches are useful in some contexts; however, in a pedagogical context, the way an expert solves a problem might be different from how a novice solves a problem. Hence, the expert's trace might not be beneficial for our objectives.

Though, there are many approaches, which tackle the problem of DDA in different gaming contexts. However, they do not discuss the adaptation of pedagogical aspects of the game. It would be interesting to link the adaptation in a pedagogical scenario with that of the gaming aspects. In this way, we can modify the gaming elements or levels according to the generated pedagogical scenarios. However, this is not part of the scope of this thesis.

2.3.4 Summary

There are many approaches, which use serious games for educational purposes. Here we reviewed three kinds of serious gaming approaches, namely: Authoring Tools, Game Based Learning systems, and systems based on dynamic difficulty adjustment. For each category of approaches, we presented different approaches that are more relevant to our objectives. Table 2.1 shows the comparison of the reviewed systems according to our criterion (see page 14).

Recall the criterion we have defined for our review, the first is that the approach should be usable with a variety of pedagogical domains. To fulfil this criterion, the approach needs an explicit modelling of the pedagogical domain, so that different pedagogical scenarios for different domains could be automatically generated. Though the authoring tools have the provision to represent pedagogical elements, they do not perform any pedagogical modelling. This means that any automatic intelligent reasoning, is not easy to do for the generation of pedagogical scenarios. GBL approaches have the same problems. Both kinds of approaches demand tight integration of the pedagogical and gaming aspects, which means that a game cannot be used with multiple pedagogical domains.

The authoring tools provide the tools to model the pedagogical scenario, though, most of them do not consider the pedagogical or didactic properties of the pedagogical elements. They define their pedagogical scenarios in relation to the gaming scenarios. GBL approaches have the pedagogical scenarios embedded in them; consequently, they do not provide with the tools, to model the scenarios.

The third criterion is to help the learner in a step-by-step manner towards his pedagogical objectives. Mostly, these approaches have the capacity to guide their learners. However, this guidance is not adaptive in nature, i.e these approaches can define a step-by-step path, but this path gets used by all the learners. Some approaches offer a bit more, and they define paths for different groups of learners. However, no approach provides individual paths for individual learners.

Some of these approaches provide with the option of manually adapting the pedagogical or gaming elements in the games. Most of the time there is no adaptation of the pedagogical scenario, and even some of the approaches, which try to provide adaptation they do so for a group of learners and not for an individual learner.

Not every approach uses the learners' interaction traces as knowledge sources to adapt the pedagogical scenarios. However, there are some approaches that do so.

Furthermore, last but not the least, all the approaches qualify the final criteria of being usable in serious games as they are all serious games.

Since, none of the existing serious game based approaches satisfy all of our criteria; therefore, we searched for a solution elsewhere. And because the work deals with the automatic generation of pedagogical scenarios, we extended the sphere of our research to include systems designed to deliver adaptive education to learners. Since the number of such approaches are numerous, we conducted a review of them, the next section presents this review.

	Domain Independence	Resource Adaptation	Traces for updating	Step-By-Step Guidance	Serious Game	Pedagogical Scenario
<e-Game>/<e-adventure> [Moreno-Ger 2007a, Moreno-Ger 2008a]/ Blood-Game HCT [Torrente 2009]	yes	no	no	yes	yes	pre-defined
[Carro 2006]	Not Mentioned	no	yes	yes	yes	yes - static
[Bielikov´a 2008]	yes	no	no	yes	yes	no
[Bikovska 2007]	could be	no	no	yes	yes	yes - static
[Carron 2007]	no	no	yes	yes	yes	yes - static
[Chang 2008] [Brown 2009] [Tashiro 2009]	no	no	yes	yes	yes	yes - static
[Hodhod 2009]	yes	no	yes	yes	yes	yes
[Lo 2008]	Not Mentioned	no	no	yes	yes	no
[Togelius 2007] [Jennings-Teats 2010, Hunicke 2004, Yang 2007]	no	no	yes	no	yes	no
[Ram 2007]	yes	no	yes	yes	yes	no

Table 2.1: Comparative table of different approaches

2.4 Scenario Generation in AEHS

Traditional Technology Enhanced Learning (TEL) systems provide the learner few ways of adaptive learning. Consequently, the learning gains from these systems are not great [Mulwa 2010]. In the context of personalization, The problem of presenting the learner with personalized learning scenarios is part of the systems known as Adaptive Educational Hypermedia Systems (AEHS). AEHS proposes to tailor the information delivered to the learners according to their needs as compared to the "one-size-fits-all" technique of the traditional course [Brusilovsky 2001a]. The functionality of an AEHS gets defined by [Brusilovsky 2001a] as:

> By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user.

Formally, AEHS are defined by [Henze 2004] as a quadruple:

$$(DOCS, UM, OBS, AC) \tag{2.1}$$

where,

DOCS : **Doc**ument **S**pace belonging to the hypermedia system as well as associated information. The associated information may include annotations, domain graphs that model the document structure (e.g. a part-of structure between documents, comparable to a chapter - section - subsection - hierarchy), or knowledge graphs that describe the knowledge contained in the document collections (e.g. domain ontologies).

UM : **U**ser **M**odel stores, describes and infers information, knowledge, preferences about an individual user

OBS : **Obs**ervations about user interactions with the AEHS. These interactions are recorded in the user model.

AC : **A**daptation **C**omponent: rules for the adaptation functionality.

The adaptation functionality varies from approach to approach. From a recommendation of a particular learning resource to a particular learner, or adapting the learning strategy of a learner, etc. A fairly recent and comprehensive review of the existing AEHS and the different types of adaptation they provide is done in [Knutova 2009]. The approaches under the umbrella of AEHS that are focused on the selection or generation of personalized scenarios are called *Course Generator*.

Course Generation has been considered by researchers for long. The idea behind a course generator is to develop a system that, for every particular learner, produces a course plan that helps the learner in achieving his pedagogical goals. There are two main approaches to this generation process:

1. Course Sequencing, and
2. Course Generation

In the following sections, we present a review of these two approaches. We dedicate a section to each of these approaches. Each section starts with an introduction to that approach followed by our analysis. Each section ends with a summary. Finally, at the end of the section 2.4 we present an overall analysis of all the reviewed approaches.

2.4.1 Course Sequencers

According to [Brusilovsky 2003a],

> **course sequencing** is a well-established technology in the field of intelligent tutoring systems (ITSs). The idea of course sequencing is to generate an individualized course for each student by dynamically selecting the most optimal teaching operation (presentation, example, question, or problem) at any moment.

The adaptation provided by these approaches is highly dynamic in nature. Different approaches have applied the idea of adaptive sequencing of learning objects in different contexts. At the early stages of its development, approaches aim at sequencing only one pedagogical operations [Barr 1976][McArthur 1988]. **Task Sequencing** was the approach used by some other systems [Eliot 1997, Rios 1999, Brusilovsky 1993]. The idea behind task sequencing is to arrange the order of questions or exercises. Some other approaches [Capell 1993, Brusilovsky 1994] store the courses in large chunks, where each chunk represents a complete lesson with information and exercises. Then these chunks get presented in an orderly fashion to the learner, according to the learner's competencies and requirements. Other advanced systems [Khuwaja 1996, Brusilovsky 1992, Vassileva 1992] were able to sequence complicated courses, which contained examples, presentations, tests.

[Van Marcke 1990, Van Marcke 1992, Van Marcke 1998] propose a Generic Tutoring Environment (GTE). The instructional knowledge represented in GTE composed into instructional tasks, instructional methods, and instructional objects. Tasks represent activities to be accomplished during the teaching process. The tasks get performed by methods that decompose tasks into subtasks down to a level of primitives. They call, the tree that results from the repeated decomposition of tasks into sub-tasks by methods, a task structure.

Though, in GTE, the question of presenting the learner with appropriate resources gets addressed based on the difficulty of the exercise and the performance of the learner; however, it does not, in its present state, allow the selection to be made of required competencies. This information tells whether a learner has all the required competencies to access this resource. Moreover, GTE also does not entertain the idea of a notion of *Pedagogical Scenario*; thus the learner cannot tell the system to generate specific scenarios.

[Vassileva 1995, Vassileva 1998b, Vassileva 1997, Vassileva 1998a] present a Dynamic Courseware Generator (DCG). The decision regarding pedagogical elements like, resources, domain concepts, gets made based on rules. These include, how to present a concept and how to test a learner. They made a clear distinction in the different domain knowledge. Concepts of a domain are in a tree-like structure, where the nodes represent the concepts, and links are the pedagogical relations between those concepts for e.g. pre-requisite, etc. All the pedagogical resources get represented by HTML pages. Each resource is in relation with domain concepts. Furthermore, each resource has a type i.e. it has a role like introduction, exercise, example, and theorem.

The architecture of this approach can be seen in the figure 2.7. The course gets planned in two steps: *content planning* and *presentation planning*, as proposed in an earlier work by [Wasson 1990]. The learner goals and his present knowledge get utilized to create a path connecting the concepts known by the learner and the goal-concepts. The content planning does this operation. Presentation planning makes use of various plans to present the learner with appropriate pedagogical resources. Based on these plans the AI-planner of DCG decides what to present next to the learner.

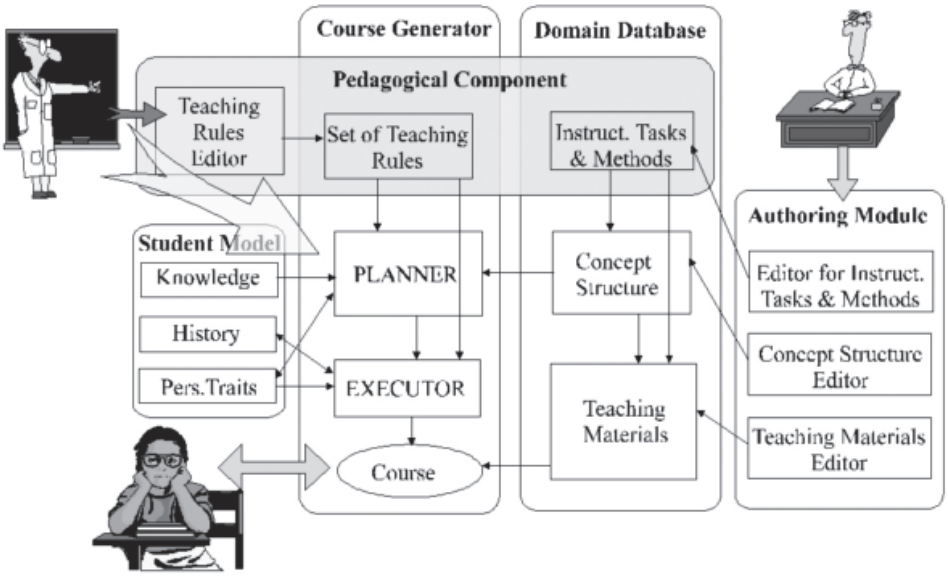


Figure 2.7: Architecture of DCG [Vassileva 1998a]

These plans represent the structure of the *Pedagogical Scenarios*. There are four types of plans *hierarchical*, *advanced organizer*, *basic concept*, and *discovery*. Each plan defines a sequence of tasks to be accomplished by the learner for e.g. hierarchical method uses the sequence *introduce, explain, give example, give exercises, and give a test*. Where the structure of *give exercises* is shown in the figure 2.8.

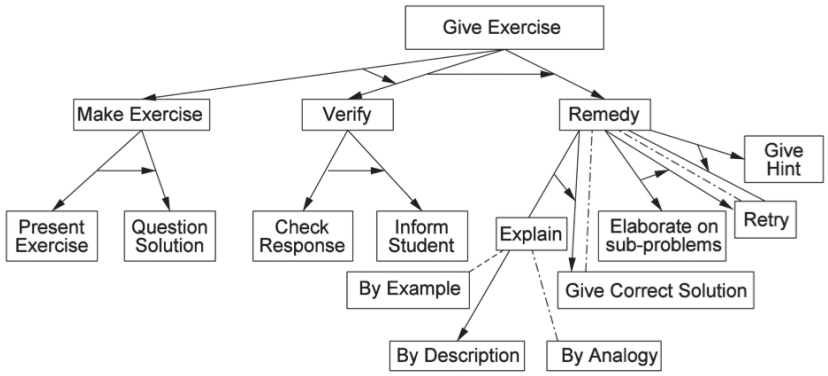


Figure 2.8: The structure of the task *Give Exercise*

DCG in general seems to attain almost all of our objectives; however, there are some limitations, from our point of view. In DCG, no provision gets provided to adapt further a pedagogical resource once it gets selected. This is possible where the resources can be parametrized for e.g. mini-games. This parametrization allows adaptation at a much higher level of granularity. The scenarios generated to take into account only the current concept; this inhibits the planner from presenting the learner with the resources related to other related concepts. For example, a learner may be interested in learning about a concept as well as its pre-requisite concepts at the same time. Furthermore, since, DCG targets to be usable with traditional educational environments, it does not consider its use with serious games.

[De Bra 2006] proposes an approach for creating and delivering web-based courses with adaptive navigation support. This approach makes use of the open source adaptive hypermedia platform AHA! [Bra 1998, Bra 2001]. AHA!'s adaptation engine filters content pages and link structures according to the user model.

Adaptive content is provided via conditional fragments, and the links are adaptively annotated according to the values in the user model. To deliver the adaptive course, a domain concept structure gets maintained in the form of a graph, where, the nodes are concepts and the links represent the *pre-requisite* relation between them (though the possibility of adding other kinds of relationships has also been descried [De Bra 2002]). Generally, every concept is related to a single resource (usually HTML pages or any XML based resource), though relation with multiple resources is also possible. Upon request from a learner for a concept, the resources are adapted via link hiding and blocking, and presented to the learner in the form of a Hypermedia document.

The structure of the domain gets generated based on the concept graph and the links to these concepts are annotated along with the content on the page. However, the resources are not typed i.e. typical resources are HTML pages in AHA! and they cannot be categorized according to there type. A resource related to a concept can be an example of that concept, or an introduction of that concept, etc. The adaptation rules are associated with every document, and this is a step away from

generic adaptation mechanisms.

[Specht 1998] proposes another sequencer, ACE - adaptive courseware environment. It is inspired by many previous AEHS [Weber 1997, Brusilovsky 1996], they also try to enrich the concept-based representation with integrating different learning materials and their roles in the learning process. Similarly, to that of DCG, in ACE's concept graph, each node represents a concept or a set of concepts. Each concept is linked with different types of learning materials that explain different aspects of the concept. The edges of the domain structure represent prerequisite relations. Either a default strategy gets used to select the pedagogical resources for a concept or the course's author can manually define a sequence of resources for a concept. The authors argued that the manual plan can be modified, dynamically, according to the learner's competencies.

The rules for presentation planning are attached to every concept, which is hard to maintain in case of a large number of concepts in the domain model. The learning paths, through the concept graph, get defined manually by the authors.

[Heraud 2004, Heraud 2000] presents Pixed (Project Integrating eXperience in Distance Learning) a research project attempting to use learners' interaction logs gathered as learning episodes to provide contextual help for learners trying to navigate their way through an ITS. They use a Notional Graph, linking together *notions to learn* by relations representing precedence (prerequisites) between notions and representing the mastering level to fit the prerequisites. Resources are connected to each notion and users could add intermediate *notions* (with corresponding resources) to their own course (for example by navigating out of the official course on the web) and they can add alternative resources for a particular notion. A scenario gets represented by a specific notional graph. Depending on the results of the tests (for each notion), the path in the graph gets adapted. In case of failure of the proposed scenario, it was possible for the learner to reuse a successful scenario of an other student by adapting it to his own context.

They propose a model to describe a learning session, a way to log learners' interaction and to decompose it into learning episodes. Then they used case-based reasoning paradigm to offer contextual help to the learner. A leaner navigates in the notional graph, if he is in need of help, PIXED uses CBR techniques to present the learner with an adapted path used by other learners in similar situations. The CBR cycle of PIXED is shown in the figure 2.9.

They presented a new perspective by using CBR technology for adaptation. The Users' traces get used to provide adaptation. The interaction traces of the past users get used to guide the new users. They have the possibility to guide the learner in a step-by-step manner towards his pedagogical goals.

Recall that a course sequencer selects the best resource at any time based on the performance of the user. Though, not every system, has an explicit notion of a pedagogical scenario. The approaches present the possibility to adapt the scenario dynamically.

However, there are some limitations to these approaches. In some of these approaches, the authors manually define the learning paths. This is hard to maintain

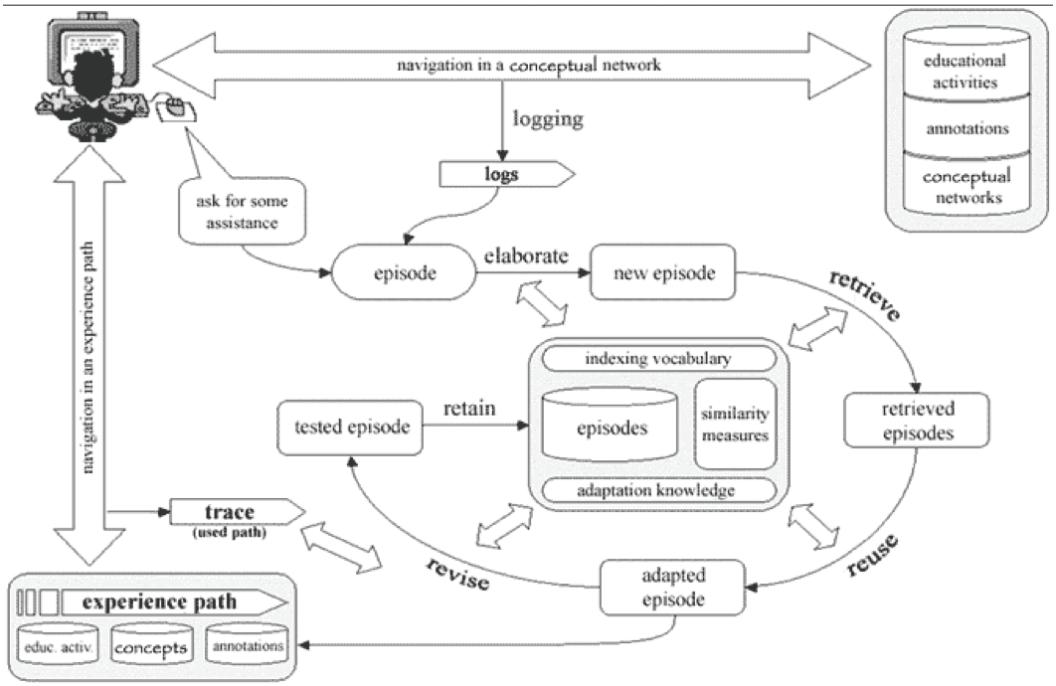


Figure 2.9: CBR cycle of PIXED [Heraud 2004, Heraud 2000]

in case of a large number of concepts. Add to it the fact that each learner can have a learning path and the problem becomes exponential.

Some of these approaches do not provide with the possibility for the dynamic adaptation of pedagogical resources, i.e. a pedagogical can be dynamically selected, and after the selection, it could be dynamically adapted according to the profile of the learner.

Whereas, in approaches like PIXED, which can guide a learner step-by-step towards his pedagogical goals, the adaptation provided is only experience-based and not expert based.

In the next section, we present the review of course generators.

2.4.2 Course Generators

In addition to the course sequencers, there are course generators. According to [Ullrich 2010], course generators are defined as:

> A course (ware) generator assembles a sequence of educational resources that support a student in achieving his learning goals. The selection of the learning resources takes information about the learner into account, for instance his competencies and preferences. Course generation (CG) offers a middle way between pre-authored *one-size-fits-all* courseware and individual look-up of learning objects.

There are many examples of course generators presented in the literature like [Masthoff 2002, Caumanns 1998, Ahanger 1997, Kettel 2000]. Here, we present a select few, which offer the solutions closest to our defined criterion.

[Specht 2001, Kravcik 2004a] presents a Web-based Intelligent Design and Tutoring System (WINDS). This system is very similar to ACE (adaptive courseware environment) and hence suffer from the same limitations. [Shahin 2008] is another similar system.

[Libbrecht 2001a] presented a course generator for the platform **ActiveMath**. ActiveMath [Melis 2001, Melis 2006] is a web based ITS for mathematics. The generator generates a personalized course in a three-stage process. Step one is the **Retrieval of content**, where, given the learning goals of a learner, all the concepts and their corresponding educational resources get selected from a knowledge base, which is necessary to achieve those learning goals. Step two is the **Applying of pedagogical knowledge**, where the educational resources get filtered accordingly for the learner. Step three is the **Linearization** of the graph. This process results in a personalized path through the domain knowledge graph for a learner.

This generator serves specifically for the ActiveMath and keeps into account the specifics of the technical aspects of ActiveMath, hence it is not easy to use it in different contexts. Similar to all the other reviewed generators, pedagogical scenarios can only contain pedagogical resources for only one concept. This is not a major limitation from our point-of-view. However, the reasoning process to the selection of pedagogical resources uses an expert-system like approach, forcing to enter all the rules beforehand, therefore, making it difficult to maintain for a large knowledge base.

In addition to the traditional AI techniques, researchers have also used statistical techniques to either to select the best learning path or to recommend the best learning resource for a learner for e.g. Neural networks in [Idris 2009] and [Seridi 2004], particle swarm organization [De-Marcos 2008], mining-techniques in [Hsieh 2010], petri-nets in [Huang 2008b].

[Karampiperis 2005c, Karampiperis 2005a] also used statistical techniques to generate a course most suitable to the learner. Instead of first selecting the concepts and then for each concept selecting the educational resources, they first calculate all possible courses that reach a set of concepts and then select the best suited one, according to a utility function. The course generation process can be seen in the figure 2.10.

In addition to the traditional domain concept layer. They also maintain a *Learning Goals Layer*, which is a graph, where, the nodes represents goals and the edges the relations between those goals. Each goal contains a certain number of concepts. The concepts are in relation with educational resources (contained in the *Content Layer*). The educational resources are also in relation with each other via pedagogical relations.

Whenever, a learner selects a learning goal as a target, the generator selects all the concepts related to the learning goal, then selects all the pedagogical resources and the resources connected to those resources. This happens for all the

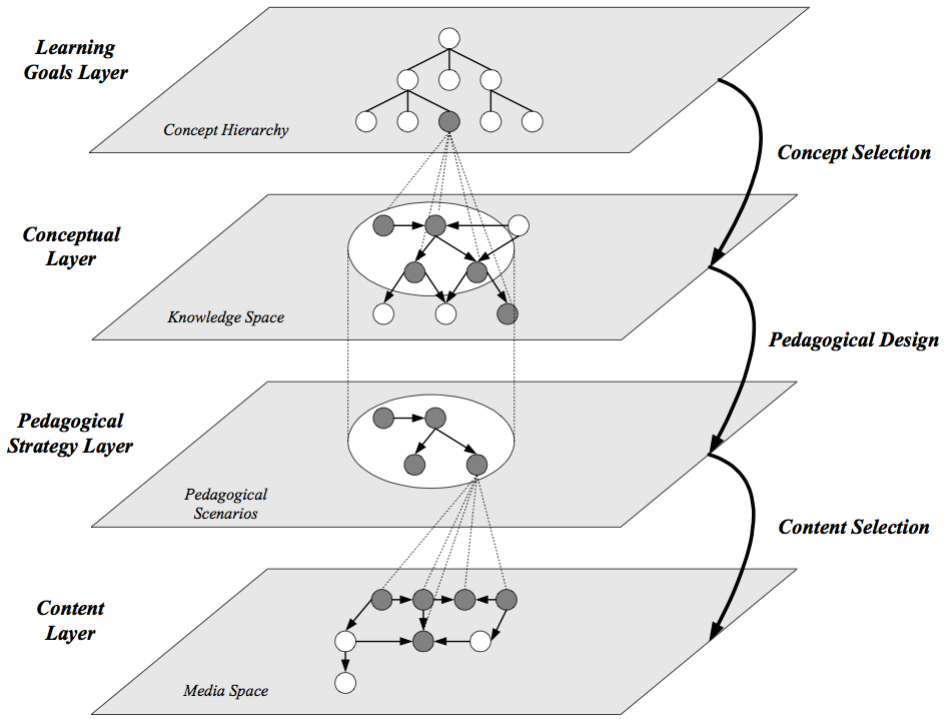


Figure 2.10: The scenario generation process of [Karampiperis 2005c]

selected concepts. This process generates many graphs (*Learning Paths Graph*). Then based on a utility function, which takes into account the learner competencies and preferences, their approach selects the best possible paths for the learner.

However, the relations between pedagogical resources are necessary for including other resources in the same scenario. This requirement makes the addition of pedagogical resources a complex process, since not only the pedagogical resource has to be related to a concept, it also has to be related with the other resources that the expert wants to include in the same scenario. Furthermore, not in every domain knowledge there is a relation between different pedagogical resources (like in the project CLES) for e.g.two educational resources for the same concept, but a different difficulty level might not have a relation with each other. Moreover, the educational resources get annotated, by the expert, with respect to a learning perspective of a learner. This process is cumbersome in case of a high number of pedagogical resources and learners.

In [Bouzeghoub 2005, Duitama 2005], the authors proposed an approach for the delivery of educational components to the user according to his/her abilities, preferences and pedagogical goals. Their proposed architecture can be shown in the figure 2.11.

They have three models namely the domain model, the user model and the educational component (EC) model. The EC model represents the Educational

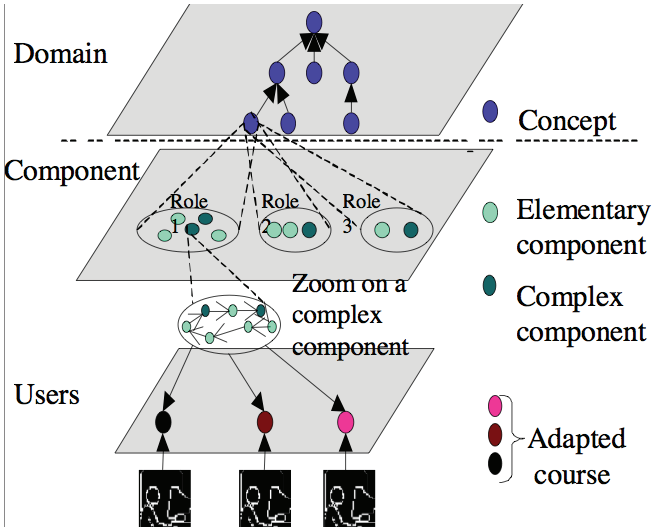


Figure 2.11: The architecture of Duitama [Bouzeghoub 2005, Duitama 2005]

resources, and they can either be atomic or structured. In case of the latter, an EC is further composed of different atomic or structured EC(s). They generate course in a course based, and goal based fashion. In the course based scenario, the user chooses a component, and the system adapt this component according to the user. While, in the goal based scenario, the user chooses the concept(s), and the system chooses the component.

The association of adaptation rules with each EC makes the generation process a bit more tedious for the author on one hand; on the other hand, it makes the adaptation static. The absence of the notion of pedagogical scenarios is also one of the limitations of this approach, from our point of view.

[Capuano 2002, Sangineto 2007] presents LIA (Learning Intelligent Advisor). In their paper, they presented with an explicit and well defined formalization of the user model, the cognitive states of the user, and the domain model where they define in detail the types and roles of relations. The user gives LIA with a set of target concepts. LIA creates a Presentation based on these target concepts. A Presentation is a collection of Learning Object(s) (LO) of two types. One is the LOs which explain the concepts, and the other is the LOs used to test the knowledge of the target concepts. LIA, then for each target concept, searches their respective atomic concepts. An atomic concept has no decomposition. Then for each atomic concept the system creates a presentation with suitable LOs for the learner.

Though it generates a course keeping in account the cognitive states of the user, the notion of pedagogical scenario (Presentation) is static. The structure is to always the same; starting from the presenting the explanation of a concept followed by tests, while this may work for certain cases, it is not generic in nature. They do not provide any provisions to adapt the pedagogical resources, as well.

[Viet 2006] has built ACGs system to create adaptive courses for each learner based on the learner's evaluating demand, ability, background and learning styles.

However, no notion of pedagogical scenarios gets presented by the authors. The authors did not discuss the dynamic adaptation of resources, as well.

[Carro 2003] presents the use of adaptation techniques to generate dynamically adaptive, and collaborative Web-based courses. These courses gets generated at runtime by selecting, at every step and for each student, the most suitable collaborative tasks to be proposed, the time at which they are presented, the problems to be solved, the most suitable partners to cooperate with and the collaborative tools to support the group cooperation.

However, their system is a rule-based system, thus making it difficult to add rules and the adaptation difficult. Furthermore, no notion of a pedagogical scenario gets presented by the authors. The authors have not discussed the general applicability of their approach in different contexts.

The course generator PAIGOS for the ActiveMath platform has been presented by [Ullrich 2007, Ullrich 2008, Ullrich 2009b, Ullrich 2010]. This generator tries to address the problems of course sequencer (lack of course structure) and course generator (lack of dynamic real-time adaptation). The work is between the course generation technique and course sequencing techniques. It proposes the use of formalized complex pedagogical scenarios. They have used Hierarchical Task Network Planning (HTN-planning) to formalize the pedagogical knowledge and generate the pedagogical scenarios. They also present the formalization of their scenarios. The idea behind is that they have a skeleton scenario where there are different tasks. Some of these tasks are static, generated like in course generation, and the others are dynamic tasks. These dynamic tasks get generated when the user requests them. Hence, these tasks gets generated by keeping into account the most up-to-date information about the user, like in a course sequencer.

They define different types of scenarios. The description of these scenarios can be seen in the table 2.2. The idea is that different learners at different times would like to study a concept or topic from different perspectives. If he is new to the topic he will like to *Discover* the topic, are if he is confident about his competency about a concept/topic then he may like to test his knowledge with *trainWithSingleExercise*.

Though the authors have detailed their approach, and it also adapts dynamically for a learner. However, the scenarios get generated given a concept or a set of concepts. This concept(s) get chosen either by the learner himself or by someone else. After the selection of concepts, then the scenarios can be generated by the HTN-planner. It could be possible that a learner is unable to select the set of concepts, which the learner should study to achieve his pedagogical goals. In this case, in order to guide the learner step-by-step towards his pedagogical goals, firstly, the concepts, that should be learned by the learner, have to be selected and then the scenarios should be generated.

Planning techniques have also been employed, by other approaches to generate courses. [Limongelli 2008] uses PDL planner to sequence learning resources keeping into account the learning styles of the learner. The pedagogical scenarios they define

Identifier	Description
discover	Discover and understand fundamentals in depth
rehearse	Address weak points
trainSet	Increase mastery of a set of fundamentals by training
guidedTour	Detailed information, including prerequisites
trainWithSingleExercise	Increase mastery using a single exercise
illustrate	Improve understanding by a sequence of examples
illustrateWithSingleExample	Improve understanding using a single example

Table 2.2: Scenarios description of PAIGOS [Ullrich 2007]

are quite simple. Since, they only consider learning resources they have the same limitations as that of PAIGOS.

[Keenoy 2004] presents *SeLeNe* (self e-learning networks) . In SeLeNe, a learner searches for educational resources using simple keyword-based queries that get matched against author and subject information. A *Trails and Adaptation service* personalizes the queries by reformulating and adding conditions to the query (e. g., the learner's language), and by ranking the results in order of relevance to the learner. The learner can request a personalized sequence of interactions through the resources (a trail). Trails get calculated based on relationship types that hold between resources. The adaptation provided by SeLeNe uses the ordering of the query results and adaptation of the query. The learner can define his goals. SeLeNe does not prove the adaptation knowledge of the learning resources.

Recall that a course generator generates a course one time by selecting the best resources possible considering the learner's profile. These approaches, unlike course sequencers, can present the user with a structure of the course. However, the lack of dynamic adaptation in the generated courses can frustrate a user, as the learner's competence can vary while interacting with the scenario. PAIGOS tries to addressee this issue by presenting an approach which lies between the course generators and course sequencer. However, it does not guide the learner step-by-step. These approaches do not provide the possibility to adapt a pedagogical resource. Since, all of these approaches consider only the pedagogical aspects; consequently, they do not take into account the serious game specificities. Therefore, it is difficult to use them with serious games in their actual form.

2.4.3 Summary

Table 2.3 shows the comparison of various approaches, reviewed above, according to our criterion (see page 14). This table gives an idea what currently exists in literature, which criteria they satisfy and where they lack. Off course, our objective is to satisfy all the criterion.

This table shows that none of the existing approaches satisfy all the criteria. None of these systems are designed to work with serious games, hence; they do not take into account the serious game specificities. Consequently, it is not easy to use them with serious games. Furthermore, when we also consider that a pedagogical resource can also be adapted according to a learner, then none of the systems take this property into account as well. Therefore, this PhD research work, propose a system that is capable of generating pedagogical scenarios, independent of the pedagogical domain, keeping into account the learner competencies and objectives. These scenarios will be generated keeping into account the serious game specificities, thus making our system oriented towards serious games as well. The scenario will guide the learner, step-by-step towards his learning goals. His interaction traces will be used to update the system and provide the learner with adaptive scenarios.

In the next chapter, we present our contributions in the form a system and its knowledge models. This system satisfies all the criterion.

	Domain Independence	Resource Adaptation	Traces for updating	Step-By-Step Guidance	Serious Game	Pedagogical Scenario
Generic Tutoring Environment (GTE) [Van Marcke 1990]	yes	no	no	no	no	no
Dynamic Courseware Generator (DCG) [Vassileva 1995]	yes	no	yes	yes	no	yes
AHA! [Bra 1998, Bra 2001]	yes	no	yes	no	no	not defined clearly
ACE [Specht 1998],	yes	no	yes	manually defined	no	yes
WINDS [Specht 2001, Kravcik 2004a]	yes	no	yes	manually defined	no	yes
ActiveMaths [Libbrecht 2001b]	yes - difficult to replicate	no	no	no	no	yes
[Karampiperis 2005c, Karampiperis 2005a]	yes	no	yes	yes	not verified	yes
[Bouzeghoub 2005, Duitama 2005]	yes	no	no	yes	no	no
LIA [Capuano 2002, Sangineto 2007]	yes	no	yes - not dynamic	yes	no	yes - static
ACGs [Viet 2006]	yes	no	no			
[Carro 2003]	no	no	yes	yes	no	no
PAIGOS [Ullrich 2007, Ullrich 2008]	yes - difficult to replicate	no	yes	no	no	yes
SeLeNe [Keenoy 2004]	yes	no	no	yes	no	no
PIXED [Heraud 2004, Heraud 2000]	yes	no	yes	yes	no	not structured

Table 2.3: Comparative table of different approaches

Chapter 3

Contributions

Contents

3.1 Introduction . . . 42
3.2 Knowledge Modelling . . . 42
3.2.1 Three Layer Architecture . . . 44
3.2.2 Domain Concept . . . 46
3.2.3 Pedagogical Resource . . . 50
3.2.4 Game Resource . . . 53
3.2.5 Learner Profile . . . 54
3.2.6 Presentation Model . . . 57
3.2.7 Adaptation Knowledge . . . 59
3.3 Scenario Generator . . . 60
3.4 Scenario Generation Algorithms . . . 63
3.4.1 Concept Selector . . . 63
3.4.2 Pedagogical Resource Selector . . . 68
3.4.3 Serious Resource Selector . . . 69
3.5 Learner Profile Updating Through Interaction Traces . . . 70
3.6 Formal Validation . . . 72
3.7 Summary . . . 74

This chapter presents our propositions to the two research questions. We present the knowledge modelling, this includes the knowledge organization in a three layer architecture (section 3.2.1), the models we have proposed for each of these layers (section 3.2.2, section 3.2.3, section 3.2.4), the learner profile (section 3.2.5), presentation model (section 3.2.6) and adaptation knowledge (section 3.2.7). The section 3.3 presents the model for scenario generation. Next, in the sections 3.4, we detail the algorithms that are used to generate the pedagogical scenarios. In the section 3.5, the updating of the learner profile through interaction traces is presented. The last section 3.6, presents the formal validation of our proposed models.

3.1 Introduction

As mentioned before, the objective of this research work is to propose models and processes that adapt the pedagogical content in a serious game according to the learner's competencies, skill and pedagogical goals. We have also identified two characteristics that are essential to the propositions we make in this research work. These characteristics are 1) the approach should be generic, and 2) the learner interaction traces should be used for continuous acquisition of knowledge. Two research questions serve as benchmarks for achieving the objectives with the desired characteristics. The research questions are

> **Question 1: What is the personalization knowledge to get for supporting the generation of adaptive pedagogical scenarios in a serious game environment? How to represent this knowledge?**

> **Question 2: What is the inference process for exploiting properly the personalization knowledge?**

The response of the first question requires, on the one hand, identifying the different types of knowledge that are necessary to represent the pedagogical domain as well as the serious game. On the other hand, it also requires organizing and modelling these different types of knowledge in a way that is in line with the proposed characteristics.

The contributions, which answer the first research question are presented in the section 3.2. To address the second research question, we propose a model of a scenario generator. This generator, presented in the section 3.3, makes use of different types of knowledge for generating adapted pedagogical scenarios for serious games.

3.2 Knowledge Modelling

The identification of the different types of knowledge requires an analysis of the related work. The related work has similar research objectives to our research objectives.

As mentioned in the previous chapter, this research work has common grounds with two types of approaches in particular, namely: *scenario generation in AEHS* and *scenario generation in serious games*. The objective of this analysis is, on the one hand, to list down the different types of knowledge elements used by these approaches. On the other hand, identifying the knowledge elements that could be required for answering the first research question.

The approaches that deal with generation of scenarios in AEHS reveal some interesting patterns. Almost all of them try to design or model the pedagogical or educational domain. The pedagogical domain model, in general, is a composition of the domain concepts and the pedagogical resources. A domain concept

is "an abstract representation of an information item from the application domain [De Bra 1999]". A pedagogical resource provides different types of information that can be used to support learning of a domain concept. Approaches like [Dagger 2005, Albert 2009, Kontopoulos 2008, Duitama 2005] propose approaches for modelling of domain concepts as well as the pedagogical resources.

In addition to the domain concepts and pedagogical resources, some authors have explicitly detailed other aspects of a pedagogical scenario. For example, [Vassileva 1996, Shahin 2008] model the manner in which the pedagogical resources should be organized in a pedagogical scenario. [Ullrich 2009a] proposes formal modelling of the pedagogical scenarios. While some of the approaches [Karampiperis 2005c, Cristea 2003] model the learning goals hierarchy.

In summary, the knowledge elements used by similar approaches for the scenario generation process in AEHS are: *Domain Concepts*, *Pedagogical Resources*, *Scenario Presentation Structure*, and *Learning Goals*.

In the context of scenario generation in Serious Games, several approaches, like [Burgos 2008, Moreno-Ger 2008b, Dung 2010, Carron 2007, Bieliková 2008] have identified to model the pedagogical resources or pedagogical content along with the serious game elements. The game elements describe the gaming environment that can contain objects like non-playing characters, decor, challenges, goals (both gaming and pedagogical), rewards, etc. [McNamara 2010, Mills 2007] propose to use the domain knowledge model during the design process of a game. Many approaches consider the game elements as a playground to access the pedagogical content or the pedagogical resources.

This analysis resulted in the identification of certain knowledge elements used by the scenario generation approaches for serious games. These elements are *Pedagogical Aspects* and *Game Resource*[1].

The potential list of all the knowledge elements that could be necessary to answer the first research question includes, *Domain Concepts*, *Pedagogical Resources*, *Scenario Presentation Structure*, *Learning Goals* and *Game Resources*.

In general, a graph like structure represents the *learning goals* [Cristea 2003, Karampiperis 2005b]. This graph usually is acyclic i.e. it does not have any cycles. Each node of this graph represents one or more domain concepts. The learning goals graph is highly domain dependent. Learning goals are useful for one domain; however, they cannot be generalized to be used with multiple pedagogical domains. The learning goals can simply be defined as a set of domain concepts. Consequently, we do not need an explicit model for the representation of learning goals.

The *scenario presentation structure* is a way of organizing the pedagogical resources in a pedagogical scenario. When generating a scenario, this structure is necessary in determining which types of resource are to be selected and how should they be organized. In general, the scenario presentation structure is generic in nature i.e. any scenario presentation structure can be used with any pedagogical domain

[1]Note that this analysis has been done through a very specific view point (i.e. our objectives) and is not meant gives a complete account of the serious game creation process.

or serious game. In fact, scenario presentation structure does not neither have a direct relation with the pedagogical domain nor the serious game. We also use this structure in the scenario generation process (see section 3.2.6). However, because of the structure's domain-independent nature it does contribute towards neither the pedagogical domain nor the serious game domain. Consequently, we do not consider it as essential for the resolution of the research question.

After analysing the related works and identifying the potential knowledge elements, we selected three elements required to address the research question. These elements are, *Domain Concepts*, *Pedagogical Resources* and *Game Resources*. Domain concepts and pedagogical resources represent the pedagogical domain. The game resources represent the serious game elements.

Any learner-centred approach should have a structure to represent the learner. We represent the learner in the form of a learner profile (see section 3.2.5). This profile contains the learner's competencies, preferences, skills and pedagogical objectives. In general, the learner-centred approaches consult the learner profile to provide adaptation to the learner. We use the learner profile for generating adaptive pedagogical scenarios according to a learner. The learner profile could be modelled independently of the pedagogical domain and the serious game i.e. the structure of the learner profile remains the same irrespective of the pedagogical domain and serious game. Consequently, the learner profile is not essential to answer the first research question.

The response to the first research not only requires the identification of the necessary knowledge elements, but also requires identifying and organizing the models required to represent the knowledge elements. To model the pedagogical domain, we need to model the domain concepts, as well as the pedagogical resources related to the domain concepts and the game resources.

The section 3.2.1 presents the organization of the three knowledge elements. The proposed organization ensures that the approach we propose remains generic in nature. We also present the modelling of the different knowledge elements.

3.2.1 Three Layer Architecture

The characteristic of being generic means that our approach should have the capacity to be used with a variety of pedagogical domains and serious games. This implies that any system designed using the proposed approach should be able to use any pedagogical domain with a variety of serious games and vice versa. This characteristic could be achieved by making sure that different knowledge elements are "loosely-coupled". This means that the impact of a change in any one knowledge element should be minimal on other knowledge elements. For example, if we want replace a pedagogical resource with another pedagogical resource, then this replacement should not force a change in the domain concepts or the game resources. The concept of loosely-coupling different elements of the system is quite popular in software engineering. In pure computer science terms, this idea referred to as "Separation of Concerns (SoC)". Dijkstra was the first one to use the term SoC

[Dijkstra 1974], the idea has since taken an essential place in software design. This idea is, in short terms, defined by [Win 2002] as:

> software should be decomposed in such a way that different "concerns" or aspects of the problem at hand are solved in well-separated modules or parts of the software.

This principle has shown its worth everywhere in software design sphere, like in the internet protocol (IP) which uses four different layers, and each layer's functioning is independent of the other. Hence, any change in anyone of the layer does not forces a change in the other layers. Another popular example is that of HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and Javascript. Here; we can also observe the SoC principle in full force i.e. HTML structures the webpage, CSS styles the webpage and Javascript manages the user interaction with the webpage.

Keeping in mind this principle, we propose a multilayer organization of the knowledge elements. The figure 3.1 shows this organization.

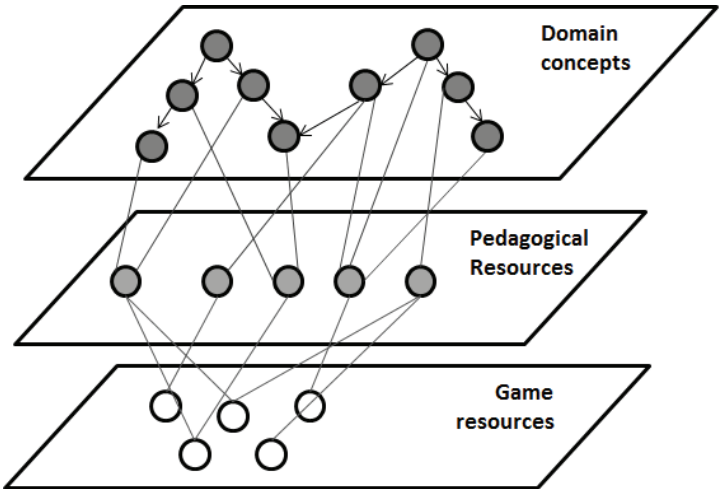


Figure 3.1: The three knowledge layers

1. The domain concepts layer: represents the abstract aspects of a pedagogical domain in the form of domain concepts.

2. The pedagogical resources layer: represents the concrete knowledge about a pedagogical domain in the form of pedagogical resources.

3. The game resources: represents the serious game resources.

This organization allows to separate different aspects of the scenario generation process. Thus, the game resources can be designed and organized without worrying

about the design aspects of a pedagogical domain. Likewise, a pedagogical resource can be replaced by other without a need to modify the first layer. Because the pedagogical resources are not "tightly-coupled" with the serious game resources, therefore, a pedagogical resource can be used with many serious game resources. Similarly, because of the "loose-coupling" between elements, a serious game resource can also be used by many pedagogical resources.

In the figure 3.1, it can be observed that each layer contains some arrows, intra and inter layers. The inter layer arrows represent the connection between the elements of one layer to the elements of the other layers. We have proposed models for representing these relations along with the models for other layers. The intra layer arrows can be observed in the domain concept layer. These arrows represent the relations between the domain concepts. These relations are pedagogical in nature.

In the section 3.2.2, we present the modelling of the domain concept layer, which includes the domain concepts and the relation between them. The section 3.2.3 presents the modelling of the pedagogical resources and how they connect to the domain concepts. The section 3.2.4 presents the modelling of the serious game resources, and how they connect with the pedagogical resources.

3.2.2 Domain Concept

This layer contains the representation of the domain concept knowledge in the form of concepts and the relation between these concepts[2]. A concept can be defined as "an abstract representation of an information item from the application domain. [De Bra 1999]".

In order to define the formalization to be used to model the domain concepts, we conducted a review of the related approaches.

The idea behind domain concept modelling, from a pedagogical scenario generator point of view, is to organize the domain concept knowledge in such a manner that it would be possible to create pedagogically correct scenarios.

The simplest examples of designing the domain concept knowledge can be seen in [Peachey 1986, Mitrovic 1996, Vassileva 1990, Leinhardt 1998], where only one relation (pre-requisite) exists between the concepts. This relation means that if one domain concept X is a pre-requisite of another domain concept Y, then it is necessary to teach X before Y.

The modelling of domain concepts can be a complicated task, requiring more than one pedagogical type of relations for correctly representing the pedagogical domain. For example, the approaches presented in [Vassileva 1996, Brusilovsky 2003a] use concepts structure to represent the domain concept knowledge. They propose to model the relations between the domain concepts using AND/OR graphs [Nilsson 1971]. AND/OR graphs can represent formalism of the domain concepts, because of their power of expressiveness they can be used as a decomposable pro-

[2]When we refer to the first layer as "Domain Concept Knowledge" and this refers both the domain concepts and the relations between those concepts. The "Domain Concept" refers only to the domain concepts and not the relations between them.

duction rule systems [**?**]. The nodes in a AND/OR graph represent concepts and the arcs in the graph represent relationships between the concepts. There are many other possible semantic relationships, for example, causal, temporal, analogy, simple prerequisite, etc.

[Ahmad 2007] uses relations like *explains*, *elaborates*, etc.. Similarly, approaches like [Bieliková 2006, Dagger 2005, Farrell 2004, Fischer 2001, Duitama 2005] use custom defined, UML-type relationships to define their domain concept knowledge.

We can summarize this related work by saying that there is no universally accepted way of modelling domain concepts. The modelling should be done according to the requirements of the domain. For our objective, we propose the formalization of a *metamodel* for modelling the domain concepts and the relations between these domain concepts. The motivation behind this proposition is to leave the modelling of the eventual domain concepts and the relations open for implementation. The system using this metamodel will have the possibility to define many relation types between concepts and assign any kinds of properties to the domain concepts. This proposition will make it possible for this research work to be used with a variety of pedagogical domains.

We formalize the model of domain concepts (DM) as an acyclic graph:

$$DM =< C, R > \tag{3.1}$$

Where,

C : represents the set of domain concepts of the pedagogical domain.

R : represents the set of relations between the domain concepts.

C is defined as:

$$C =< id, P > \tag{3.2}$$

Where,

id : unique identifier of the domain concept.

P : properties of the concept. These properties are of type <attribute, value>, where *attribute* is the name of the property, and *value* is the property's value. For example, <"name","concept name">, <"description","text describing the concept">, <"context of use","the text describing the context in which the concept should be studied">, etc.

And, R is defined as:

$$R =< C_{From}, T, RC > \tag{3.3}$$

Where,

C$_{\text{From}}$: the origin or source domain concept of the relation.

T : type of relation. The following elements represents T:

$$T =< Name, Description, F_{TYPE} > \tag{3.4}$$

Where,

Name : the name of the relation.

Description : the description of the relation.

FTYPE : the function used to calculate the impact of the domain concept CFrom on the domain concept CTo linked via this relation. These values can be used to update the learner's profile in the system. More detail of this updating is provided in the section 3.5.

RC : Set of relation of concepts is defined as:

$$RC =< C_{To}, F, Value > \tag{3.5}$$

Where,

CTo : target domain concept of the relation, the direction of relation is from CFrom to CTo

F : Function: this function calculates the value used by FTYPE

Value : if the function F is absent, then FTYPE uses Value to calculate the dependencies between the concepts of this relation.

There is a plethora of relation types described throughout the literature [Wu 1998, Albert 2009]. In order to show the use of the proposed metamodel to model real-world relation types, we present the modelling of some relations. We use these relations to model project CLES's knowledge. Some of these relations are not a contribution of this research. Many other approaches ([Karampiperis 2005a, Duitama 2005]) also use these relations. These relations are Has-Parts, Required, Order, Type-Of and Parallel.

Has-Parts (X, Y1... Yn): This relation indicates that the domain concepts y1, y2 ... yn are the sub-concepts of the domain concept x. For example, Has-Parts (Perception, auditory perception, visual perception), Has-Parts (Maths, Addition, Multiplication, Division, Subtraction). This means that the knowledge of Maths contains four sub-concepts, namely: Addition, Multiplication, Division, and Subtraction. In other words, the knowledge of Maths is equal to the combined knowledge of Addition, Multiplication, Division and Subtraction. If a learner wants to master the concept X, then he has to master the concepts Y1... Yn.

Required (X,Y): This relation indicates that to learn concept X, the concept Y has to be learned sufficiently. For example, Required (Oral Language, Perception).

Order (X, Y): This relation means that it is preferable to present the concept X before concept Y. For example, Order (Visual perception, Auditory perception) i.e. it is better to present Visual perception before Auditive perception.

Type-Of (X, Y): This relation shows that the domain concept Y is a type-of the domain concept X. This relation can be considered as a specialization relation, for example, Type-Of (Master in Science, Master in Computer Science).

Parallel (X, Y): This relation indicates that the domain concepts X and Y are parallel concepts and must be studied and tested simultaneously. For example Parallel (Oral Language, Memory) i.e. Memory and Oral Language should be studied and tested together.

The semantics of the value propagated between two domain concepts depends on the function F_{Type}. For example, Has-Parts relation could mean that the contribution made by the parts contributes towards the whole. A relation like Required could represent the minimum mastery of domain concept(s) required, by a learner to learn a domain concept.

The proposed meta-model of a pedagogical relation can be used to model almost any kinds of relations. The pedagogical relation of type AND can be represented by using the Required relation type. For example, if we need to represent that a learner needs to learn the domain concept A AND the domain concept B AND the domain concept C before start learning the domain concept X. We can use the Required relation like Required(X,A), Required(X,B), and Required(X,C).

Similarly, the pedagogical relation of type OR can be represented by using a combination of Required and Type-Of relation types. For example, if we need to represent that a learner needs to learn the domain concept A OR the domain concept B OR the domain concept C before start learning the domain concept X. We can use the following formulation: Required(X,T), Type-Of(T,A), Type-Of(T,B), and Type-Of(T,C). The formulation reads, the learner needs to learn T before X, and T is either A or B or C. This implies that the learner requires either A or B or C to learn X.

We present an example of the use of the proposed pedagogical relations.

3.2.2.1 Example

Recall the example presented in the section 1.1. The student Jack wants to learn a concept Division, In order to understand the concept Division, Jack requires mastering sufficiently the concepts Addition and Multiplication. The relation between Division, Addition and Multiplication can be represented by using the relation *Required*. Based on our meta-models, we present the modelling of this relation below:

- R_1 = <Division, $T_{Required}$. $RC_{Addition}$>
- R_2 = <Division, $T_{Required}$. $RC_{Multiplication}$>
- $T_{Required}$ = <"Required", "This relation defines the pre-requisite relationship between two concepts", $F_{Required}$>
- $RC_{Addition}$: <Addition, null, 50%>
- $RC_{Multiplication}$: <Multiplication, null, 10%>

This relation means that the learner requires to master sufficiently the domain concept Addition and Multiplication before start learning the domain concept Division. In this example, $RC_{Addition}$ and $RC_{Multiplication}$ means that the learner needs to master at least 50% of the domain concept Addition and master at least 10% of the domain concept Multiplication to learn the domain concept Division.

Now, that we have described the domain concept modelling formalizations. The next section presents the modelling of the pedagogical resources.

3.2.3 Pedagogical Resource

While the domain concepts are the abstract representation of the pedagogical domain's information item, pedagogical resources are the concrete information about that domain concept. These resources provide different types of information that can be used to support learning of a concept. The occurrence of different types of resources can be easily observed in any text-book.

[Koper 2000, Ullrich 2007] have identified the properties that should be kept into account for making the pedagogical resource model globally reusable. These properties are:

Domain independence : This means that our model should cater all types of pedagogical resources independent of the pedagogical domain.

Pedagogical flexibility : Any author or course designer could use our model to implement any pedagogical strategy.

Completeness : The model should cover as many types of pedagogical resources as possible.

Machine process-ability : This model should have the property to make the pedagogical resource easy to find and re-use.

There are currently some standards used to describe a pedagogical resource like Learning Object Metadata (LOM) [Committee 2002]. These standards use a plethora of properties that are often irrelevant to a course designer. Consequently, it poses an extra burden on the course designer. For example, LOM has a property called learningResourceType that categorizes a pedagogical resource. This property mixes two aspects:

1. Pedagogical resource's types, for examples : exercise, simulation, questionnaire, etc.

2. Pedagogical resource's form, for example : graph, table, slide, etc.

This organization further poses a decision making problem, as pointed by [Ullrich 2007]. The decision making process needs to differentiate between the technical form of the resource and the resource's pedagogical type. Furthermore, LOM specification does not include some of the many commonly used types like definition, example, etc.

Formally, the model we propose to represent a pedagogical resource (PR) is:

$$PR =< Id, Type, Parameters, EvaluationFunction, Solution, Characteristics, ConceptRelations > \tag{3.6}$$

Where,

id : The unique identifier of the resource.

Type : The type of the resource. This field is very important and is, sometimes, very specific to a particular pedagogical domain. Therefore, we have left this up-to the designer to define their proper types. A quick review of the literature can produce a ton of types for example: theorem, law of nature, procedure, fact, introduction, remark, conclusion, explanation, exercise, exploration, invitation, real-world problem, proof, demonstration, example, counter example, etc. For our application, **we have used *mini-games* as pedagogical resources**. An example of a mini game has been presented in chapter 1 (section 1.2).

Parameters : In the case where the behaviour of a pedagogical resource can be adapted, then the adaptation can be provided via assigning different values to these parameters. The manipulation of these parameters can be used to tweak the difficulty level of the pedagogical resource. They are in the form of $<$*attribute, value*$>$, where *attribute* defines the name of the parameter, and *value* defines the value given to that parameter.

Evaluation Function (Optional): If the resource allows to evaluate the learner's mastery, like through an exercise or a Multiple Choice Questions (MCQ), then this function evaluates the learner's response.

Solution (Optional): If the resource is an evaluative resource, then the solution is the correct response of this pedagogical resource. The solution can be represented via a value. For example, for the question "3 + 2 = ?", the value "5" is the solution.

Characteristics : represent the meta-information about the resource, for example, the author's name, date of creation, language etc. They are in the form of <attribute, value>, where *attribute* defines the name of the characteristic, and *value* defines the value given to that characteristic.

ConceptRelation : this contains the concepts that are in relation with the resource. These concepts are of the form:

<Domain Concept, Required Knowledge>, where:

Domain Concept : The Id of the concept related to the resource.

Required Knowledge : The mastery of the concept required by a learner to access this resource. This is a value that represents the *Difficulty Level* of a pedagogical resource. An attribute-value pair of the form *<concept, value>* represents this property. The *concept* represents the domain concept. The *value* represents the mastery of the concept required, by the learner, to access this resource. This value represents the *Difficulty Level* of a pedagogical resource.

ImpactFunction (Optional) : If the pedagogical resource is an evaluative resource then this function calculates the impact of the learner's response on the learner's mastery on the domain concept.

Note that a pedagogical resource can be in relation with more than one concept and vice versa.

To demonstrate the use of this model the next section presents an example.

3.2.3.1 Example

Retake the example of section 1.1. Suppose we have a pedagogical resource P1 in relation with the concept Division. The modelling of this resource is as follows:

P1 = <Id_{P1}, "definition", null, null, null, <text,"Division is often shown in algebra and science by placing the dividend over the divisor with a horizontal line, also called a vinculum or fraction bar, between them.">,«$ID_{Division}$,0%,null» >

This presentation reads as follows: the resource P1 has an id "Id_{P1}", it is of the type "definition", the text of this definition is "....by placing the dividend over the.....". P1 relates to the concept Division and it does not require any prior mastery of the concept to be utilized by the learner. The 'null' value of the 'ImpactFunction' shows that P1 does not contribute towards the learner's mastery of Division.

The concept Division is in relation with the concept "Addition", there is another pedagogical resource P2. P2 is an exercise and is in relation to both the concepts Division and Addition. The model describing P2 is as follows:

P2 = <Id$_{P2}$, "exercise", null, FunctionEvaluation$_{P2}$, "25", <"question phrase","What is the solution for 5 * 10 / 2?">, < <ID$_{Division}$,10%,ImpactFunction$_{Division}$>, <ID$_{Multiplication}$,30%,ImpactFunction$_{Multiplication}$> > >

This representation reads as follows: P2 has an id "Id$_{P2}$", P2 is an exercise. FunctionEvaluation$_{P2}$ evaluates the learner's response. The value "25" represents the correct response of P2. The phrase "What is the solution for 5 * 10 / 2?" represents the question phrase of the exercise. P2 can only be presented to a learner who has at-least a mastery of 30 % of the concept Multiplication. The learner needs a little mastery (10%) of the domain concept Division to access P2. The function ImpactFunction$_{Division}$ updates the value of the learner's mastery of Division in the learner profile based on the learner's response. Similarly, the function ImpactFunction$_{Multiplication}$ updates the value of the learner's mastery of Multiplication in the learner profile based on the learner's response.

3.2.4 Game Resource

We consider a *game resource* as either a static object or an object attributed with an interactive or proactive behaviour according to the game. We consider only the game resources that are in relation with a pedagogical resource. This means that we aim to model just the elements that present the pedagogical resources. We do not aim to model the serious game construction process. In the proposed formalizations, a game resource has the possibility to be in relation with one or more pedagogical resources.

Formally, the model we propose to represent a game resource (GR) is:

$$GR =< Id, Characteristics, PedagogicalRelations > \tag{3.7}$$

Where,

id : The unique identifier of the game resource.

Characteristics : They represent the meta-information about the resource, for example, the author's name, date of creation, language etc. All these information in the form of <attribute, value>, where *attribute* defines the name of the characteristics, and *value* defines the value given to that characteristic.

PedagogicalRelations : This represents the pedagogical resources IDs that are in relation with the game resource.

In response to the first research question, we presented the organization of the three knowledge elements in a multi-layer architecture. We also presented the formalization of the meta-model for the representation of these knowledge. Before we present the contributions, which we have proposed in response to the second research question, we present, in the next sections, the modelling of others knowledge

needed to generate adaptive scenarios, namely : learner profile, presentation model and adaptation knowledge.

3.2.5 Learner Profile

The moment one decides to propose an approach for providing the learner with adapted learning scenarios, it becomes imperative for the approach to take into consideration some representation of the learner. This representation can contain all types of information that the system may need to perform the adaptation, for example, learner's background information, competencies about a particular pedagogical domain, vital statistics, cognitive abilities, skills, preferences, beliefs, habits, etc. The terms **learner model** or **learner profile** or **user model** refer to the representation of a learner in a system.

It is defined by [Brusilovsky 2007] as:

> a representation of information about an individual user that is essential for an adaptive system to provide the adaptation effect, i.e., to behave differently for different users. For example, when the user searches for relevant information, the system can adaptively select and prioritize the most relevant items....When the user reaches a particular page, the system can present the content adaptively.

We consider the learner model as an abstract representation of a learner in a system, whereas a learner profile shows the representation of a learner at any instance of time. However, most of the time the different authors use the terms learner model and learner profile interchangeably [Brusilovsky 2007]. Each learner profile represents a person in the system. This profile is, in general, the only way to provide personalized services to a person. Therefore, after its initial creation it should have the capacity to evolve with the evolution in the learner mastery of domain concepts. The proposed approach aims at providing adaptive pedagogical scenario; therefore, it is necessary to have a learner model. This learner model should contain information relevant to the scenario generation process. This model should include information about the learner's masteries of the pedagogical domain and the possibility to keep a record of the learner's interactions. Useful assumptions about the learner can be made by using the learner's interaction history.

There are a number of systems that have tried to create a learner model according to their needs. Therefore, before proposing a learner model, we looked at the existing techniques of learner modelling to draw inspiration.

[Brusilovsky 2007] has performed an excellent review of the existing learner modelling techniques. In general, the information modelled about a learner is the learner's background, goals, interest, individual traits, knowledge and sometimes the context of work. Different approaches have used different techniques to model different kinds of needs in the learner model like scalar model, overlay model, concept-level models, keywords models, etc.

In order to represent the learner's mastery of the domain concepts, we have decided to use the *overlay model*. In terms of content, the profile is composed of general information about the learner, its skills on the domain concepts, based on *overlay model*, and its *interaction traces*.

An overlay model is a subset of the domain concept knowledge. It represents the learner's knowledge as the subset of the expert's knowledge. [Vanlehn 1987] defines an overlay model as:

> Some student modelling approaches can represent only missing conceptions. Conceptually, the student model is a proper subset of the expert model. Such student models are called overlay models because the student model can be visualized as a piece of paper with holes punched in it that is laid over the expert model, permitting only some knowledge to be accessible. A student model, therefore, consists of the expert model plus a list of items that are missing.

In [Clauzel 2011], a modelled interaction traces is defined as:

> a trace explicitly associated with its trace model. A trace model is an ontology that describes the vocabulary of the trace. A trace results from the observation of the interactions between a user and her system, it has a temporal extension related to the time of the observation. A trace is composed of observed elements (or obsels) representing the interaction between the user and the system. Each obsel has a set of attributes/values that are related to the temporal extension of the trace (e.g. it can be related to an instant or a temporal interval)...a trace can contain relations between obsels...A trace model is then a set of observed element types and relations types.

The formalization that we propose to model a learner's profile is as follows:

id : The unique identifier of the learner.

Personal information : The information like the learner's name, date of birth, e-mail, education background, etc.

Motivational Level : This value can help us in selecting the difficulty level of the pedagogical resources. If a learner has high motivation, than a more difficult exercise can be presented to him. However, if a learner has low motivation, then an easier exercise can be selected for him. According to [Pintrich 1999], there is a strong correlation between motivation and performance. This property makes used of this correlation.

Preferences : This property allows selecting the pedagogical resources that correspond to the preferences of the learner. The learner can describe his preferences in the form of cognitive categories as described by [Felder 1988].

Some examples of the Felder's cognitive categories are Sensing versus Intuitive Learner, Visual versus Verbal Learner, Active versus Reflective learner, Sequential versus Global Learner, etc.

Competences : These represent the overlay of the domain concepts. The learner's mastery of the domain concepts is kept as a score of the domain concepts (see section 3.2.2). Each competence is in the form of a tuple <Concept, Value>, where:

Concept : The id of the domain concept.

Value : The learner's mastery of the concept. This value can be qualitative or quantitative in nature.

Interaction traces : The traces represent the learner's interaction history. The idea behind is to track the entire learner's interaction. This will, on the one hand, help the system to update the learner profile and, on the other hand, help analysing the learner's interaction patterns. This analysis can help the expert to study the evolution of the learner's profile and make appropriate decisions for the learner. We can also use traces in order to make propositions, to the expert, about the potential modification in the domain concepts organization. **A trace represents a learning session of a learner**. Formally, we represent a trace **T** as follows:

$$T=< \text{Begin-Date, End-Date, Presentation Model, Pedagogical Goals, Pedagogical Scenario, } O_1, O_2 \ldots O_n>$$

Where,

Begin-Date : The time at which the learner starts interacting with the scenario.

End-Date : The time at which the learner stops interacting with the scenario.

Presentation Model : The presentation model used to structure the scenario presented to the learner in the session. The presentation model is detailed in the section 3.2.6. Keeping this information will help us in the validation of the effectiveness of the presentation model.

Pedagogical Goals : The learning goals of the session. A set of tuples <domain concept, value> represents the pedagogical objectives. The **domain concept** represents the 'ID' of the domain concept, and **value** represents the mastery of the domain concept the learner wants to achieve.

Pedagogical Scenario : Whenever, we generate a pedagogical scenario for a learner for a set of pedagogical objectives, this property records the generated scenario. It contains all the concepts, pedagogical resources and the game resources generated by the system.

O*i* : The observed elements, they are characterized by the following:

O_i=<Concept, Pedagogical Resource, Serious game resource, Pedagogical resource level, Learner's Response, Time of response, Evaluation of learner's response, Time-stamp, Changes in profile>

Where

Concept : The domain concept, which is being, interacted by the learner.

Pedagogical Resource : The pedagogical resource the learner is interacting with.

Serious game resource : While playing the game, which game resource is being interacted by the learner.

Pedagogical resource level : As mentioned earlier, it is possible to adapt a pedagogical resource by using its parameters. This adaptation can take the form of setting the appropriate difficulty level for the pedagogical resource according to the learner's profile. If the domain concept related a pedagogical resource is not sufficiently mastered by a learner, then a lower difficulty level of the pedagogical resource can be chosen. The adaptation knowledge (described in the section 3.2.7) of a pedagogical resource provides the adaptation. We keep track of the pedagogical resource's difficulty level in this property.

Learner's Response : In case the learner is interaction with a pedagogical resource of the type exercise or test, then this property contains the learner's response to the pedagogical resource.

Time of response : The time taken by the learner to respond to a pedagogical resource of type test or exercise.

Evaluation of learner's response : Every pedagogical resource of type test has also a evaluation function. This function evaluates the learner's response and the result of this response is kept in this property.

Time-stamp : The exact time stamp of the event.

Changes in profile : In the case, where the learner's interaction with a pedagogical resource results in a change in his profile, this property records this change.

We describe in the next section the presentation model, which we have proposed.

3.2.6 Presentation Model

In general, almost all the scenario generators not only select the pedagogical resources, but they also organize them according to a predefined structure. This

structure defines the type of the pedagogical resources to be selected. Some approaches [Brusilovsky 2003b, Vassileva 1992] refer to this organization as **presentation plan** while some call it **formalized scenarios** [Ullrich 2010]. Whatever the name, the idea is to organize the generated scenarios according to some learning theory. We call this structure **presentation model**.

The structures of the presentation models are different by different authors. The figure 2.8 shows the presentation model for a task of type "Give Exercise" [Van Marcke 1992]. The scenario starts by an exercise ("Make Exercise"), then a test to "Verify" the learner's response, and in case the learner has performed badly, then it will present the learner with a remedy "Remedy". This presentation model is domain independent.

Figure 3.2 shows the presentation model proposed in [Ullrich 2010]. This model allows a learner to learn a domain concept with different perspectives.

1. Description. The course starts with a description of its aim and structure. Then, for each goal concept selected by the learner, the following sections are created:
2. Introduction. This section motivates the usefulness of the concept using adequate resources such as examples and introduction texts. It also contains the unknown prerequisites concepts.
3. Develop. This section presents the concept and illustrates how it can be applied.
4. Proof. For some concepts, such as mathematical theorems, proofs or other evidence supporting the concepts is presented.
5. Practice. This section provides opportunities to train the concept.
6. Connect. This section illustrates the connections between the current concept and related concepts.
7. Reflection. Each course closes with a reflection section, which provides the learner with opportunity to reflect on what he has learned in the course.

Figure 3.2: Formalized scenario "Discover" [Ullrich 2010]

[Sangineto 2007] presents a similar but simpler approach that represents the presentation model as an ordered set of learning resources followed by a set of test resources.

In our presentation model, we try to do something very similar to these works. Thus, we propose to allow scenario designers to implement whichever educational strategy they want to implement. This model is very similar to that of [Sangineto 2007], but more complex in nature. It allows the placement of any type of pedagogical resource in any order. The Presentation Model (PM) we propose can be described as,

$$\begin{aligned} PM =< & PR^1_{Type}\left\{Annotation^1 \dots Annotation^M\right\}, \\ & PR^2_{Type}\left\{Annotation^1 \dots Annotation^M\right\} \dots \\ & PR^N_{Type}\left\{Annotation^1 \dots Annotation^M\right\} > \end{aligned} \tag{3.8}$$

Where

- PR_{Type} refers to the type of a pedagogical resource and,
- N,M > 0, and,

- $\text{PRType}^1 < \text{PRType}^2 < \ldots \text{PRType}^{N-1} < \text{PRType}^N$ is an **ordered list** of pedagogical resources in function of their types (refer to section 3.2.3 for more details), and
- *Annotation* represents the different annotations that can be used in the PM.

For example, a presentation model can contain the following list:

1. Introduction
2. Definition
3. Example
4. Example
5. Counter Example
6. Description
7. Exercise
8. Exercise

Following this presentation model, any domain concept selected in the pedagogical scenario starts with presenting a pedagogical resource of type "introduction" of the domain concept, followed by a pedagogical resource of type "definition" and then a couple of pedagogical resources of type "examples" and a "counter example". Then a pedagogical resource of type "description" related to the domain concept, followed by two pedagogical resources of type "exercise".

The course designer defines in advance the structure of the presentation model, but the actual selection of the pedagogical resources depends upon the learner's profile.

The presentation model can be made more complex than it already is by adding annotations. These annotations could tell the module to do some extra work. For example, an annotation of type "@IncludePreRequisite Introduction" will force the module to search the pedagogical resources of type "Introduction" of all the prerequisite concepts. Another example can be "@Obligatory" which will force the inclusion of pedagogical resource irrespective of whether the learner is aware of the resource or not.

The next section presents the description of the adaptation knowledge that allows to adjust the difficulty level of pedagogical resource according to the learner.

3.2.7 Adaptation Knowledge

The idea behind adapting a pedagogical resource is straightforward; some of the pedagogical resources have the possibility to be parameterized[3]. This parameterization means that the behaviour of a pedagogical resource can be tweaked. This

[3]See for example, the parameters of a mini-game related to memory presented in the section 1.2 of chapter 1

tweaking can take many forms like modifying the access modalities of the pedagogical resource, the difficulty level, the appearance, etc. This research focuses mainly on the tweaking of the difficulty level of a pedagogical resource. An adaptable pedagogical resource allows a domain expert to design only one pedagogical resource with many levels of difficulties, instead of putting as many pedagogical resources as there are difficulty levels.

The actual nature of the process of adaptation can be varied. It can be a set of rules, an expert system or an automatic exercise generator. We present the protocol that needs to be followed by any process implementing the Adaptation Knowledge. The protocol is as follows: the process receives as input the learner's profile and the learner's pedagogical objectives. The process then chooses the right parameters values for the pedagogical resources based on the learner's masteries of domain concepts (defined in the learner's profile via "Competences") and the pedagogical objectives. The process can use a different strategy for every pedagogical resource or use the same strategy for all the pedagogical resources. The choice remains with the process designer/domain expert.

To illustrate this protocol take, for example, an adaptable pedagogical resource P. P is in relation with a domain concept C. P has two parameters $param_1$, and $param_2$. The parameter $param_1$ can assume one of the following values: $value_{11}$, $value_{12}$ and $value_{13}$. The possible values for $param_2$ are: $value_{21}$, $value_{22}$ and $value_{23}$. An adaptation process can use different combinations of parameter values of P to adapt it to the learner. For illustration purposes, suppose that the adaptation process is a rule-based system. This process can adapt P by using rules of the type listed below.

Rule 1 : If (learner's mastery of C $>$ 10 and $<$30) and
If (the pedagogical objective for C $\geq$ 40 and $<$ 50) then
param1 = value11
param2 = value21

Rule 2 : If (learner's mastery of C $>$ 40 and $<$90) and
If (the pedagogical objective for C = 100) then
param1 = value12
param2 = value23

The choice, of different combinations of the parameters can make the pedagogical resource difficult or easy.

Up to this point, we have defined the models for all the knowledge elements that are necessary to answer the second research question. The next section presents the contributions towards the model of a scenario generator.

3.3 Scenario Generator

The figure 3.3 shows the principal model of the architecture of the proposed scenario generator. Recall that we have organized the three essential knowledge elements in

three layers (c.f section 3.2.1); the model we propose for a scenario generator also generates the pedagogical scenario in three steps. The first step deals with the domain concepts layer, the next with the pedagogical resources layer and the final step with the game resource layer.

The dotted vertical lines in the figure 3.3 divides the architecture into three parts. The left part of the figure represents the knowledge models related to the pedagogical domain and the serious game. The part on the right hand side represents the models that are necessary to adapt and structure and the difficulty level of the generated scenario. Finally, the middle part represents the scenario generation process.

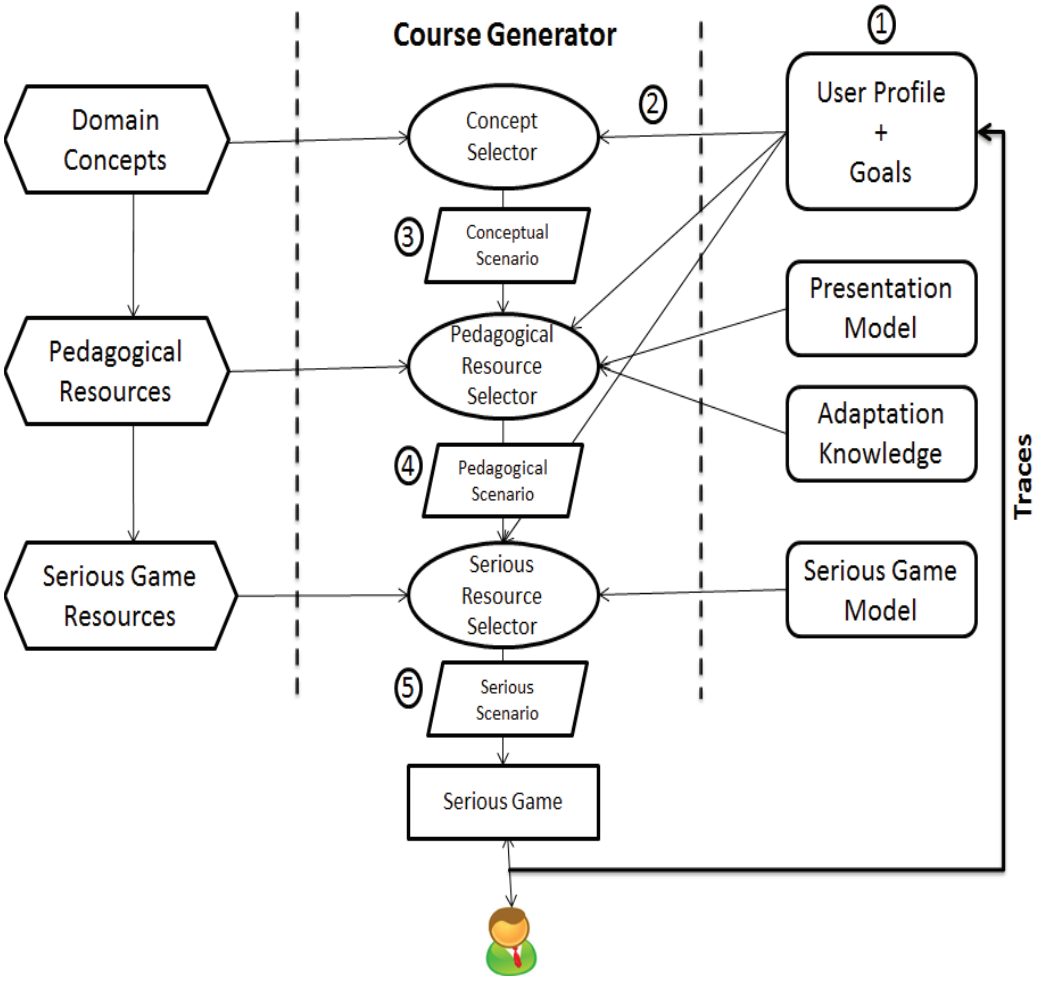


Figure 3.3: Principal model of scenario generator

The process of generating pedagogical scenario is as follows: (1) the domain's expert(s) enters the domain's knowledge and the learner profile in the system according to models presented in the previous sections. In each session, the generator receives as input the pedagogical goals of the session as a list of learning objective as <concept, value>, where the *concept* is the target domain concept, which the

learner wants to master, and the *value* represents the level of domain concept's mastery the learner wants to achieve. The selection of the pedagogical goals can be done either by the learner or the domain expert.

In the step (2), the generator, according to the selected goals and the learner's profile, selects the domain concepts from the domain concepts graph. The selected concepts are those that are necessary for the learner to achieve his pedagogical goals. The module **Concept Selector** performs this selection by consulting the learner profile to verify, which concepts are already sufficiently mastered by the learner. The output of this module is the **Conceptual Scenario**. The conceptual scenario contains all the domain concepts, and their levels of difficulty, which are necessary to achieve the learning goals of the learner. Formally we define a conceptual scenario as:

Conceptual Scenario = { $<$<RC$_1$, $RMAS_1$>, ... , <RC$_N$, $RMAS_N$> $>$},
where, RC : Required domain concept, and RMAS : Required domain concept's mastery

In the step (3), the module **Pedagogical Resource Selector** receives the conceptual scenario as input. The purpose of this module is to select for each domain concept in the conceptual scenario the appropriate pedagogical resource. For this, the generator consults the presentation model and the learner's profile to select these pedagogical resources. In this process, the generator uses interaction traces in order to avoid the repeated selection of the same pedagogical resources. In case the pedagogical resources are adaptable, the generator consults the adaptation knowledge to adapt their difficulty levels for the learner. The output of this module is a **Pedagogical scenario.** This scenario comprises pedagogical resources with their adapted parameters. Formally we define a Pedagogical scenario as:

Pedagogical Scenario = { $<$<RC$_1$, $PR_1^1 < params > \ldots PR_M^1 < params >$>
$\ldots$<RC$_N$, $PR_1^N < params >$, ...,$PR_P^N < params >$> $>$},
where, N,M,P ≥ 0, RC = Required domain concept, PR = Pedagogical Resources, and params = the adapted parameters for the learner

In the step (4), the module **Serious Resource Selector** receives as input the pedagogical scenario. This module is responsible for associating the pedagogical resources with the serious game resources (the game resources are defined in the Serious Game Model). The output of this module is the **Serious Scenario**. Formally we define a Serious Scenario as:

Serious Scenario = { $<$<RC$_1$, $SGR_1(PR_1^1 < params >)$,
...,$SGR_2(PR_M^1 < params >>)$...<RC$_N$, $SGR_1(PR_1^N < params >)$,
...,$SGR_N(PR_M^N < params >)> >$},
where, RC : Required Concept, PR = Pedagogical Resources, SGR : Serious Game Resources and params : the adapted parameters for the learner

The serious game receives as input the serious scenario. The serious game engine will initialize itself with the pedagogical resources and the serious game resources. The learner interacts with the pedagogical scenario via the serious game. All these interactions are stored in the learner traces. The generator uses the traces to update the learner's profile, and modify the pedagogical scenarios according to the performance of the learner.

In the next section, we propose the pseudo-algorithms for the three scenario generator modules.

3.4 Scenario Generation Algorithms

As mentioned in the previous section, three modules namely *Concept Selector*, *Pedagogical Resource Selector* and *Serious Resource Selector* handle the process of pedagogical scenario generation given pedagogical goals and learner's profile. We have already described the general functionality of these modules. In this section, we present the algorithms for these modules.

3.4.1 Concept Selector

The purpose of this module is to select and order a list of domain concepts, which a learner requires to achieve his pedagogical goals. The input to this module is the pedagogical goals. These pedagogical goals represent the domain concepts a learner wants to learn. The output of this module is a set of domain concepts and their mastery, which a learner needs to learn for achieving his pedagogical goals.

The algorithm 3.1 describes the functioning principle of the **Concept Selector** module. The input to this algorithm is a list called **TargetConceptList**. This list contains a set of tuples <Domain Concept (C), Required Mastery (RM)>, where, **Domain Concept and Required Mastery** signify the pedagogical goals in terms of domain concept and their target mastery respectively. The expected output is a list called **Conceptual Scenario**. It contains the domain concepts and their masteries required by a learner.

The algorithm starts by checking, for every domain concept (C) in the TargetConceptList (Lines 3), whether the learner has sufficient mastery of C. If he has sufficient mastery of C, then the algorithm ignores C and treats the next domain concept. If the learner does not have sufficient mastery of C, then the algorithm searches all the domain concepts that are necessary to learn C. For this, the algorithm analyses the relation between the other domain concepts and C.

Recall that each *relation* has one source concept and one or more target concept(s), and it has a type. Each relation type, noted RT, has to provide a function **Selection-Strategy$_{\text{Type}}$**. Different RTs use different strategies to implement Selection-Strategy$_{\text{Type}}$. Selection-Strategy$_{\text{Type}}$ searches for the domain concepts (**SDC**), which are in relation of type RT with C. It also calculates the masteries of SDCs, which the learner requires for his pedagogical goals. Selection-Strategy$_{\text{Type}}$ proposes only the SDCs, which the learner needs to learn but has not mastered

Algorithm 3.1 Concept Selector

Input: TargetConceptList = { < <$\text{Concept}_1, RM_1$>, ... , <$\text{Concept}_N, RM_N$> >} , where RM = Required Mastery , Learner Profile.

Output: Conceptual Scenario = { < <$\text{RC}_1, RMAS_1$>, ... , and <$\text{RC}_N, RMAS_N$> >}, where, RC = Required domain concept, RMAS = Required domain concept's mastery

DATA: Conceptual Scenario = null

```
1: function ConceptSelector \\
2:     foreach Domain Concept C ∈ TargetConceptList do
3:         if C is not sufficiently mastered by the learner then
4:             foreach Relation Type RT∈ available list of relations do
5:                 Result ← F^RT_Type(C, RM_C)
6:                 if Result ≠ {} then
7:                     Conceptual Scenario ← Conceptual Scenario + Result
8:                 end if
9:             end foreach
10:        end if
11:    end foreach
12: end function
```

sufficiently. Selection-Strategy$_{\text{Type}}$ needs, as input, the target domain concept TC and TC's required mastery, which the learner needs to learn. The Selection-Strategy$_{\text{Type}}$'s output is in the form { < <RC_1, RCM_1> ... <RC_N, RCM_N> >}, where, RC = Required Domain Concept, RCM = RC's Mastery.

Afterwards, the module stores the output of Selection-Strategy$_{\text{Type}}$ in a variable *Result*. The module adds Result to the conceptual scenario. This process is repeated for all the Cs in the TargetConceptList.

The calculation of Selection-Strategy$^{RT}_{Type}$ depends on the type of relations. For demonstration purposes, we present Selection-Strategy$_{\text{Type}}$'s algorithms for three different types of relations namely: Required, Has-Parts and Type-Of. Note that these algorithms present a certain manner for representing the functioning of Selection-Strategy$_{\text{Type}}$. Different systems can use different algorithms for Selection-Strategy$_{\text{Type}}$.

The next section describes the algorithm for the Selection-Strategy$_{\text{Type}}$ related to Has-Part.

3.4.1.1 Has-Parts

The algorithm 3.2 shows the working of Selection-Strategy$_{\text{Type}}$ for the relation Has-Part. The variable ResultList represents the output.

The algorithm in line 2 searches for all the domain concepts that are in relation of type Has-Parts with TC. A list HasPartList maintains the result of the search.

Algorithm 3.2 Has-Part function

Input: Target Domain Concept = TC, Target Mastery = TM
Output: ResultList : { < <RC$_1$, RCM_1> ... <RC$_N$, RCM_N> >},
where, RC: Required Domain Concept, RCM : RC's Mastery
DATA: ResultList = null

```
1: function HasPartFunction
2:     HasPartList ← Search all domain concepts in relation of type Has-Parts
                     with TC.
3:     Participation ← Calculate the mastery level, which the learner should have
                     for all domain concepts in HasPartList to achieve his
                     pedagogical goals.
4:     foreach Domain Concept HPC ∈ HasPartList do
5:         if the learner does not master sufficiently HPC then
6:             ResultList ← ResultList + <HPC, Participation>
7:             ResultList ← ResultList + ConceptSelector(HPC, Participation)
8:         end if
9:     end foreach
10: end function
```

At line 3, the algorithm calculates the mastery of every domain concept in the HasPartList, which the learner should have. One way to perform this calculation is to divide the TM with the number of domain concepts in the HasPartList. Off course, this calculation is only possible when the mastery levels are in numeric form. The variable Participation stores the result of this calculation.

In line 4-8, the algorithm verifies for every domain concept (HPC) in the HasPartList, whether the learner has sufficient mastery of HPC. If not, then the algorithm adds the HPC and the variable Participation to the ResultList. Then the algorithm calls ConceptSelector with HPC and Participation as input. The algorithm also adds the output of this call to the ResultList. The purpose of this call is to repeat the same process with HPC.

For example, a learner chooses to learn a domain concept A with the mastery level of "50%". A is in a relation of type Has-Part with concepts A1 and A2. Using the proposed models, this relation can be modelled as follows:

- R_A = <A, $T_{\text{Has-Parts}}$, RC1, RC2>
- RC1: <A1, null, 50>
- RC2: <A1, null, 50>

The algorithm starts by searching for all the domain concepts that are in relation of type Has-Parts with A. In this case, these domain concepts are A1 and A2. The algorithm will then calculate the Participation for A1 and A2. The required mastery level is "50%", so the Participation will be "25%" (Required Mastery (50%) / number of concepts (2) = 25%). Afterwards, the algorithm verifies whether the

learner masters sufficiently A1 and A2. The ResultList includes the domain concepts and their masteries, which the learner needs to learn.

The next section shows the algorithm for the relation type Required.

3.4.1.2 Required

The algorithm 3.3 shows the working of Selection-Strategy$_{\text{Type}}$ for the relation Required. The variable ResultList represents the output.

Algorithm 3.3 Required Type function

Input: Target Domain Concept = TC, Target Mastery = TM
Output: ResultList : $\{ < <\text{RC}_1, RCM_1> \ldots <\text{RC}_N, Comp_N> >\}$,
where, RC = Required Domain Concept, RCM = RC's Mastery
DATA: ResultList = null

```
1: function RequiredTypeFunction
2:     RequiredTypeList ← Search all domain concepts in relation of type
                          Required with TC.
3:     foreach Domain Concept RTC ∈ RequiredTypeList do
4:         if the learner does not master sufficiently RTC then
5:             Participation ← || learner's mastery of RTC - the value in the relation
   between RTC and TC.||
6:             ResultList ← ResultList + <RTC, Participation>
7:             ResultList ← ResultList + ConceptSelector(RTC, Participation)
8:         end if
9:     end foreach
10: end function
```

The algorithm in line 2 searches for all the domain concepts that are in relation of type Required with TC. A list RequiredTypeList maintains the result of the search.

In line 3-8, the algorithm verifies for every concept (RTC) in the RequiredTypeList, whether the learner masters sufficiently RTC or not. If not, then the algorithm calculates the required participation of RTC. This calculation takes the absolute difference between the learner's mastery of RTC and the value defined in the relation between RTC and TC. The variable Particiaption holds the result fo the difference. Afterwards, the algorithm adds RTC and Participation to the ResultList. The algorithm then calls ConceptSelector with RTC and Participation as input. The algorithm also adds the output of this call to the ResultList. The purpose of this call is to repeat the same process with RTC.

For example, a learner chooses to learn a domain concept A with the mastery level "50%". A is in a relation of type Required with the domain concept B. This means that the learner needs to master B before learning A. Using the proposed models, this relation can be modelled as follows:

- R_{A} = <A, $\text{T}_{\text{Required}}$, RC_1>

- RC1: <B, null, 20>

The algorithm first searches for all the domain concepts that are in relation of type Required with A. In this case, it is only B. The algorithm then verifies whether the learner has sufficient mastery of B. If not then the algorithm calculate the Participation for B. If the learner knows nothing of B i.e. Learner's Mastery of B = 0%, then the Participation is 20% (‖ Learner's Mastery of B (0%) - relation value (20%) ‖ = 20%). The ResultList includes the domain concepts and their mastery levels, which the learner needs to learn.

The next section shows the algorithm for the relation type Type-Of.

3.4.1.3 Type-Of

The algorithm 3.4 shows the working of Selection-Strategy$_{\text{Type}}$ for the relation Type-Of. The variable ResultList represents the output.

Algorithm 3.4 Type Of function

Input: Target Domain Concept = TC, Target Mastery = TM
Output: ResultList : { < <RC_1, RCM_1> ... <$\text{RC}_N, Comp_N$> >},
where, RC = Required Domain Concept, RCM = RC's Mastery
DATA: ResultList = null

```
1:  function TypeOfFunction
2:      TypeOfList ← Search all domain concepts in relation of type Type-Of
                     with TC.
3:      if Learner masters sufficiently any domain concept in TypeOfList then
4:          return empty ResultList
5:      end if
6:      repeat
7:          Choose a Type Of Concept (TPC) in TypeOfList randomly
8:          if TPC is not already mastered by the the learner then
9:              Participation ← defined in relation
10:             ResultList ← ResultList + <TOC, Participation>
11:             ResultList ← ResultList + ConceptSelector(TOC, Participation)
12:             Exit Repeat Loop
13:         end if
14:     until all the TPCs are searched or none of the TOC in TypeOfList is added
                to the ResultList
15: end function
```

In algorithm in line 2 searches for all the domain concepts that are in relation of type Type-Of with TC. A list TypeOfList maintains the result of the search.

The algorithm in line 3 verifies whether the learner has already mastered sufficiently any domain concept (TOC) in the TypeOfList. If he has, then this would mean that the learner already knows TC. Consequently, the algorithm returns an empty list as output. If he has not mastered any TOC in TypeOfList, then the

algorithm chooses randomly a TOC from the TypeOfList. The algorithm adds the TOC and the variable Participation to the ResultList. The variable Participation represents the value defined in the relation between TOC and TC. Then the algorithm calls ConceptSelector with TOC and Participation as input. The algorithm adds the result of this call to the ResultList. The purpose of this call is to repeat the same process with TOC.

For example, a learner chooses to learn a domain concept A with the mastery level "50%". A is in a relation of type Type-Of with the domain concept B and the domain concept C. This means that the learner needs to either master B or C to master A. Using the proposed models, this relation can be modelled as follows:

- R_A = <A, $T_{Type\text{-}Of}$, RC_1, RC_2>
- RC_1: <B, null, 20>
- RC_1: <C, null, 20>

The algorithm first searches for all the domain concepts that are in relation of type Type-Of with A. In this case, they are B and C. Then the algorithm verifies whether the learner has mastered sufficiently B or C. If he has, then the learner has already mastered A and does not need any further learning. If the learner has not mastered B and C, then the algorithm chooses randomly a domain concept. Let's suppose that the algorithm chooses C. Then the algorithm uses the variable Participation to store the value defined in the relation between C and A. The ResultList includes the C along with the variable Participation.

The next section describes the use of the conceptual scenario by the module "pedagogical resource selector".

3.4.2 Pedagogical Resource Selector

The purpose of this module is to select the appropriate pedagogical resources for every concept in the Conceptual Scenario. For this, the module uses the Presentation Model (PM), Pedagogical Goals (PG) and the Learner Profile (LP). The output of this module is the *Pedagogical Scenario*. The Pedagogical Scenario contains a list of pedagogical resources for every domain concept in the conceptual scenario. The module also adapts the pedagogical resources using the adaptation knowledge. The algorithm 3.5 presents the working of this module.

The selection process goes as follows: firstly, for each domain concept (C) in the conceptual scenario, the algorithm searches for the pedagogical resources (PR), in relation with C, of type T as described in the Presentation Model (PM) (line 4). If there are more than one PRs of type T associated with C, then the module add the PR, which the learner has not seen or not mastered sufficiently (line 6). The module performs this verification by analysing the learner's interaction traces stored in the profile. The module also consults the pedagogical resource's adaptation knowledge to select its parameters according to the learner's profile and his pedagogical goals. Finally, the algorithm adds the pedagogical resource to the **PedagogicalScenario**.

Algorithm 3.5 GeneratePedagogicalResources

Input: Learner Profile (LP), Presentation Model(PM),
Conceptual Scenario (CS), Pedagogical Goals (PG)

Output: Pedagogical Scenario = { $<$ $<$RC$_1$, PR_1^1 $<$ *params* $>$, ...,PR_M^1 $<$ *params* $>>$,...,$<$RC$_N$, PR_1^N $<$ *params* $>$, ...,PR_M^N $<$ *params* $>>$ $>$}, where, RC = Required Concept, PR = Pedagogical Resources, and params = the adapted parameters for the learner

DATA: PedagogicalScenario = null

```
1: function GeneratePedagogicalResources \\
2:     foreach Concept C ∈ Conceptual Scenario do
3:         foreach resource type (T) in PM do
4:             ResourceList ← Search repository for resources of type T
                              related with C
5:             foreach pedagogical resource PR in ResourceList do
6:                 if the learner has not already seen the PR and the PR is not
                          included in PedagogicalScenario then
7:                     params_PR ← AdaptationModel_PR(LP,PG)
8:                     PedagogicalScenario ← (c, PR< params_PR >)
9:                 end if
10:            end foreach
11:        end foreach
12:    end foreach
13: end function
```

PedagogicalScenario is sent as input to the module serious resource selector (section 3.4.3). This module links the pedagogical resources with the serious game resources.

3.4.3 Serious Resource Selector

This module associates the pedagogical resources, in the Pedagogical Scenario, with the serious game resources according to the learner's profile and Serious Game Model (SGM). The output of this module is a list called Serious Scenario. This list contains the concepts of the conceptual scenarios, pedagogical scenario's pedagogical resources, and the serious game resources initialized with the adapted pedagogical resources.

This algorithm 3.6 shows the working of this module. The module starts by selecting a serious game resource (SGR) for each of the pedagogical resources (PR) in the PedagogicalScenario. The SGR can be an object in the gaming environment like a chair, a table, or a Non-Playing Character (NPC). Then the module consults the learner profile to verify whether the SGR is appropriate for the learner. If SGR is appropriate, then the module adds the it to the list.

We have mentioned earlier that all interactions between the learner and the seri-

Algorithm 3.6 GenerateSeriousGameResources

Input: LearnerProfile (UP), Serious Game Model(SGM), PedagogicalScenario (PS)

Output: SeriousScenario = { < <RC$_1$, $SGR_1(PR_1^1 < params >)$, ...,$SGR_2(PR_M^1 < params >>)$...<RC$_N$, $SGR_1(PR_1^N < params >)$, ...,$SGR_N(PR_M^N < params >)$> >}, where, RC = Required Concept, PR = Pedagogical Resources, and params = the adapted parameters for the learner

DATA: SeriousScenario = null

```
1: function GenerateSeriousGameResources
2:     foreach Concept C ∈ PS do
3:         foreach pedagogical resource PR of C in PS do
4:             SGR ← Find a Serious Game Resource (SGR) related with the PR
5:             if SGR is not appropriate for the learner then
6:                 Ignore this resource and continue with the next one
7:             else
8:                 SeriousScenario ← (c, SGR initialized with PR)
9:             end if
10:        end foreach
11:    end foreach
12: end function
```

ous game are stored in interaction traces. The learner profile records the interaction traces. In the section 3.5, we present the process of learner profile updating using the traces.

3.5 Learner Profile Updating Through Interaction Traces

The updating of learner profile means the updating of values associated with the Concept Competence. This updating is based on the learner's performances on the pedagogical resources of type *test* (exercise, problems, MCQ, question, etc). We illustrate this principle by an example.

Suppose, a learner has to interact with a pedagogical resource Y of type *test*. The resource Y is in relation with the domain concept C. The learner responds to the pedagogical resource Y. The evaluation function (see section 3.2.3) associated with Y evaluates the learner's answer. Since Y is an evaluative resource, the ConceptRelation of Y has an ImpactFunction. The "Impact Function" updates the value of the learner's mastery C in the learner's profile.

As mentioned earlier, the process of updating the learner's profile is necessary in order to keep track of the learner's evolving competencies. This updating affects the accuracy of the pedagogical scenarios proposed to the learner, which in turn will help increase the learner's performance.

The process of updating the values in Concept Competence of a learner's profile is as follows: each evaluative pedagogical resource has an *Evaluation Function* that evaluates the learner's response for that resource. Some pedagogical resource can also be adapted by using their parameters. The "Adaptation Knowledge" adjusts these parameters for each learner. These parameters can make the resource easy or difficult. The "Adapted Difficulty Level" denotes this variation in difficulties. Note that the "Adapted Difficulty Level" represents a pedagogical resource's level of difficulty, which adds to the "Difficulty Level" represented by the "Required Knowledge".

The learner interacts with an evaluative pedagogical resource and gives a response. The pedagogical resource's "Evaluation Function" evaluates the learner's response. The "Evaluation Value" represents this evaluation. The "Evaluation Function" keeps into account the actual response and the time taken by the learner, to respond. Now, the "Impact Function" takes into account the "Evaluation Value" and the pedagogical resource's level of difficulty and calculates a "Update value". If the pedagogical resource can be adapted then the "Adapted Difficulty Level" is:

pedagogical resource's "Adapted Difficulty Level" = *Function* ("Difficulty Level", "Level of difficulty chosen by the Adaptation Knowledge")

If the pedagogical resource cannot be adapted then it's "Adapted Difficulty Level" is:

pedagogical resource's "Adapted Difficulty Level" = "Difficulty Level"

The "Impact Function" updates the value of the learner's mastery of the domain concept with the "Update value". This function is of the form:

updated value of the learner's mastery of the domain concept in his profile = *Function* ("Evaluation Value", "Adapted Difficulty Level")

For example, suppose a concept **C** associated with a resource **P**. **Flt** denotes the "Impact Function". The **Fp** represents P's "Evaluation Function". The P's "Adapted Difficulty Level" is **L**. The learner's response to P is **R** and the time taken by the learner is **T**. The "Evaluation Function" Fp calculates an "Evaluation Value" **E** based on R and T.

$$E = F_P(R, T) \tag{3.9}$$

The function Fimf represents the "Impact Function". Fimf calculates the updates value of the learner's mastery of C as follows:

$$Updated\, value\, of\, Learner's\, Mastery\, of\, C = F_{IMF}(E, L) \tag{3.10}$$

In the next section, we present the formal validations of all the proposed models.

3.6 Formal Validation

Recall that we have identified (chapter 2) the characteristics, which the proposed approach should satisfy. These characteristics are:

- Domain independent Architecture
- Flexible scenario structure
- Step-By-Step learner guiding
- Adaptation of Pedagogical Resources
- Continuous Knowledge Acquisition
- Serious Game oriented

In this section we show that the contribution proposed in this chapter satisfy these characteristics.

Domain independent Architecture: In the section 3.2.2, we presented the meta-models for representing a pedagogical domain. These models represent a pedagogical domain in terms of domain concepts and the relation between these domain concepts. The domain concepts are abstract entity of a pedagogical domain. We tested the models by successfully modelling some of the domain concepts belonging to Mathematics.

A domain model can be:

(a) A single concept (description and attributes) -> no intra-layer relationship is possible. One or more relations with the pedagogical resources layer (inter-layer) are possible.

(b) There is more than one concept. Relations are possible between the concepts (intra). Each concept in the domain graph, thus, can also has "inter" relations (as shown in a).

As intra-layer relations between the domain concepts are independent of the inter-layer relations, the conceptual model represented by the intra-layer concepts relations, the pedagogical relations are not impacted by changes in the conceptual model, in fact the possible modifications are:

1. modification of a intra-layer relation between concepts. As there is an independence with inter-layer relations, the pedagogical resources are not impacted.
2. removal of a concept, the concept and its relations (intra-layer) with other concepts are deleted. The (inter-layer) relation between the concept and the pedagogical resource is also deleted but without having an impact on the pedagogical resource itself.

The same argument can be demonstrated for showing the independence between the pedagogical resources and the serious game resources.

Attention: this implies that intra layer relations are independent from the inter layer relations, which means that the semantics of a inter layer relation does not depend on intra layer relations and vice versa. We do not allow the composition of inter and intra layer relations.

Flexible scenario structure: By flexibility in a scenario structure we mean to allow the domain expert to define the organization of his scenario. For this, we have to allow the expert to define the ordering of the pedagogical resources in a scenario. In the section 3.2.6, we presented a model that can be used to structure a pedagogical scenario. This models can represent any organization of pedagogical resources in a pedagogical scenario. The course designer/domain expert can use this model to implement a variety of learning theories. He needs to identify the types of pedagogical resources he wants in the scenario and the scenario generator searches the repository that type of resource. This resource is selected according to the learner's profile. This shows the flexibility in structuring pedagogical scenarios offered by the proposed models.

Step-By-Step learner guiding: By this we mean not only allow the learner his target concepts with adaptive learning activities, but also to help the learner in learning the concepts, which are required by the learner to achieve his pedagogical goals.

We proposed a model of an architecture of a scenario generator in the section 3.3. The proposed model allows the learner to guide the learner starting from what the learner already masters towards his pedagogical goals. While generating pedagogical scenarios for a learner, the generator starts by generating the **Conceptual Scenario**. This conceptual scenario contains all the domain concepts that the learner requires to achieve his pedagogical goals. This way the generator allows, not only, the learner to learn the domain concepts in the pedagogical goals, but also, the domain concepts that will make the learning easier.

Adaptation of Pedagogical Resources: By this we mean to allow adaptable pedagogical resources in the resource repository. The generator not only selects the resource that is appropriate for the learner, but also adapt the adaptable resource dynamically according to the learner.

For this we allowed the designer to define the parameters of a pedagogical resource. These parameters can be used by the generator to adapt the resource. Consequently, our generator after selecting a resource also consults a module "Adaptation Knowledge" (section 3.2.7) to adapt the pedagogical resource. The section 3.2.7 demonstrates the adaptation of pedagogical resources.

Continuous Knowledge Acquisition: By this mean to use the learners' interaction as knowledge source for making assumptions about the learner and to

provide adaptation to the learner.

As we have mentioned earlier, when the interaction between a learner and the serious game generate the learner's interaction traces. We have proposed to record, continuously, these traces in the learner's profile. In the learner profile, these traces serve as knowledge sources for updating the learner's profile (section 3.5) and the pedagogical scenarios.

Serious Game oriented: We have proposed a three-layer organization of the pedagogical domain and the serious game (section 3.2.1). This organization allows to use pedagogical scenarios in serious games. The proposed scenario generator associates the pedagogical resources with the serious game resources.

3.7 Summary

This chapter presented our contributions for answering the first two research questions. The questions require identifying the knowledge models required to generate adaptive pedagogical scenarios in serious games and to propose a model for a pedagogical scenario generator. To answer these questions, we present the knowledge models for modelling domain concepts, pedagogical resources, and serious game resources. We also proposed a multi-layer organization of these models. This organization makes the knowledge models domain independent; thus, allowing representing various pedagogical domains. We also proposed some other models, which are necessary to generate pedagogical scenarios. These models include learner profile, presentation model, and the adaptation knowledge.

We also presented the model of a pedagogical scenario generator. This generator generates scenarios in three steps. These steps to the three layer organization of the knowledge elements. We also presented algorithms, which we have used to generate pedagogical scenarios.

We also showed how to update the learner profile using his interaction traces. In the end, we presented the formal validations of the propositions.

Chapter 4

GOALS: Generator Of Adaptive Learning Scenarios

Contents

4.1 Objectives of GOALS 76
4.2 Different Types of Users 77
4.2.1 System Administrator 78
4.2.2 Domain Expert 78
4.2.3 Learner 78
4.3 Configuration of GOALS by the expert 79
4.3.1 Projects Management 79
4.3.2 Learners Management 80
4.3.3 Knowledge Editor 82
4.3.4 Presentation Model 88
4.3.5 Learner Profile 89
4.3.6 Scenario Generator 90
4.4 Scenario Generation in GOALS by the learner 91
4.5 Illustrative Example 92
4.6 Technical Architecture 94
4.6.1 Presentation Layer 95
4.6.2 Business Layer 98
4.6.3 Data Access Layer and Resource Layer 99

This chapter describes the functionality and technical architecture of the platform GOALS. This platform implements the theoretical models described in chapter 3. This chapter is organized as follows : the section 4.1 describes the need for GOALS and its general functionalities. GOALS has different types of users, namely: administrator, expert and learner. The section 4.2 presents the role of each of them. Section 4.3 shows the different interfaces, of GOALS, required by the expert, to enter the domain knowledge. In the section 4.4, we present interfaces to allow learners to generate pedagogical scenarios. Section 4.6 discusses the technical architecture of GOALS.

4.1 Objectives of GOALS

In the previous chapters, we have discussed the theoretical aspects of our propositions for a generic generator of a pedagogical scenarios in serious games. Therefore, it was necessary to put theory into practice for real-world utilization. For this to happen, it is imperative to have, at least, a running prototype of the proposed approach.

Hence, we have developed an on-line GOALS. GOALS is an abbreviation for **G**enerator **O**f **A**daptive **L**earning **S**cenarios. The purpose of GOALS, on the one hand, is to present the opportunity to pedagogical domain experts, of different pedagogical domains, to present their learners with personalized pedagogical scenarios via a variety of serious games. On the other hand, GOALS also allows serious game designers to make their games available to be used with many pedagogical domains.

However, developing GOALS does not simply means to implement the models and algorithms. This means to develop a platform, which allows its users, to manage the complete course designing process and the necessary parts of the serious game. It also includes the knowledge management process, the users (domain experts, learners and system administrators), and the pedagogical scenarios. Furthermore, GOALS should also allow learners to interact with the scenarios via GOALS and view the evolution of their profile.

The step of determining all of these functional requirements is arguably the most important one. By functional requirements, we mean to [Malan 1999] "capture the intended behaviour of the system. This behaviour may be expressed as services, tasks or functions the system is required to perform."

In the paragraphs that follow, we outline all the functionalities that we had to implement in GOALS.

Multiple Users : We have designed GOALS for use with multiple domains; therefore, it is necessary to have the possibility to allow multiple pedagogical domain expert or serious game designer to manage their proper knowledge models. In addition, there will be many types of users who will be using GOALS for different purposes. There will be domain experts, for designing the domain knowledge models; learners, to use the pedagogical scenarios, and administrators, who will be responsible to manage the technical aspects of GOALS.

Multiple Control Panels : Multiple types of users will access the GOALS platform; therefore, it is necessary to show these users only what they need to know. For example, a learner has nothing to do with the domain modelling process. Hence, a learner using GOALS only needs to see the final configuration of the domain model. Similarly, the technical aspects should be managed by the administrators, the domain expert is responsible for managing domain knowledge modelling and the learner profiles.

Multiple Learners : When there is a pedagogical domain, there will be a set of learners associated with it. GOALS should provide a domain expert with

possibility to manage learners' profiles. This management comes in the form of setting the right values in the profile. These values correspond to a learner's masteries regarding a particular pedagogical domain. He can also associate or disassociate a learner to a project.

Multiple Projects : A domain expert might be interested in managing multiple domain knowledge models. This allows a domain expert to be responsible for multiple courses at the same time. We use the term "Project" to refer to a particular pedagogical domain and knowledge related to it. GOALS should allow a domain expert the possibility to create and manage multiple projects.

Knowledge Management : Associated with each project there is a certain set of knowledge. This knowledge includes the domain concepts, the pedagogical resources and the serious game resources and models that are necessary to generate pedagogical scenarios. GOALS allows the creation of concepts, their properties and the relation between them. This goes for all the other resources. Furthermore, the knowledge creation process should be intuitive in nature i.e. it should be easy to do. We organize the domain knowledge in the form of a graph; therefore, it is desirable to have a visual knowledge management process.

Scenario Generation : The experts should be able to generate different scenarios for different learners and validate the results. This process helps the expert to pre-visualize the scenarios, which the learners are going to use. Moreover, a learner should also be able to use GOALS to generate scenarios and interact with them.

Visualization : We store the learners interactions in the "Interaction traces". We use these traces to update the learners' profiles. Therefore, it is necessary to show the different aspects of these traces. The domain expert may find it difficult to interpret the traces in their raw form. Hence, we need to transform these traces to show them to the expert. The expert should be allowed to make queries on the traces.

We have identified the functional requirements that are necessary for the platform GOALS. In the next section, we present the different types of users that can use the GOALS platform.

4.2 Different Types of Users

As we have mentioned earlier (in section 4.1), we have different types of users: the administrators, the domain experts and the learners. Each of these users has a different purpose for using GOALS. In this section, we describe their roles.

4.2.1 System Administrator

As the name suggests, a system administrator administers the use of GOALS. This means to ensure that the technical aspects of the platform are in order. He has the possibility to grant access to different users and define their roles. He can access the database directly and change it if necessary. The figure 4.1 shows the interface to access the database.

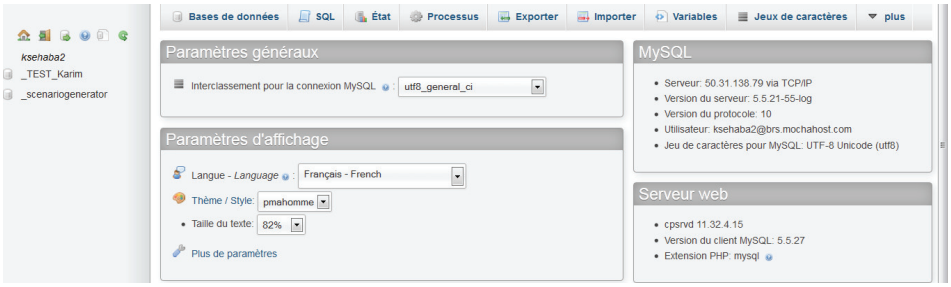


Figure 4.1: DataBase Access Interface

The other users like learners or domain experts can ask for the administrator's assistance in case of a technical problem. They can also make a demand for the addition or modification of functionalities in GOALS. The administrator can also manage the resources uploaded on the server by the domain experts.

The next section describes the role of the domain expert.

4.2.2 Domain Expert

The domain expert is the user who is responsible for managing a pedagogical domain(s) organized in project(s). In GOALS, a *Project* represents all the knowledge related to a pedagogical domain. This includes the domain concept knowledge, the pedagogical resource knowledge, the game resource knowledge, learner profiles, the presentation model, the adaptation knowledge and the possibility to generate and test scenarios for different learners and pedagogical objectives.

A domain expert can use GOALS to manage multiple projects and learners. The managing of projects means to create, delete, and modify the information related to a project. Similarly, it is also possible in GOALS for an expert to manage the learners by adding, deleting and modifying the information related to the learners. Furthermore, GOALS also allow to associate or dissociate learners from a project.

In the next section, we present the role of the learner in GOALS.

4.2.3 Learner

Recall that, in GOALS each learner profile can be associated with one or more projects. A learner can use GOALS for two reasons, firstly, to interact with a pedagogical scenario of a project, which the platform generates according to some

pedagogical goals defined by the learner (in some application domain, the goals are defined by the expert). Secondly, to visualize the profile i.e. to show him their current masteries regarding a particular pedagogical domain.

After the identification of different types of users, we present the different interfaces implemented in GOALS for different functionalities. In the next section, we present the interfaces for an expert.

4.3 Configuration of GOALS by the expert

The figure 4.2 shows the login interface of GOALS. This interface allows an unified interface for all types of users i.e. administrator, expert, and learners. GOALS supports internationalization with the use of English and French.

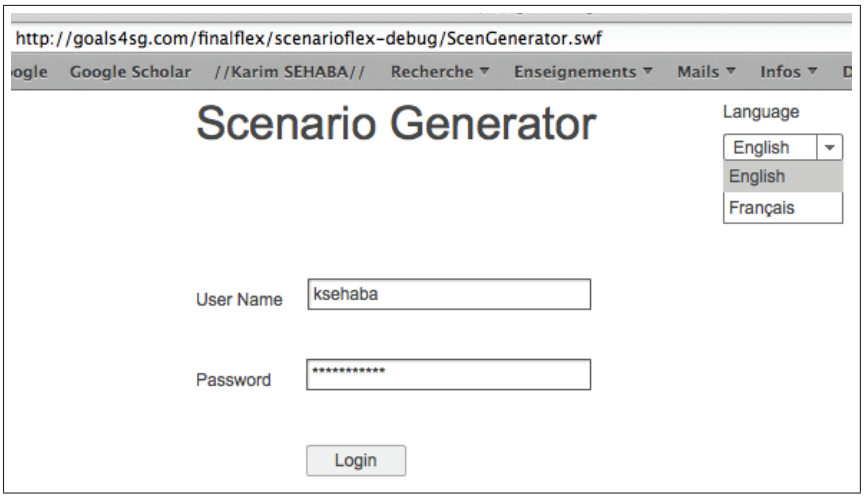


Figure 4.2: Login interface

This figure 4.3 shows the interface, which allows the expert to manage **Projects** and **Learners**. In the next section, we present the process of managing projects.

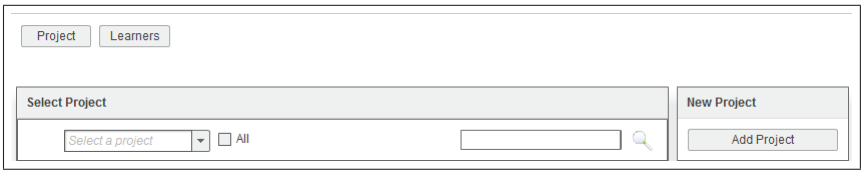


Figure 4.3: Management Interface for an Expert

4.3.1 Projects Management

As we have mentioned before, it is possible for an expert to manage multiple projects. A project contains information about a pedagogical domain. This information in-

cludes the domain concept knowledge, the pedagogical resource knowledge, the game resource knowledge, the presentation model, the adaptation knowledge, the learners associated with the project, and the possibility to generate scenarios for learners. An expert can create, modify, and delete multiple projects.

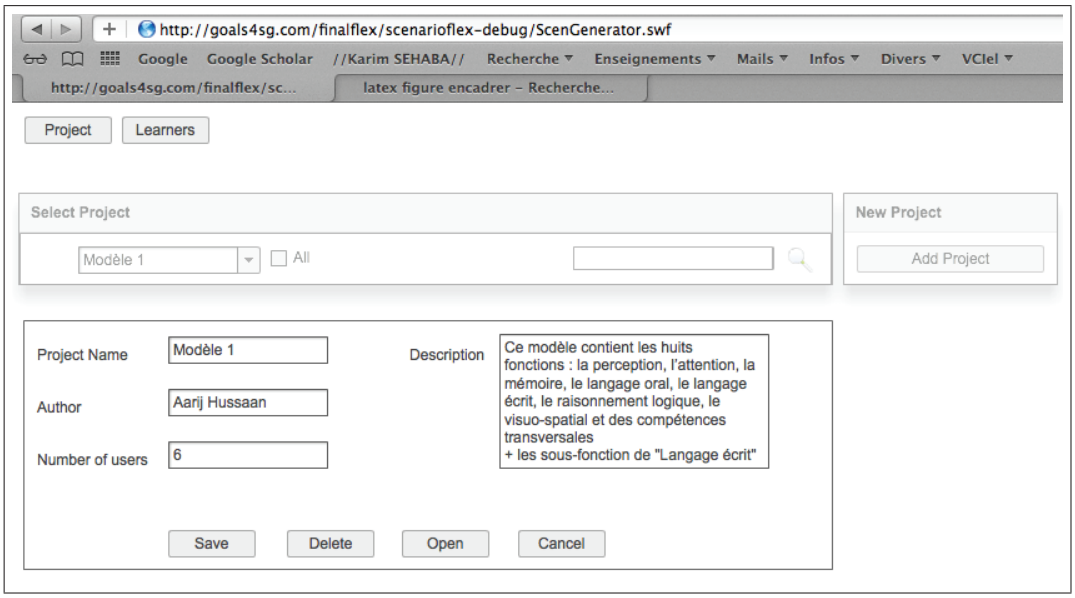


Figure 4.4: Project management interface

As can be seen in the figure 4.4, to create a new project, the user has to click the **Add Project** button. Clicking this button will show the interface shown in the middle of the figure. Here, the expert can define a project's name. The *Author* field contains the name of the expert. The *Number of users* signifies the number of learners currently associated with the project, this number is zero in case of a new project. It is also possible to enter the text describing the project in the *Description* field. At this point, two options are available, either to create this project by clicking the **Save** button, or to cancel the project creation process by using the **Cancel** button.

Existing projects can also be modified by selecting a project from the combo box in the **Select Project** tab. Selecting a project will present the expert with the option to either modify the project (similarly to that of project creation) or **Open** the project for further knowledge entry regarding the project.

Similarly to project, in GOALS, the learner profiles can also be managed. The next section describes this management.

4.3.2 Learners Management

Figure 4.5 shows the Learner Management interface. The learners created here are independent of the projects. This means that an expert has to associate them to

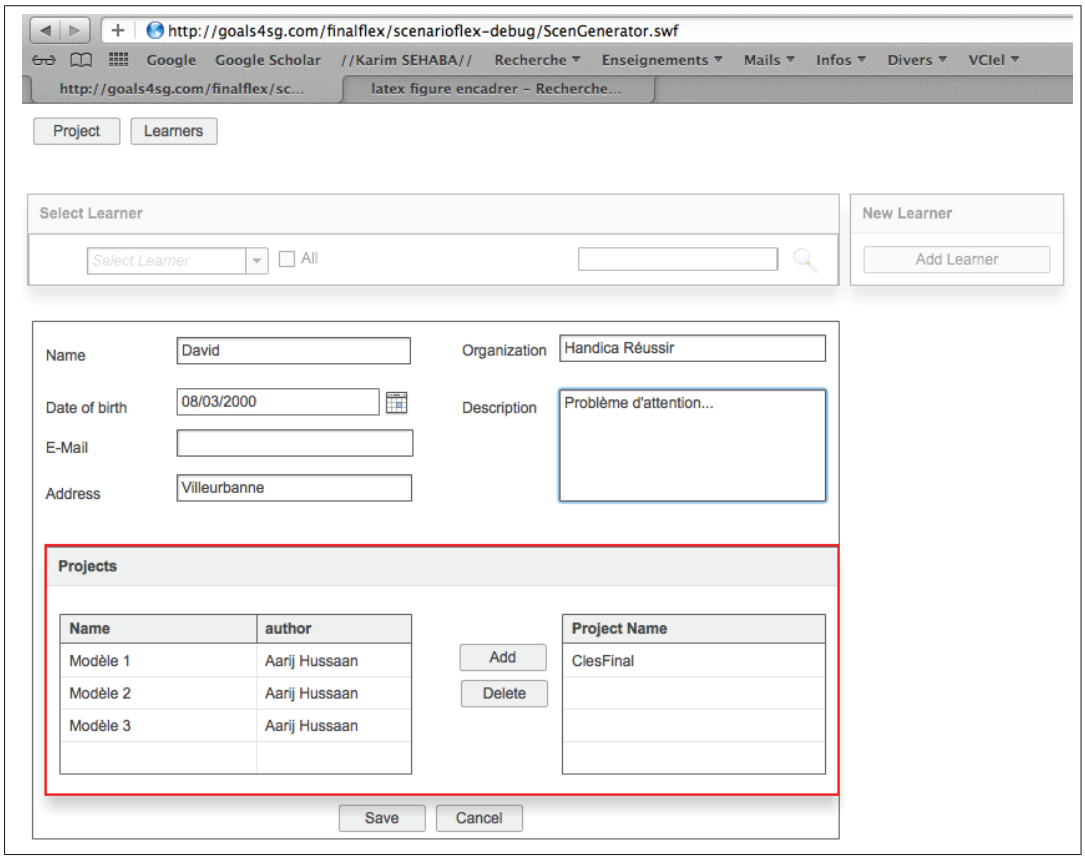


Figure 4.5: Learner management interface

projects. To create a learner profile the expert has to click the **Add Learner** button. This opens the interface, which allows the expert to enter information about the learner. This information includes learner's *Name, Date of birth, E-mail, Address, Organization* and any *Description* of the learner. The expert enters this information in the respective fields of the interface. Afterwards, the expert can either click the button **Save** to create a learner or click the **Cancel** button to cancel the learner creating process.

A learner could also be associated with more than one project. The interface allows the association of learners to projects, as shown by the red bounding box in the figure 4.5. For this, the expert selects the projects from the grid on the left and then click **Add** to create the association. Similarly, if the expert wants to dissociate a learner from a project, he selects a project from the grid on the right and click **Delete**.

The next section describes the interfaces for knowledge creation.

4.3.3 Knowledge Editor

In this section, we describe the interfaces of the GOALS platform to manage the knowledge relating to the three layers of knowledge elements. These three layers contain the domain concept knowledge, the pedagogical resource knowledge and the serious game resource knowledge. The expert starts this process by selecting a project, and opening it. The figure 4.6 shows the interface of a control panel, which the expert uses to create these different types of knowledge.

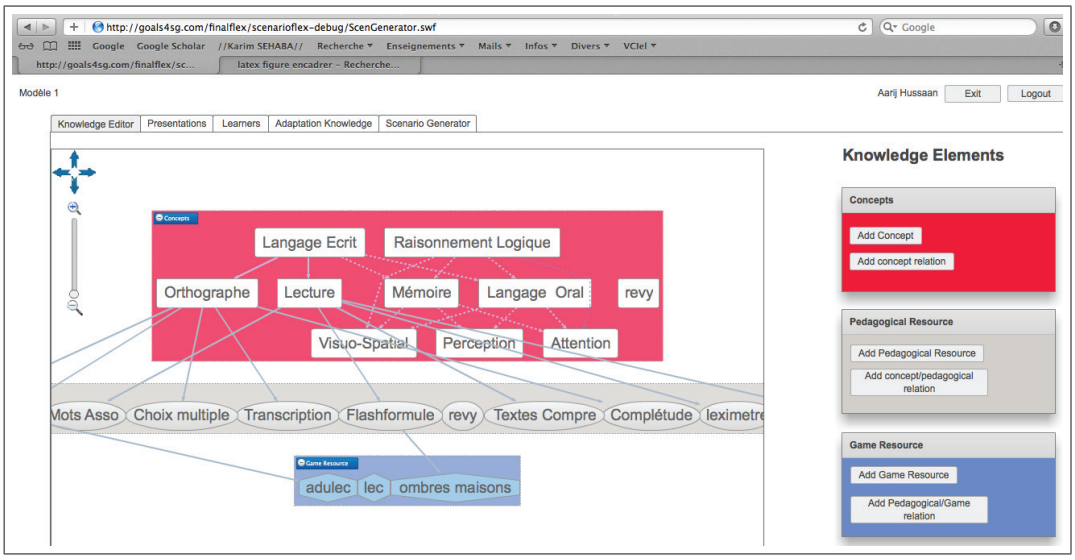


Figure 4.6: Expert's control panel

The first part of this interface is the **Knowledge Editor**, this is where the expert visually creates the knowledge. In this interface, we can observe the organization of the knowledge elements in a graph-like manner. On the left side of this interface, we have the knowledge entry area. This is where the knowledge can be visualized, created and modified in a graph-like manner. The rose-coloured layer contains the domain concepts and the relations between them. The grey-coloured layer contains the pedagogical resources. The blue-colouered layer contains the serious game resources. Different kinds of arrows represent the different types of relation between these layers. We refer to the visual elements representing the domain concepts, pedagogical resources, and game resources as **Nodes**. We use the term **Links** to refer to the visual elements describing different kinds of relations. We have used a Flash-Based visualization library "Kalileo Diagrammer" for the visual elements.

In the platform GOALS, the elements can be interacted by using a mouse. The elements can be moved around, placed anywhere in the designated area, and modified by the expert. The expert can double click on any of the elements of the graph, including nodes and links, to open a contextual menu. This contextual menu allows the option to either open the interface for the modification of the element or to delete the element.

On the right hand side of the interface shown in the figure 4.6, in the tab **Knowledge Elements**, we have buttons that can be used to create different types of elements. These elements include: domain concepts, relation between these concepts, pedagogical resources, relation between a concept and a pedagogical resource, game resources, and the relation between a pedagogical resource and a game resource.

In the next section, we describe the interfaces related to the creation and modification of a domain concept, pedagogical resources, serious game resources and the relations between them.

4.3.3.1 Domain Concept Knowledge

The interface for both the creation and modification of a domain concept is the same. The only difference being that when creating a concept all fields of the interface are blank, and while modifying a concept, these fields have the data of that concept. The expert can modify a concept by double-clicking the domain concept node, and then selecting the **Edit** option from the contextual menu. The expert can also create a concept by clicking the **Add Concept** button on the right hand side of the figure 4.6.

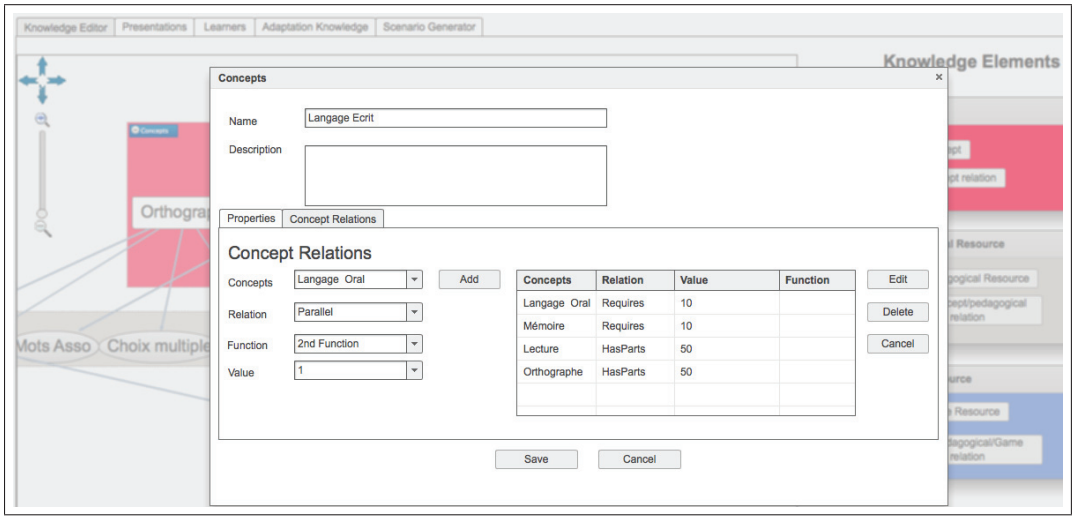


Figure 4.7: Concept pop-up screen

The figure 4.7 shows the interface for creating or modifying a domain concept. Through this interface, it is possible to define (or to modify) the concept's name, the description, the properties of a concept and the relations between this one and other concepts. To add a relation, in the **Concept Relation** tab, the expert has to select the target concept i.e. the concept with which the relation has to be made, afterwards, he selects the relation (recall that, we have many types of relations: Has-Parts, Required, Type-Of and Parallel). Then either the expert selects a function associated with this relation, this function calculates the impact of the source

concept on the target concept, or he assigns a value to this relation. To store the relation, he clicks the **Add** button. This process can be repeated to add as many relations as the expert requires. The added relations can also be modified by selecting a relation from the grid on the right, and clicking the **Edit** button. Then the expert can change the target concepts, relation type, function and the value. He can then re-click the **Edit** button to save the changes or the **Cancel** button on the right of the grid to discard the changes. Similarly, an existing relation could be deleted.

The figure 4.7 shows the interface for modifying a concept "Langage Ecrit". This concept is in a relation "Required" with two other concepts "Langage Orale" and "Mémoire". Langage Ecrit is also in relation of type "Has-Parts" with two concepts "Lecture" and "Othographe".

The figure 4.8[1] shows the interface to modify the properties. We can add a property by defining a name via the *Name* and a description via the *Description* field, and then clicking the **Add** button to add property to the concept. We can edit a property by selecting it from the grid on the right hand side, and clicking the **Edit** button. After modifying we can re-click the **Edit** button to register the changes or click the **Cancel** button, on the right of the grid, to discard the modifications. Similarly, to delete a property, we select a property from the grid on the right hand side and click delete.

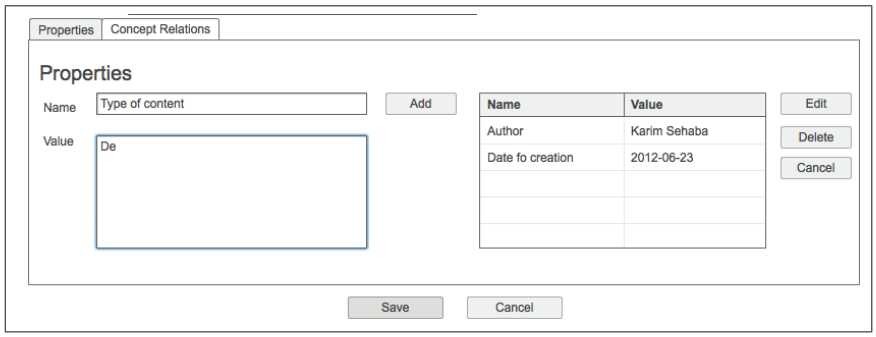


Figure 4.8: Property edit interface

After the modification of the concept's properties and relations, the expert can either save or discard the changes. To save the changes, he clicks the **Save** button. To discard the changes, he clicks the **Cancel** button. These buttons are at the bottom of the interface.

Similarly, as of domain concepts, the pedagogical resources can also be created or modified as shown in the next section.

[1] We have shown the interface to modify properties separately because it is the same interface and methods for the concept, pedagogical resource and game resource modification/creation interfaces.

4.3.3.2 Pedagogical Resource

Similar to the interface of the domain concepts, the interface for both the creation or modification of a pedagogical resource is the same. In order to modify a pedagogical resource, the expert needs to double-click on the pedagogical resource node and select the **Edit** option from the contextual menu. In order to create a pedagogical resource, he needs to click on the **Add Pedagogical Resource** button shown on the right hand side of the figure 4.6. The figure 4.9 shows the interface for the creation/modification of pedagogical resources. Through this interface, the expert can define a pedagogical resource's name, description and type (Recall that, there are many types for a pedagogical resource). The administrator pre-defines the list of pedagogical resource types. The properties of a pedagogical resource can also be modified. Moreover, It is also possible that a pedagogical resource contains some external resources like mini-games or documents in different forms (pdf, html, swf, etc). These files can be presented to the learner. An expert can use this interface to upload these files and associate them with the pedagogical resource. This can be done via the **Upload File** button.

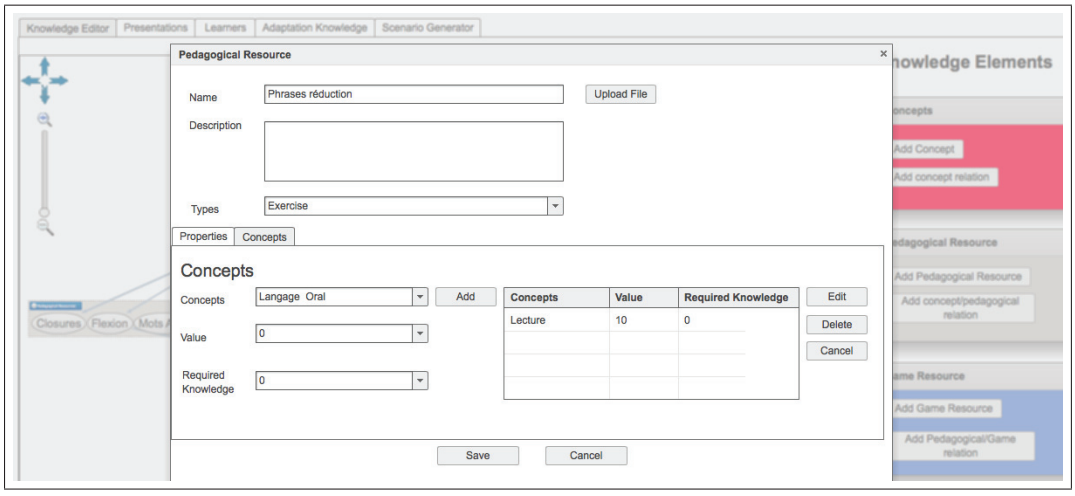


Figure 4.9: Pedagogical pop-up interface

Furthermore, the relations between the pedagogical resource and domain concept(s) can also be created or modified. To add a relation, in the **Concepts** tab, the expert has to select the *concept* from the drop-down list, on the left. Then he assigns a *value* to this relation. He also defines the *required knowledge* for this resource and the concept, this required knowledge defines the concept's mastery the learner needs to access this resource. After selecting all the elements, he clicks the **Add** button to create the relation. In the GOALS platform, it is also possible to add as many relations with as many concepts as is necessary. Furthermore, a relation can be modified by selecting it, from the grid on the right and click the **Edit** button. Then the expert can change the concepts, value, and the required

knowledge, after editing, he can either re-click the **Edit** button to save the changes or click the **Cancel** button on the right of the grid to discard the changes. Similarly, an existing relation could be deleted.

The process of creating and modifying properties is similar to that of the domain concepts.

After the modification of the pedagogical resource's properties and relations, the expert can either save or discard the changes. To save the changes, he clicks the **Save** button. To discard the changes, he clicks the **Cancel** button. These buttons are at the bottom of the interface.

The next section describes the creation or modification of the serious game resources.

4.3.3.3 Serious Game Resources

The interface for the creation and modification of a serious game resource is the same. In order to modify a serious game resource, the expert needs to double-click on the serious game resource node and select the **Edit** option from the contextual menu. In order to create a serious game resource, he needs to click on the **Add Game Resource** button shown on the right hand side of the figure 4.6. The figure 4.10 shows the interface for the creation/modification of serious game resources. Through this interface, the expert can define a serious game resource's name, description and type. The administrator pre-defines the list of serious game resource types. The properties of a serious game can also be modified.

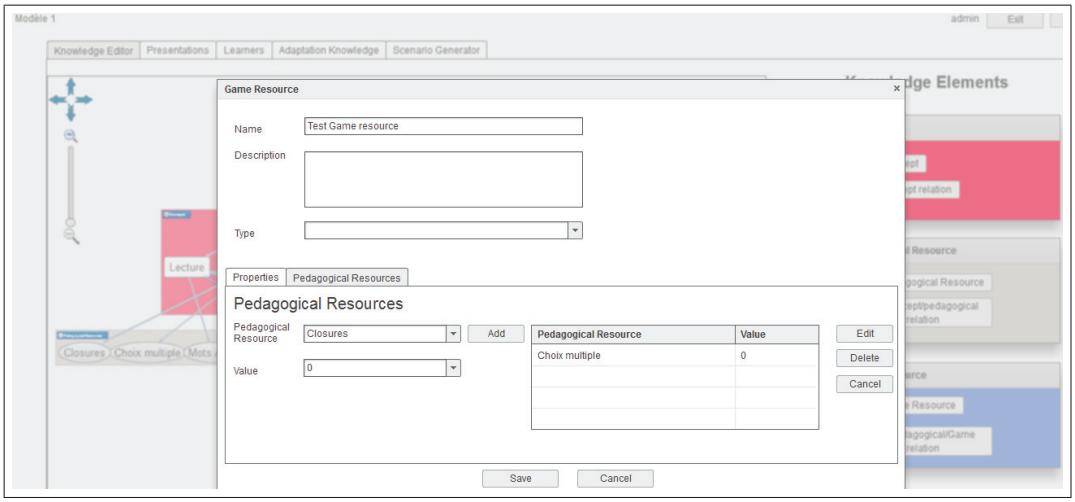


Figure 4.10: Game pop-up interface

Furthermore, the relations between the serious game resource and pedagogical resource can also be created or modified. To add a relation, in the **Pedagogical Resource** tab, the expert has to select the *pedagogical resource* from the drop-down list, on the left. Then he assigns a *value* to this relation. After selecting

all the elements, he clicks the **Add** button to create the relation. In the GOALS platform, it is also possible to add as many relations with as many pedagogical resources as is necessary. Furthermore, a relation can be modified by selecting it, from the grid on the right and click the **Edit** button. Then the expert can change the pedagogical resource, and the value, after editing, he can either re-click the **Edit** button to save the changes or click the **Cancel** button on the right of the grid to discard the changes. Similarly, an existing relation could be deleted.

The process of creating and modifying properties is similar to that of the domain concepts.

After the modification of the serious game resource's properties and relations, the expert can either save or discard the changes. To save the changes, he clicks the **Save** button. To discard the changes, he clicks the **Cancel** button. These buttons are at the bottom of the interface.

Moreover, individual relations between different knowledge elements can be created/edited/deleted via separate interfaces. The next section presents these interfaces.

4.3.3.4 Relations

The figure 4.11 shows the interface for the creation of relations between concepts. The expert can access this interface by either clicking on the **Add concept relation** from the **Knowledge Elements** tab of the interface shown in figure 4.6, or by double clicking on the arrow between two concepts from the knowledge editor area, and then selecting the **Edit** option from the contextual menu.

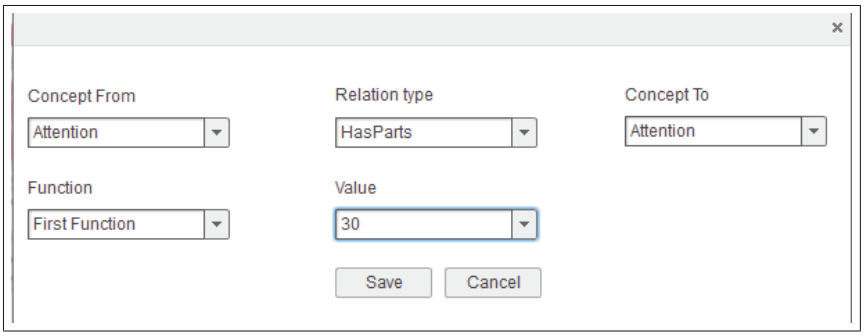


Figure 4.11: Concept Relation pop-up interface

Through this interface, it is possible to modify an existing relation or create a new one. For this, it is necessary to choose the source concept (*Concept From*), the target concept (*Concept To*), the relation type (*Relation Type*), the function (*Function*) and the value (*Value*). Once the definition is complete, these changes can be saved by clicking the **Save** button or they can be discarded by clicking the **Cancel** button.

The interface for the creation of relation between a concept and a pedagogical

resource is similar to that shown in the figure 4.11. This interface can be accessed by either clicking on the **Add concept/pedagogical relation** from the Knowledge Elements tab of the interface shown in 4.6, or by double clicking on the arrow between a concept and a pedagogical resource, from the knowledge editor area, and then selecting the **Edit** option from the contextual menu. This interface allows either modifying an existing relation or creating a new one.

The interface, for the creation of relation between a serious game resource and a pedagogical resource is also similar to the interface shown in the figure 4.11. This interface can be accessed by either clicking on the **Add pedagogical/game relation** from the Knowledge Elements tab of the interface shown in 4.6, or by double clicking on the arrow between a game and a pedagogical resource, from the knowledge editor area, and then selecting the **Edit** option from the contextual menu. This interface allows either modifying an existing relation or creating a new one.

Apart from the interfaces for manipulating the pedagogical and serious game domain, it is also possible to define the models that are necessary to generate the pedagogical scenarios. The next section describes the interfaces for defining a Presentation Model.

4.3.4 Presentation Model

This model, as mentioned earlier (section 3.2.6), is a list of different pedagogical resources type and is used to structure the pedagogical scenario. Figure 4.12 shows the interface to create the presentation model.

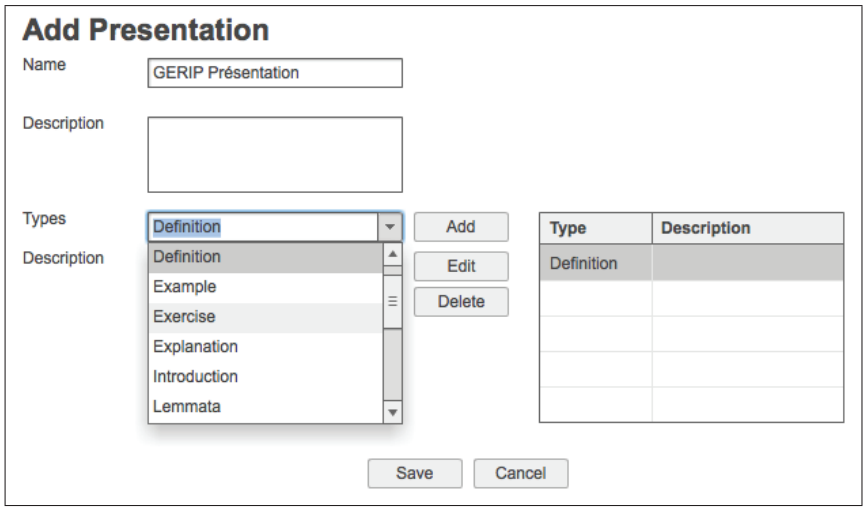


Figure 4.12: Presentation model interface

This figure shows an example of a presentation model. The expert needs to click the **Add** button, from the upper right hand side of the interface, to create a presentation model. Clicking this button opens the **Add Presentation** tab of the interface. Through this tab, the presentation's name and description can be

defined. The expert can define the model by selecting a type from the *Type* drop down list, writeing a description (optional), and clicking the **Add** button in the Add Presentation section. The expert can also modify an existing type, by selecting a type from the grid and then clicking the **Edit** button. He can delete an existing type by selecting the type, from the grid, and clicking the **Delete** button.

The expert can save the presentation model by clicking the **Save** button, or he can discard the changes by clicking the **Cancel** button. Both these buttons are at the bottom of the interface.

The GOALS platform also allows the modification of the learners' profiles associated with a project. The next section describes the learner's profile.

4.3.5 Learner Profile

In the **Learner** tab of the expert's control panel of the interface shown in figure 4.6, the learner's profile, associated with the currently worked on Project, can be modified or created. The figure 4.13 shows the interface for learner profile's creation and modification. This interface allows to see the list of all the learners currently associated with the project, in the **Select Learner** tab of the interface. The expert can select learners to modify their profiles or create profiles by clicking the **Add Learner** button. The interface, for both the creation and modification of a profile, is the same. While creating a learner's profile, all the fields in the interface are empty. While modifying a learner's profile, the fields contain information about the learner. The expert can enter two kinds of information, regarding a learner, through this interface: a learner's personal information and his domain concept masteries of the selected project.

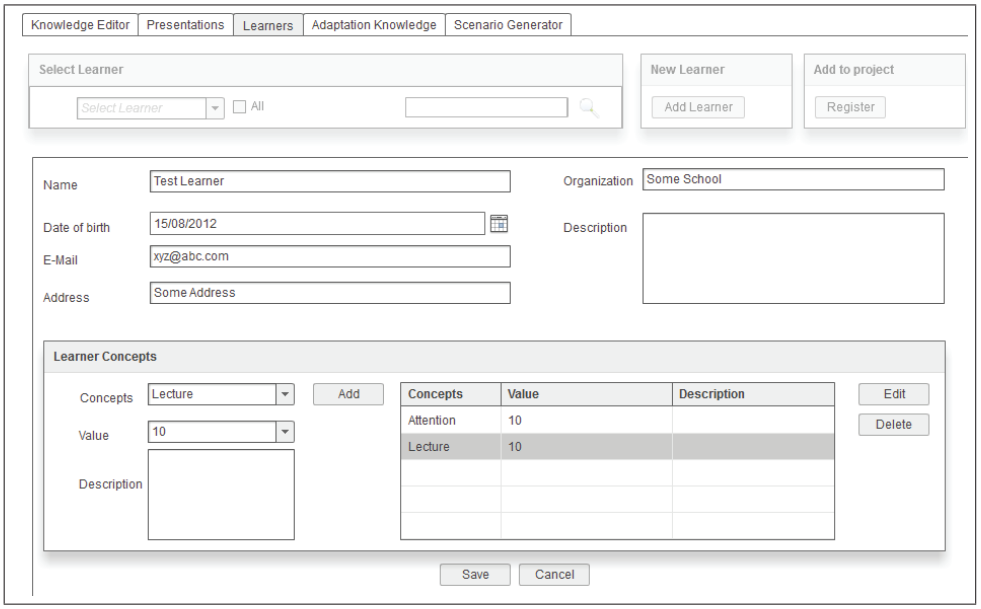


Figure 4.13: Learner profile management interface

The process of entering the personal information is similar to the process described in the section 4.3.2. In order to add the domain concept masteries in the learner profile, the expert selects the concept from the drop down list, *Concepts*, in the **Learner Concepts** section of the interface. Then he selects the value he wants to give the learner regarding the selected concepts. He can also add comments describing his decision to assign the value or anything else. Then he can click the **Add** button in the Learner Concepts section of the interface. This process can be repeated to add as many concepts in the profile as he wants.

In order to modify a concept value in the learner profile, he selects a concept from the grid, and clicks the **Edit** button. Then he modifies the concept, values, or descriptions, and then re-click the **Edit** button to save the changes. Similarly, a concept can be deleted from the learner profile by selecting the concept from the grid and clicking the **Delete** button.

The expert can save the changes by clicking the **Save** button, or he can discard the changes by clicking the **Cancel** button.

After the definition, of all the necessary models, the scenarios can be generated for different learners with different pedagogical objectives. The generated scenarios can be used to verify the models. The next section describes the interface for the generation of scenarios.

4.3.6 Scenario Generator

The figure 4.14 shows the interface to generate the scenarios. Through this interface, an expert can select the learner, for whom the scenario is to be generated, the presentation model and the pedagogical objectives. Then he can generate the scenarios and launch the mini-games, selected by the generator, via this interface. This interface helps the expert to verify the quality of the generated scenarios, and whether the scenarios are adequate for the selected learner profile.

In order to perform the scenario generation, the expert first selects the learner from the *Learners* drop-down list and the presentation model from the **Learner** section of the interface. Then he defines the pedagogical objectives. This can be done in the **Pedagogical Objective** section of the interface. He selects a concept and the concept's competence, which the learner has to achieve and clicks the **Add** button. This process can be repeated to add as many objectives as he wants. After the definition of objectives, he clicks the **Generate** button to generate the scenario.

The scenario is presented in two forms: graphical and textual. In the graphical form, the expert can see the entire domain knowledge graph with the selected concepts, pedagogical resources and the game resources in different colours. For example, in the figure 4.14, the selected concepts have a dark violet background and the selected pedagogical resources have a light violet background.

The textual version of the scenario describes in which order the concepts should be studied and which resources the learner should study.

GOALS can also be used by a learner as shown in the next section.

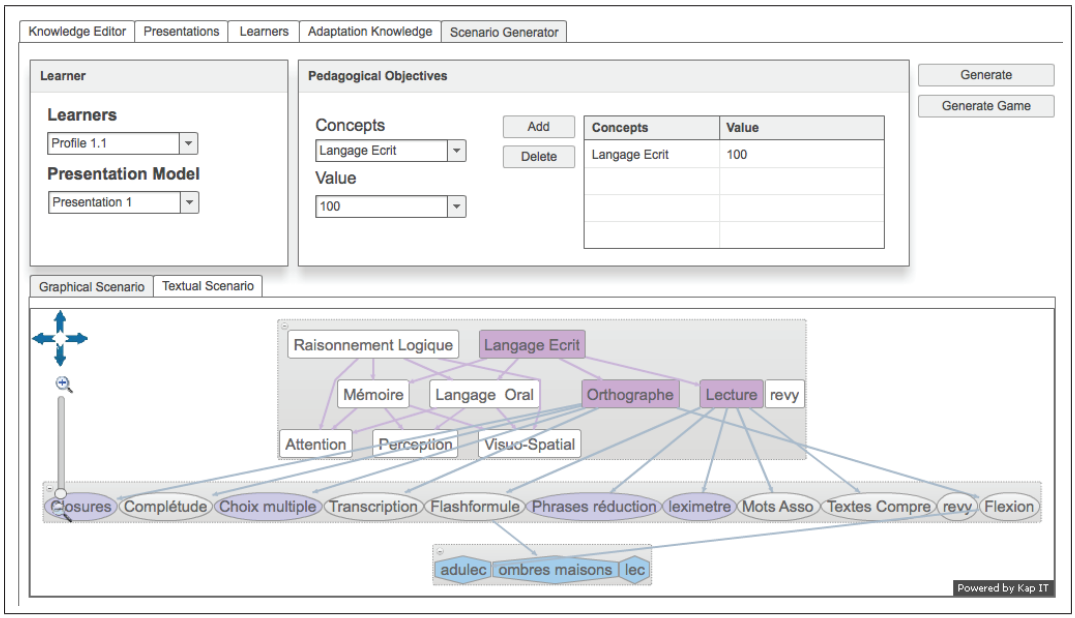


Figure 4.14: Scenario generation interface

4.4 Scenario Generation in GOALS by the learner

A learner can use GOALS to either interact with the pedagogical scenarios, or visualize his profile. A learner can interact with the pedagogical scenarios of the projects associated with him. A learner can be associated with more than one project. In this case, when a learner logs into GOALS, he can select a project, which he wants like shown in the figure 4.15.

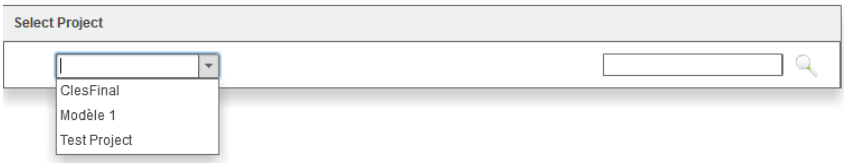


Figure 4.15: Learner associated with multiple projects

Once a learner has selected a project, he can see the interface as shown in figure 4.16. This interface is same as that in the figure 4.14 except for the fact in the interface for the learner there is no choice for selecting other learners. The process of generating the scenario is the same as described in the section 4.3.6.

The learner can also visualize his profile by clicking the **Profile** tab. The interface to show the learner with his profile is similar to that presented in figure 4.13.

In the next section, we present an example of modelling a pedagogical domain

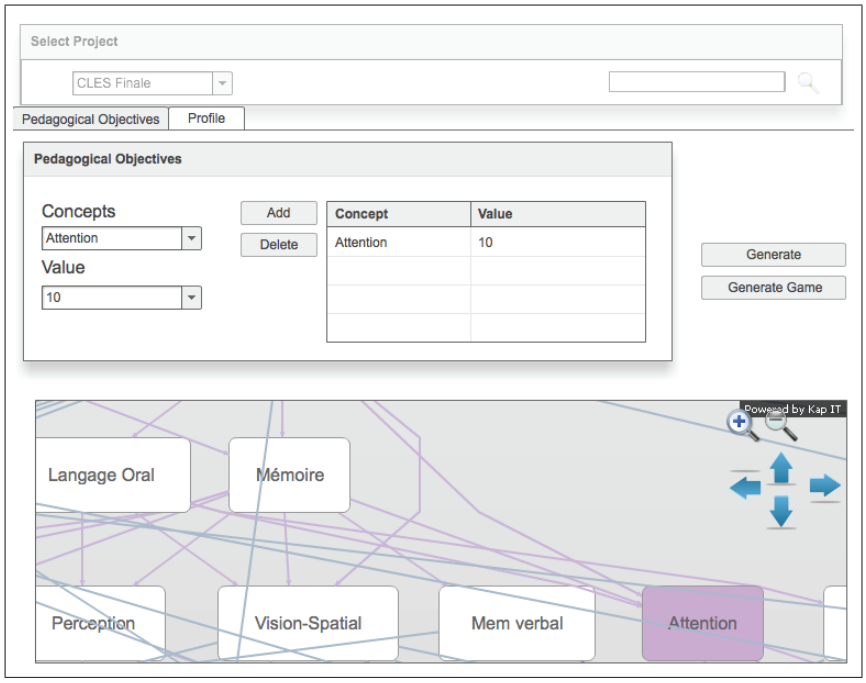


Figure 4.16: Scenario generation interface for the Learner

with GOALS, creating learners and their profiles and then generating a scenario.

4.5 Illustrative Example

To illustrate the knowledge modelling process in GOALS, we present the modelling of a simple model of mathematics. This model contains is based on the example presented in the section 1.2 of the chapter 1. The main concept of this model is **Maths**, its sub-concepts are **Addition**, **Subtraction**, **Multiplication** and **Division**. The sub-concepts are in relation with Maths with the relation **Has-Parts**. The concept Addition is a pre-requisite of the concept Multiplication. Similarly, Addition, Subtraction and Multiplication are all pre-requisites of the concept Division.

The concept Addition has two sub-concepts **Simple Addition** and **Fraction addition**, they are in relation of type Has-Parts with Addition. The same goes for the concept Subtraction, which also has two sub-concepts **Simple** and **Complex Subtraction**, and Multiplication, which also has two sub-concepts **Simple** and **Complex Multiplication**.

Each of the sub-concepts has a relation with one or more pedagogical resources. For this example, we have only included the pedagogical resources of the type 'exercise'. The figure 4.17 shows the resulting domain model. Different types of arrows between concepts represents different kinds of relations.

To organize the pedagogical scenario, we have created a presentation model containing only one pedagogical resource of type 'exercise'. We have also created

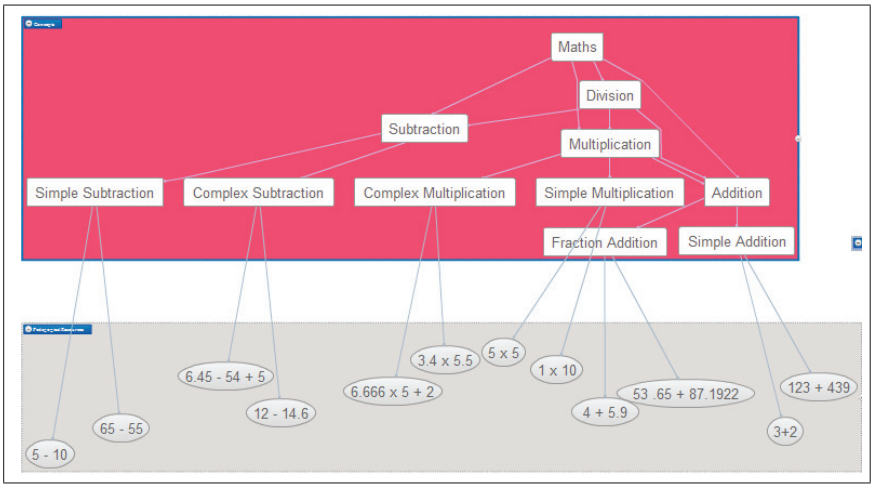


Figure 4.17: The domain model of Maths

some example learner profiles. These profiles have different characteristics, so that for the same pedagogical objectives, different scenarios could be generated. The profiles are the following:

Profile 1 : A novice, knows nothing about the mathematical domain. The *Competences* in his profile are: <ID$_{\text{Maths}}$,0>, <ID$_{\text{Addition}}$,0>, <ID$_{\text{Subtraction}}$,0>, <ID$_{\text{Division}}$,0>, and <ID$_{\text{Multiplication}}$,0>.

Profile 2 : A learner of intermediate knowledge about the domain , knows basic addition and subtraction. The *Competences* in his profile are: <ID$_{\text{Maths:}}$,30>, <ID$_{\text{Addition}}$,40>, <ID$_{\text{Simple Addition}}$,100>, <ID$_{\text{Fraction Addition}}$,10>, <ID$_{\text{Subtraction}}$,40>, <ID$_{\text{Simple Subtraction}}$,100>, <ID$_{\text{Complex Subtraction}}$,10>, <ID$_{\text{Division}}$,0>, and <ID$_{\text{Multiplication}}$,0>.

Profile 3 : An advanced learner of maths, knows pretty much all about Maths except division and a bit of multiplication. The *Competences* in his profile are: <ID$_{\text{Maths:}}$,60>, <ID$_{\text{Addition}}$,100>, <ID$_{\text{Subtraction}}$,100>, <ID$_{\text{Division}}$,50>, <ID$_{\text{Multiplication}}$,60>, <ID$_{\text{Simple Multiplication}}$,90>, and <ID$_{\text{Complex Multiplication}}$,30>

Now, we will generate the pedagogical scenarios for the three profiles with the same pedagogical objective of Maths with a target value of 100. The results of the three generated scenarios for the three profiles (Profile 1, Profile 2 and Profile 3) can be seen in the figure 4.18, 4.19 and 4.20, respectively.

As can be observed, in figure 4.18, the selected concepts (concepts in dark violet colour), and for each selected concept a single selected pedagogical resource of the type 'exercise', wherever a resource is present. For the profile 1, the generator has selected all the concepts because profile 1 masters nothing about the domain.

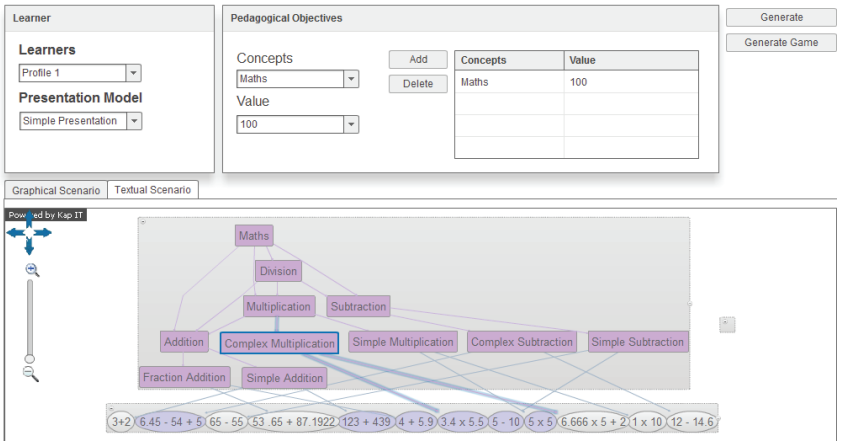


Figure 4.18: The scenario generated for Profile 1

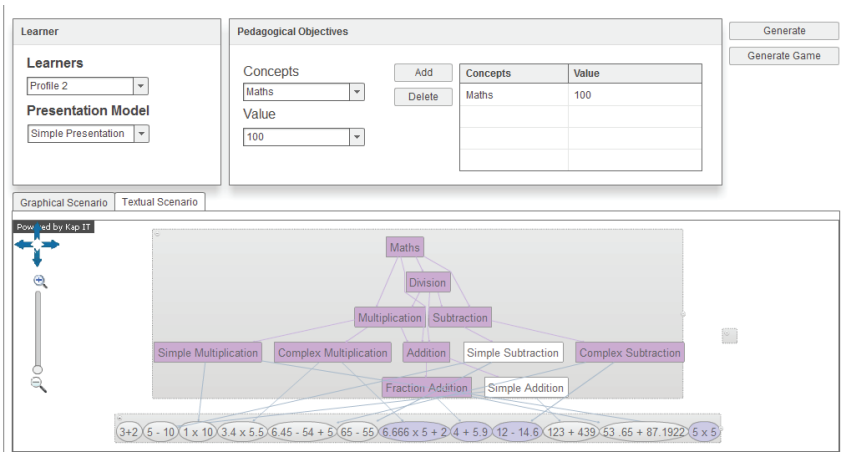


Figure 4.19: The scenario generated for Profile 2

The scenario in figure 4.19 selects all the concepts except Simple Addition and Multiplication because, the profile 2 sufficiently masters the two concepts.

Profile 3 masters much about maths, masters sufficiently Addition and Subtraction. Profile 3 also masters multiplication but not sufficiently, hence, the scenario, shown in figure 4.20, shows Multiplication as selected but with a lower value. Furthermore, the generator has not selected the concepts Addition and Subtraction.

In the next section, we present the technical architecture of the platform GOALS.

4.6 Technical Architecture

The figure 4.21 shows the technical architecture of GOALS. This architecture, based one a server/client, contains four principal layers: the presentation layer, the busi-

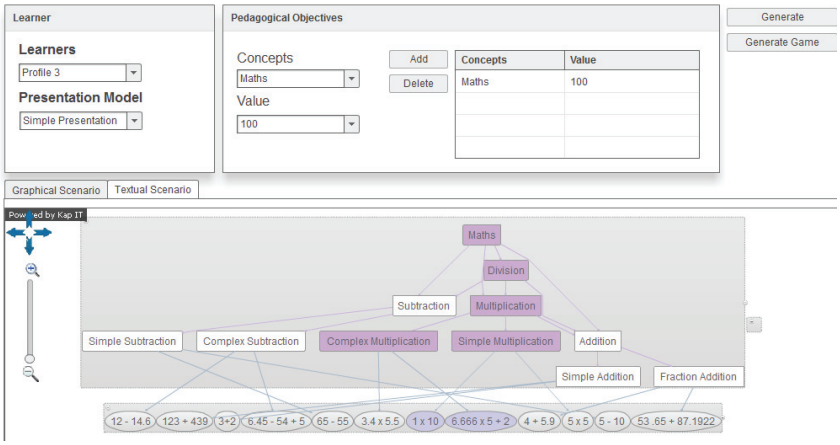


Figure 4.20: The scenario generated for Profile 3

ness layer, the data access layer and the resource layer. We detail each of the layers in the following sections.

4.6.1 Presentation Layer

This layer is responsible for dealing with the user[2] interactions. We have designed the interfaces using Adobe Flex[3] software development kit (SDK) using an IDE called Adobe Flex Builder[4]. Flex allows the creation of Rich Internet Applications (RIA). Much like the creation ofl web-site using HTML[5], HTML helps to design a page and JavaScript[6] makes the web-site dynamic. Similarly, we have used Flex to design a page and ActionScript3[7] (AS3) to handle the dynamics of the page.

For GOALS, we have used Flex as a Service Oriented Architecture (SOA), where we design interfaces with FLEX, and then connect these interfaces to actual data using services. In a Flex application, when a user accesses an application using his browser, the server sends the compiled Flex application (the SWF file) that runs inside the browser using the Flash Player plug-in. Usually, this SWF file holds only the client-side business logic. If the application needs data (in our case, from a database), it makes a request for data. The server then sends only the data, this data can be in many formats, but in our application, we use AMF3 format to map JAVA objects into AS3 objects, and the client knows how to represent this data visually. This figure 4.22 shows this process.

We have implemented the JAVA services using, and these services make use of

[2]Since GOALS can have many different types of user, therefore, the term **user** represents all types of users (learner, domain expert and the administrator

[3]http://www.adobe.com/fr/products/flex.html

[4]http://www.adobe.com/products/flash-builder.html

[5]http://www.w3schools.com/html/default.asp

[6]http://www.w3schools.com/js/default.asp

[7]http://www.adobe.com/devnet/actionscript.html

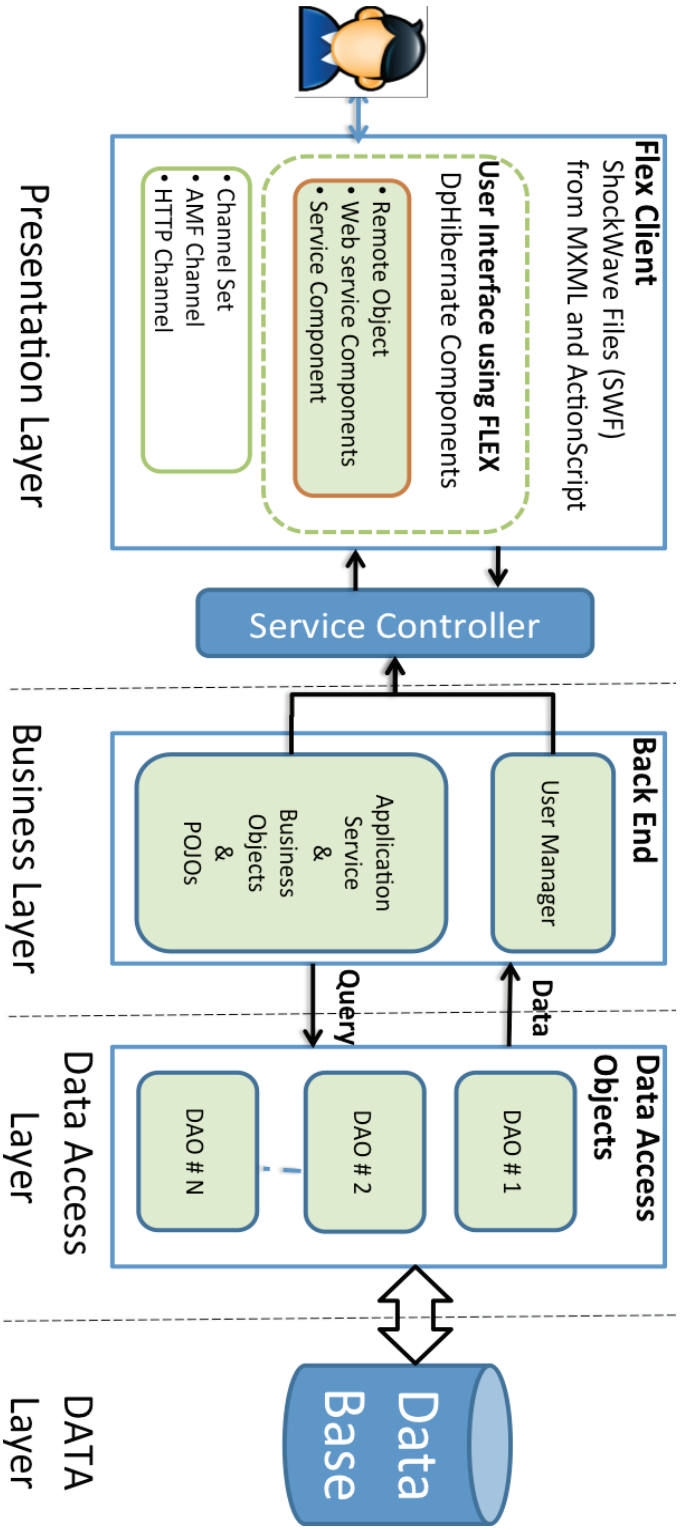


Figure 4.21: System's technical architecture

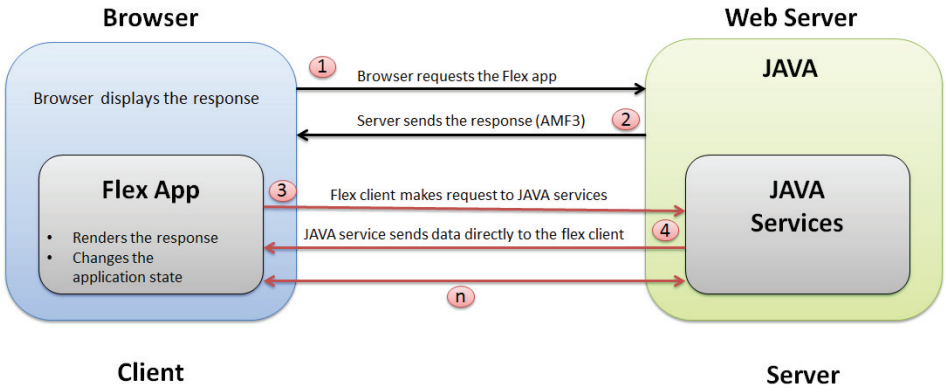


Figure 4.22: Flex application architecture

JAVA Objects for manipulating the data (we describe the JAVA classes in detail in the next section). The actual communication between the Flex application and the JAVA services can be done in many ways: REST services, WEB services, Remoting, and XML-RPC. We have opted for the *Remoting* option for the following reasons: firstly, by using remoting we can call any method exposed (made public) by the JAVA services, i.e. we use the JAVA service as an AS3 class. Secondly, since the data gets managed by using JAVA objects, therefore, by using remoting, we can map the JAVA objects to an AS3 object, and the conversion gets performed automatically. This is extremely helpful when using typed objects. Thirdly, the AMF3 (Action Message Format) for remoting is a binary format, which can be much faster and smaller compared to SOAP/XML/JSON, especially for big sets of data. And as we know, response time is off utmost importance in a web-based application.

There are some options provided by Adobe for remoting purposes like LifeCycle and BlazeDS. We have used BlazeDS, as it is open-source. Being open-source means that BlazeDS[8] lacks in some functionalities, which we need. BlazeDS does not support LazyLoading (described later), which is essential for us to optimize the application's performance. Therefore, we augmented BlazeDS with another open-source framework called *dpHibernate*[9]. dpHibernate is a custom Flex Library and a custom BlazeDS Hibernate adapter that work together to give support for lazy loading of Hibernate objects from inside Flex applications. Using this framework means that we can use the AS3 objects as JAVA objects, request objects only when needed, and persists the data into the database with only the minimum amount of data transferred between the server and the client, thus, minimizing the network traffic and optimizing the user's experience with the application.

GOALS does not only have an interface it also has a business layer, where all the functionalities of GOALS are implemented. This layer is described in the next section.

[8]http://livedocs.adobe.com/blazeds/
[9]http://code.google.com/p/dphibernate/

4.6.2 Business Layer

As shown in figure 4.22, the Flex client interacts with the JAVA services to request data. In the GOALS platform, there are many kinds of services, and each service serves different purposes. Therefore, in order to manage different kinds of services, we have developed a *Service Controller*, this service controller serves as a gateway for all kinds of service requests made from the Flex client, as shown in figure 4.21. The process is as follows, the Flex clients request a service via the service controller, the controller, based on the requested service, invokes the right methods to service the client's request. Afterwards, the methods return the service controller with the requested data, and the controller returns the data to the Flex client.

The business layer uses the JAVA JZEE[10] technologies for implementation. We have used the IDE Eclipse[11] for development. The main purpose of this layer is to service the Flex client and manage all the interaction and modification of the real-data. This data persists in a database, which we have described in the next section. One of the responsibilities of the business layer is to identify different types of users and make sure that only users, with the right credential, can access the data. The *User Manager* is responsible for this. In the GOALS platform, we store data in a relational database, and we use "Object Relational Mapping (ORM)" to access this data in the business layer. ORM is a mechanism that makes it possible to address, access and manipulate objects without having to consider how those objects relate to their data sources. ORM lets programmers maintain a consistent view of objects over time, even as the sources that deliver them, the sinks that receive them and the applications that access them change.

When using ORM, for every table in the database, we have a corresponding JAVA class in the business layer. Based on abstraction, ORM manages the mapping details between a set of objects and underlying relational databases, XML repositories or other data sources and sinks, while simultaneously hiding the often changing details of related interfaces from developers and the code they create.

We have used a highly popular ORM framework called Hibernate[12] in our business layer. Hibernate demands to represent the JAVA classes representing the database tables as Plain Old Java Object (POJO). POJOs are classes, which do not implement infrastructure framework-specific interfaces, and non-invasive frameworks such as Spring, Hibernate, JDO, and EJB 3, which provide services for POJOs. They are useful in decoupling the application code from the infrastructure framework, which helps to change the framework without changing the application code.

In the business layer, we have classes that use the POJOs to perform different operations, as demanded by the Flex client. This include fetching relevant data from the database, performing different operations on the fetched data, and returning the processed data in the proper format to the Flex client.

[10] http://www.java.com
[11] http://www.eclipse.org/
[12] http://www.hibernate.org/

Hibernate also helps in managing Lazy Loading. Lazy Loading is a design pattern to access data, in this pattern, the fetching of the data from the database is delayed until the last moment. This contributes to the overall efficiency of the program by decreasing the amount of data traffic between the application and the database.

Because we are using remoting for communicating between the presentation and the business layer, for each POJO we have a corresponding AS3 class on the Flex Client. The framework dpHibernate works on both the Flex client side and business layer side, and it uses Hibernate to provide Lazy Loading on the client side, as well.

In the next section, we describe how the business layer with the database to perform data-centric operations.

4.6.3 Data Access Layer and Resource Layer

We use a database to persists all the data. The database used is the open-source database management system (DBMS) My-SQL[13]. The data can be accessed, in the application, via POJOs. Hibernate manages this access. However, to separate the business logic from the data-access operations, we have used a Data Access Layer. The purpose of this layer is, for each of the JAVA classes, representing data base table, create a JAVA class to handle all the database-related operation. Consequently, for each of the JAVA classes we have a corresponding Data Access Object (DAO), which contains all the methods for performing database operations.

We have mentioned, that a POJO represents a table in the database, here we present an example of this representation. Figure 4.23 shows an example of database table representing an entity "User". The table has a column "id" to identify the user, a column "name" and column "email". Its corresponding POJO can be seen in the figure 4.24. Notice how each column of the user table corresponds to a data field in the POJO.

Table: USER		
id	name	email
1	abc	abc@abc.com
2	xyz	xyz@xyz.com

Figure 4.23: Table: User

The resource layer contains the actual database and the resources uploaded by the user on the server. This contain potential pedagogical resources as mini-games, pdf and word documents, etc. Figure 4.25 shows the database schema of the GOALS platform. There are twenty two entities in the database.

This application is hosted on-line at http://goals4sg.com/.

[13]http://www.mysql.com/

```
public class User{
        private Integer id ;
        private String name ;
        private String email ;
        User(){
        }
        public Integer getId()
        {
                return id ;
        }
```

Figure 4.24: Partial view of the POJO User

This concludes this chapter. In the next chapter, we will present the application context of our work, which concerns: the Project CLES.

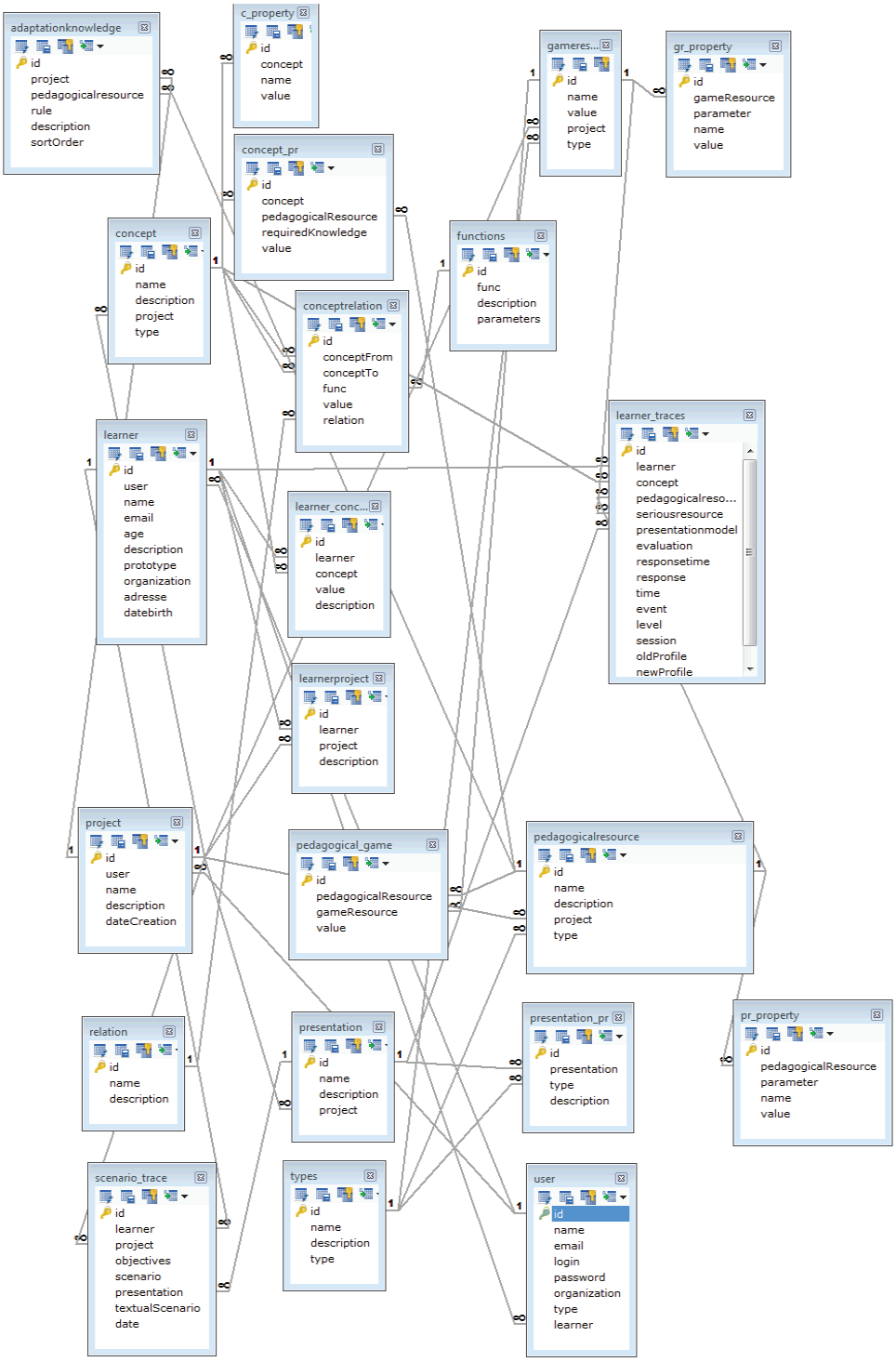


Figure 4.25: System's Database Schema

Chapter 5

Application Context : Project CLES

Contents

5.1 Context and objectives of the CLES Project 104
5.2 Partners 105
5.2.1 GERIP 105
5.2.2 Laboratory EMC 105
5.2.3 Laboratory LUTIN 106
5.2.4 Laboratory LIRIS - SILEX Team 106
5.2.5 Targeted Cognitive Functions 107
5.3 Serious Game: Tom O'Connor 109
5.4 Mini-Games 109
5.4.1 Identify Intermixed Objects (Objets entérmélés à identifier) . 110
5.4.2 Memorize and Recall Objects (Mémoire et rappel d'objets) . 111
5.4.3 Point of View (Point de vue) 112
5.4.4 Complete the Series (Séries logiques à compléter) 113
5.5 CLES Modelling 113
5.5.1 Main Concept Modelling 114
5.5.2 Sub-Concepts Modelling 116
5.6 Using GOALS for CLES 119

This chapter describes the application context of our research. We present in detail the Project CLES and the modelling of CLES's knowledge via our proposed models. Section 5.1 recalls the objectives of this project. Section 5.2 presents the different partners and their role in this project. The section 5.3 presents the serious game Tom O'Connor developed in this project. The section 5.4 presents some mini-games of this serious game. The section 5.5 shows the modelling of CLES's knowledge. Section 5.6 illustrates the uses of GOALS to generate scenarios for CLES.

5.1 Context and objectives of the CLES Project

The assessment and rehabilitation of cognitive disorders has been the subject of several research works. These studies, based on clinical trials, involve different cognitive functions such as working memory [Diamond 1989], attention [Manly 2001], auditory perception [Mody 1997], oral and written language [Broomfield 2004], etc. Along with the advent of computers, researchers have developed digital solutions for linguistic and cognitive re-mediation. The authors in [Botella 2000] use a virtual reality approach for the treatment of people suffering from claustrophobia. LA-GUNTXO [Conde 2009] is a rule-based learning appriach; its purpose is to facilitate the integration of people with cognitive disabilities in work environments. Tutor Informatico [Campos 2004] is an approach designed to help persons with Down syndrome. This approach, based on new mobile technologies, can help people overcome their disabilities and to gain more autonomy. [Sehaba 2005b] proposes an approach, which uses an educational game to help structuring of autistic children. The authors in [Parfitt 1998] propose an environment for distant learning for people with special needs.

These approaches have the advantage of being more flexible and accessible. They can also store the users' traces, which allow clinicians to monitor achievements and the evolution of their patients [Sehaba 2005a]. However, most of these approaches do not adapt to the characteristics and needs of each user. This adaptation is particularly significant since users do not have the same skills, abilities or preferences.

The main objective of the project CLES (Cognitive Linguistic Elements Stimulation) is to develop an adaptive serious game, available online, for cognitive rehabilitation and evaluation. Precisely, this project aims, on the one hand, to create for each cognitive disorder a mini-game, which targets an aspect of the disorder, while optimizing, through techniques of video games, their cognitive ergonomics. On the other hand, it aims to develop a module to generate, for each patient, personalized paths through the game keeping into account the patient's difficulties and progress. This project considers the following cognitive disorders: perception, attention, memory, oral language, written language, logical reasoning, visuo-spatial and transversal skills.

We are particularly interested, in our research, in the generation of learning paths. This means to propose an intelligent approach for the generation of learning scenarios taking into account the learner's profile, therapeutic objectives and interaction traces. In the project CLES, we have 91 mini-games, available on-line, for more than 13 200 learners, and in each mini-game, we have nine levels of difficulty. The role of the generator is to select the adequate mini-games and their level of difficulty according to the learner profile and its learning objectives. The generator should generate a personalized prescription for each patient, keeping into account:

- What the therapist have described his patient.
- A knowledge base of available of possible remedies for the disabilities to be treated.

- The history of the patients past interaction with the system and his past attempts at mini-games.

The project CLES has been conducted with many research teams. Each team has a particular competence. A brief description of each of the partner is presented in the next section.

5.2 Partners

Four research teams participated in CLES: GERIP[1], EMC[2], LUTIN[3], and LIRIS[4]. In the following sections, we provide a brief description of every partner and their contribution in the project.

5.2.1 GERIP

GERIP is a medium level French enterprise for speech therapy, it is co-founded by Mr. Philippe REVY, in 1988. Now based in Lyon, it has a team of five people and develops a catalogue of about 60 software products it sells mainly in France with a market share of 70% (in number of clients).

The activities of GERIP concern with the design of computer-assisted rehabilitation, through cognitive stimulation programs for children, adults, and the elderly and to develop tools for evaluation and re-mediation on the functions concerning language and cognitive skills. GERIP has about 51 softwares for speech education and 9 for literacy. Around 7,000 professional therapists are GERIP's customer.

GERIP participated in this project as coordinator. GERIP brought twenty years of expertise of managing projects for treating cognitive disabilities. Furthermore, GERIP's software package is the starting point for the project CLES. They have a dual competence, concerning this project: first, that of an expert speech therapist, namely skills on cognitive science and language, and the second is that of specialist developers of software for re-education. GERIP has created the knowledge base required for CLES and tested the development of it.

5.2.2 Laboratory EMC

Researchers engaged in research work at EMC "Etude des Mécanismes Cognitifs" Lab are involved in the field of learning to read and the disorders related to reading. Particularly their research focuses on:

1. development of predictors of reading through longitudinal studies conducted with children with normal reading capacities, deaf and in situation of dyslexia,

[1] www.gerip.com/
[2] recherche.univ-lyon2.fr/emc/
[3] www.lutin-userlab.fr/
[4] liris.cnrs.fr/

2. development of assessment tools and

3. the preparation and validation of systems' learning aids manned or unmanned computer (school, hospital, families) with children with dyslexia and poor readers.

In the context of CLES, EMC provided their expertise and knowledge in the areas of language, phonology and memory. Their skills were also utilized in the statistical analysis and interpretation of the results of exercises for patients.

5.2.3 Laboratory LUTIN

The laboratory LUTIN "Laboratoire des Usages en Technologies Numériques" is a platform of CNRS that explores the use of new technologies in the context of academic research and the industrial use of brain-computer interaction in video games.

LUTIN places themselves at the point technology and its usage. It experiments with new methods that take into consideration the end user, while designing and facilitating the transition from Research & Development to the prototyping of innovative products. Finally, LUTIN also works on the creation of synergies between the business and academic research and study brain-computer interaction in video games, in addition to conventional HCI machines.

LUTIN's skills related to the platforms "Game Room" and "Mobility Lab" provided support in the project CLES. Firstly, they studied the ergonomics of CLES, to adapt to the user and target a linguistic function/ cognitive states. Secondly, they performed a study of CLES's gameplay. This was crucial to keep the exercises fun-to-do. Thirdly, tests in the "Mobility Lab" ensured the portability of CLES for mobile applications such as iPhone or similar devices.

5.2.4 Laboratory LIRIS - SILEX Team

The SILEX (Supporting Interaction and Learning by Experience) team of LIRIS lab considers the user/machine couple as a single learning system, co-evolving according to the pursued activity. The observation of that co-evolving system, enabled by the modelling of activity traces, allows to make original proposals in the field of knowledge construction, user assistance, system adaptation to the user, and usage analysis by the user. In the user/machine couple, the *machine* is now to be understood as the networked digital environment of the user, obviously involving the Web in our research.

The research questions of SILEX lead us to design methods, define models and setup tools that we evaluate. Hence we connect theoretical study with applications, in domains as diverse as knowledge management, human learning or user disabilities.

The research questions of SILEX are organized around three topics:

1. Topic 1 - Knowledge dynamics and traced experience

2. Topic 2 - Co-design of situated TELS (Technology Enhanced Learning Systems)

3. Topic 3 - Interactive Adaptive Systems

Our work in the CLES project consists in the development of an adaptive system for generating pedagogical scenarios taking into account the specificity of serious game. Particularly, our research activities focus on the adaptive behaviour of interactive systems. This means to propose systems, having the capability of observing the learner's actions, via different mediums. And responding to the learner by presenting to him different personalized/adapted activities, in real-time, keeping into account the instructions of the domain expert. Our work also contributes in the context of serious games for persons in the situation of a handicap. Therefore, it also means to take into account the different specificities of every individual. In fact, in most of the serious gaming environments, the interactions between learner and the systems are mostly predefined by the game-designer as a function of some pre-conceived scenarios and they do not take into account the history of the learner and the learner's evolution.

The approach developed to fill this gap, consists in personalizing the interaction as a function of the learner's profile, his behaviour and the pedagogical objectives. This means to personalize, for example, the gaming scenario, the gameplay, the elements of the interface, the strategies adopted by the Non-Playing Characters (NPC), etc. The process of adaptation needs some behaviour analysis mechanisms of learners using their interaction traces.

5.2.5 Targeted Cognitive Functions

The cognitive functions that were considered in the project CLES are: Perception, Attention, Visuo-Spatial, Memory, Oral and Written Language, Logical Reasoning, and some transverse competencies. The project CLES targets some aspects of these eight functions. Some description of these functions and the types of exercises present for them is present below:

Perception : This is the interpretation of environmental signals perceived by our senses (sight, touch, hearing, smell ....) [Schacter 2010]. The exercises associated with it test the following sub cognitive functions: visual, body schema and auditory perception. These cognitive functions allow the acquisition and development of *language* and *reading*. These two areas are important in not only adapting to everyday life but also achieving academic and professional success.

Attention : It is the ability to focus on something or someone to gather information, process it and then perform a specific task [Anderson 2004]. Many learning difficulties are related to attentional disorders. The exercises associated with attention test the following sub cognitive functions: auditive attention, visual attention and shared attention. These cognitive functions allows

to respond adequately to many informations that occur at the same time and plays an important part in the development of memory [Astle 2009].

Visuo-Spatial : It is the ability to explore the visual field, to represent space, coordinate eye and hand, imagine the links between the elements of the environment, and to move in time and space. The exercises associated with visuo-spatial test the ability of a person to orientate himself in the environment. This ability is also essential to the acquisition of reading, spelling and logical reasoning and therefore, predictive of academic success.

Memory : It is the ability to absorb, store and reuse information [Schacter 2010]. The exercises associated with memory test the following sub cognitive functions: visual, auditive, verbal and recall memory. These functions corresponds to various brain activities like, sensory memory, short-term memory and long-term memory. The ability to memorize is very important to achieve new learning.

Logical Reasoning : It is the ability to reason from judgements of concrete operations of verbal or non-verbal propositions [Schmeichel 2003]. This is the logic that allows control structures that underlie all mathematical activities. The exercise associated with logical reasoning tests the capacities of a person that are a necessary condition for success in numeracy and mathematics.

Language (Oral and Written): This organized system of sounds or signs is at the heart of communication between people. Its analysis is very complex because it is at the crossroads of several fields (physiological, psychological, social, intellectual, motor, perceptual ...) [5][Schacter 2010]. The exercises associated with oral and written language test the following sub cognitive functions: comprehension, lexicon, phonology, denomination evocation, fluency, reading ability and spelling abilities. These functions help in the understanding of instructions, the naming of elements (name elements that can be seen), vocabulary, and syntax. The cognitive functions related to language are necessary to test and improve as a language disorder slows considerably the learning process.

Transverse Competencies : In addition to individual cognitive functions, it is also necessary for a person to utilize the combination of different sets of competencies, at the same time. The exercises related to this function test the following sub cognitive functions: judging objects, practical approach in solving things, inferring information, planning ability, and speed of processing information. This functions is also important to perform complex tasks in personal and professional lives.

[5] http://en.wikipedia.org/wiki/Linguistics

5.3 Serious Game: Tom O'Connor

The Serious Game of project CLES is an adventure game. The main protagonist of this game is a person named "Tom 'O Connor". Tom is a relic hunter (much like Indiana Jones or Lara Croft). The learner takes control of Tom in this game. Tom aims to search for a relic in a mansion, which contains great mystical powers. On his mission, Tom gets help from two of his colleagues. Their mission is to guide Tom throughout his journey by giving him tips and telling him what to do. In order to search for the relic, Tom finds himself inside one the rooms of the mansion. This room connects to one or many other rooms. Tom needs to find the key in order to exit the room and enter the next one. Each room represents one of the eight cognitive functions (attention, perception, etc). Inside each room, there are objects (chair, desk, screen, etc.). Behind some of these objects, there are hidden challenges in the form of mini-games. Tom has to interact with these objects to launch these mini-games. Tom must launch all the mini-games in the room, in order to access other parts and advance in the game. The gaming environment and some examples of the rooms can be seen in the figure 5.1. The section 5.4 gives some examples of the mini-games.

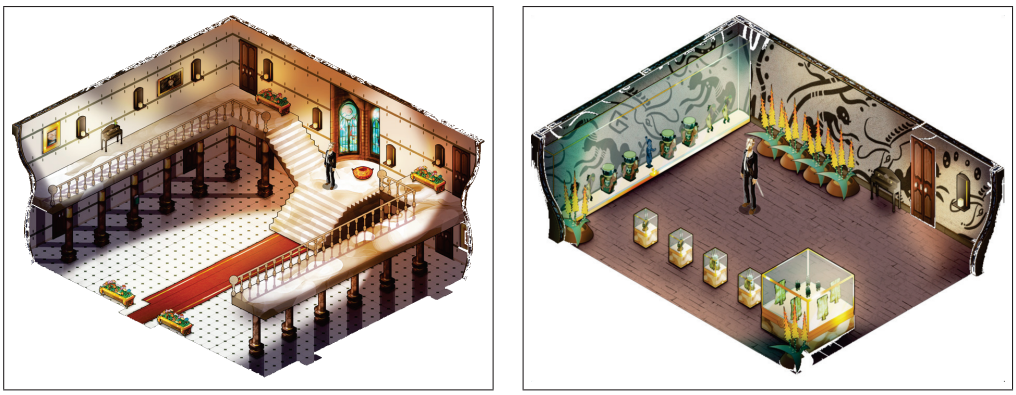


Figure 5.1: Different rooms of the Tom O'Connor game

In the context of the project CLES, we have created, for each of the eight cognitive functions, a "Main Concept" of the domain, and for each of the sub-functions a "Sub-Concept" of the domain. Furthermore, the mini-games are the pedagogical resources and the gaming objects (chairs, tables, etc.) are the serious game resources.

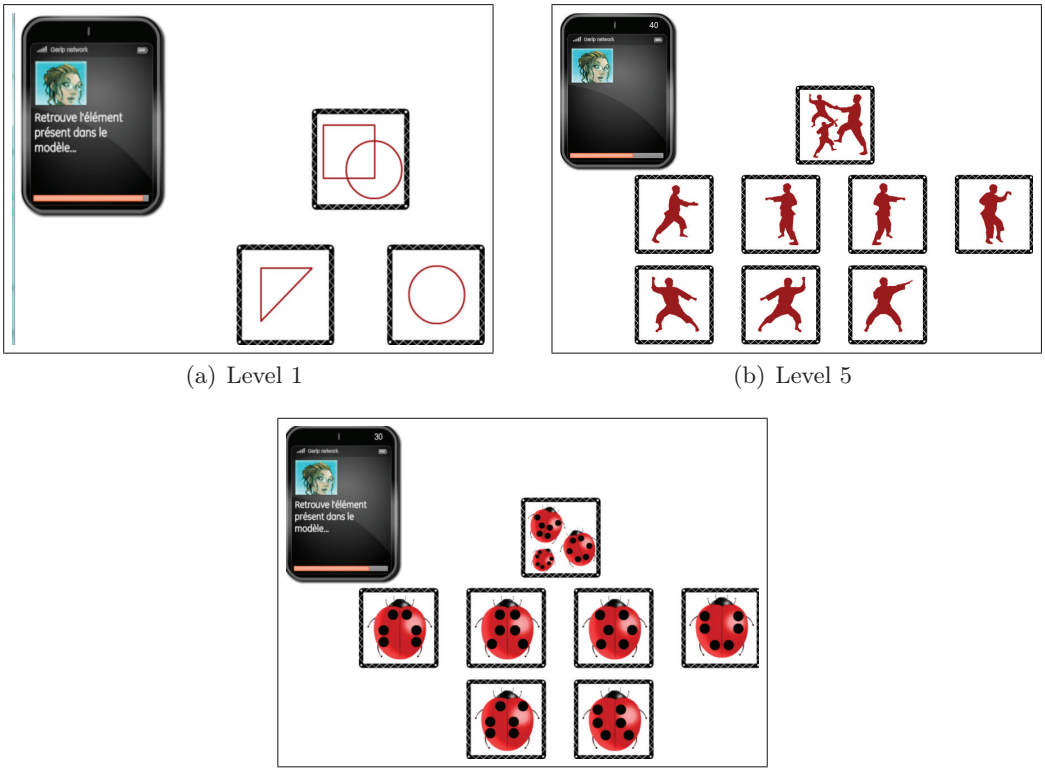
5.4 Mini-Games

Project CLES has about ninety mini-games in total. Each of these mini-games has nine levels of difficulty. In this section, we present some examples of mini-games of some cognitive functions.

5.4.1 Identify Intermixed Objects (Objets entérmélés à identifier)

The purpose of this game is to test the visual-perception of a child aged between 6-12 years. The game goes as follows: the learner sees a *Model* which contains more than one intermixed element. S/he has a number of single elements as possible responses. The learner needs to identify, among the possible responses, the element which appears in the Model. Furthermore, the learner has to do it in the allotted time.

The game helps a child to identify individual objects intermixed with other objects of the same nature. For example, the child has to identify a square, which intermixes with a triangle or a circle. The level of difficulty can be adjusted, according to a learner, by modifying the game's parameters. Different levels of the mini-game can be seen in the figure 5.2.



(a) Level 1

(b) Level 5

(c) Level 9

Figure 5.2: Different difficulty levels of the mini-game "Identify intermixed objects"

The parameters for this game are:

Type of images : Many types of images can be shown to the learner, for example, geometric shapes, letters, numbers, and characters, etc. The type of image also depends also upon the level of difficulty of the exercise.

Number of images in the model : The model can be made easier or difficult

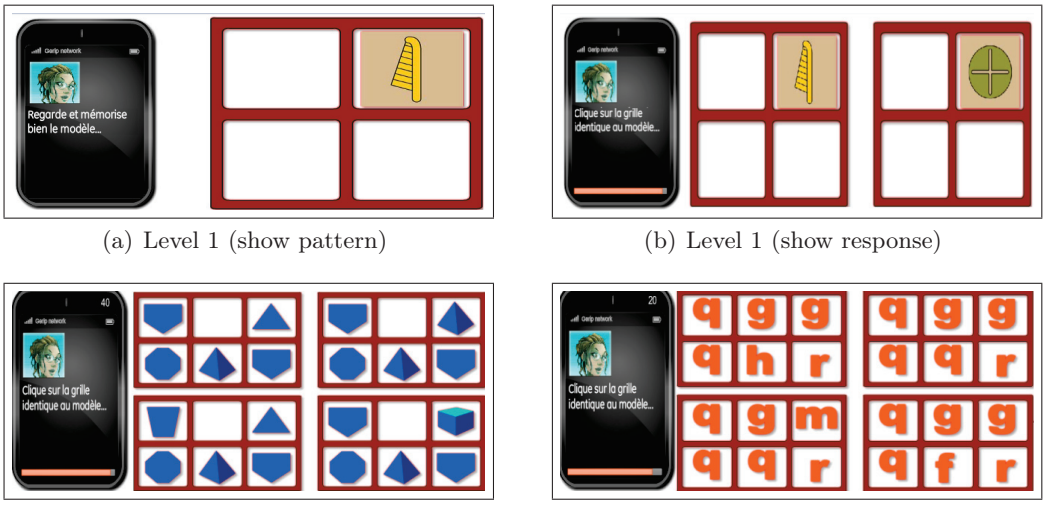
by increasing the number of images. For example, figure 5.2 shows the different levels of this game. The part *a* shows easiest level and the model contains only two elements. The part *b* shows a more difficult level and the number of elements are three in number.

Time : the time given to the player to respond can also vary according to the level of difficulty. The easiest level gives the most time to the learner and the most difficult level contains the least amount of time.

Possible responses : the easiest level of the game has the least number of options for the learner to choose from, while the number of responses increases with the difficulty level of the game.

5.4.2 Memorize and Recall Objects (Mémoire et rappel d'objets)

The purpose of this game is to test the ability to recall memorized objects of its learners. Figure 5.3 shows the interface of a mini-game on the memory. As this figure shows, the game displays a series of images (figure 5.3 part 'a') that the learner must memorize. After a certain time period, the images disappear, the learner needs to select these images among several propositions (figure 5.3 part 'b').



(a) Level 1 (show pattern)

(b) Level 1 (show response)

(c) Level 5

(d) Level 9

Figure 5.3: Different difficulty levels of the mini-game "Memorize and Recall objects"

This game can also be parametrized with the following parameters:

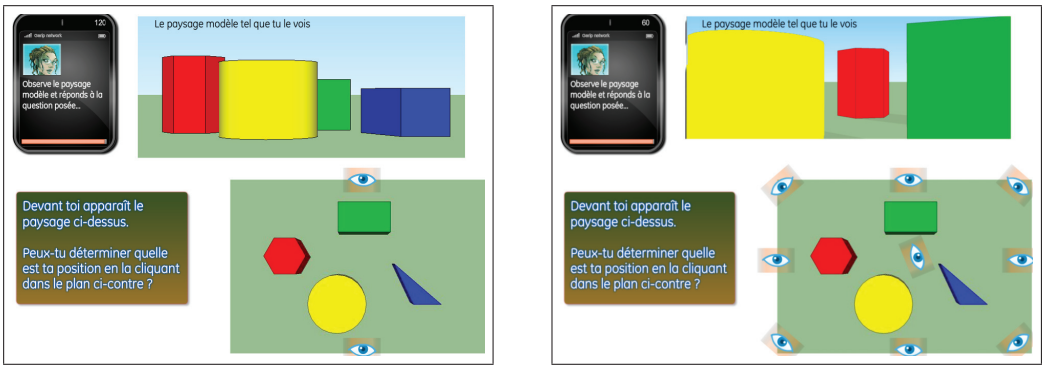
Number of images : The number of images to be memorized can be increased with the level of difficulty. More complex problems, level 5 and level 9 are shown in figure 5.3 (part c) and 5.3 (part c), respectively.

Display time : Easier level of the game allow the learner more time to memorize the image, and the more difficult levels allows less time to memorize.

Number of propositions : Easier levels allows fewer options and difficult levels show more options to the learner. For example, figure 5.3, which represents the easiest level, shows two responses. However, four responses are shown in figure 5.3 (part c), which represents the most difficult level.

5.4.3 Point of View (Point de vue)

The objective of this game is to test the visuo-spatial capacities of its learners. This means to test whether a person can orient himself in his surroundings. The learner sees a model, which contains a landscape shown from a point of view (as shown in figure 5.4). The landscape consists of a set of geometrical objects, arranged in a manner. Afterwards, the learner can see the same landscape from a birds-eye-view, and needs to select, in an allotted time, the view-angle of the model.



(a) Level 1 (b) Level 9

Figure 5.4: Different difficulty levels of the mini-game "Point Of View"

Like all the games, this game also can be made difficult or easy using some parameters:

Number of angles : The number of angles in the responses can be increased or decreased to change the level of difficulty. For example, nine angles are given to the learner, to choose from in figure 5.4 (part b) (level 9), while only two are given in figure 5.4 (part a) (level 1).

Response time : Easier level of the game allow the learner more time choose the response, and the more difficult levels allows less time to make the choice.

Number and complexity of images : The number of images can be increased or decreased to modify the difficulty. The model can contain images of different complexities, for example, geometrical, numbers, non-geometrical objects.

5.4.4 Complete the Series (Séries logiques à compléter)

This game is about the logical reasoning. In this game, the learner can see a number of images (figure 5.5) arranged in a logical series. These images can contain numbers, characters, geometrical objects, etc. One of these images contains a question mark. The objective of learner is to arrange in a logical series these images by choosing the adequate image among several possible responses.



(a) Level 1 (b) Level 9

Figure 5.5: Different difficulty levels of the mini-game "Complete the series"

The difficulty of the series can be modified using the following parameters:

Complexity : The logical series can be made simpler or more complex to vary the difficulty. For example, a rather simple series is shown in figure 5.5 (part a) (requiring a simple numeric calculation), to a more complex series, as shown in figure 5.5 (part b), requiring much more complex reasoning.

Number of responses : More the number of possible responses more it will be difficult to guess the response, or to make the choice by the learner, in case of confusion.

Response time : Easier level of the game allow the learner more time choose the response, and the more difficult levels allows less time to make the choice.

After the presentation of some of the many mini-games of project CLES, we present the actual modelling of the CLES's knowledge via our proposed models and tools.

5.5 CLES Modelling

Recall that, in addition to the eight main cognitive functions (c.f. section 5.2.5), there are also other sub-functions of each of these main functions. The knowledge of all these functions needs to be entered into the GOALS platform in order to generate adaptive pedagogical scenarios. This entering of knowledge will also help us to verify whether our proposed models are sufficient enough for CLES.

The entities called *cognitive functions* or *sub functions* in the context of CLES, we refer to them as *Concepts* in GOALS for modelling purposes. Since, the entire

CLES's knowledge model, containing all the concepts, pedagogical resources and game resources, is quite large, we present the modelling in readable pieces. The figure 5.6 shows the modelling of the eight main concepts (corresponding to the eight cognitive functions) and the relations that exist between them. All these relation are of *Required* Type.

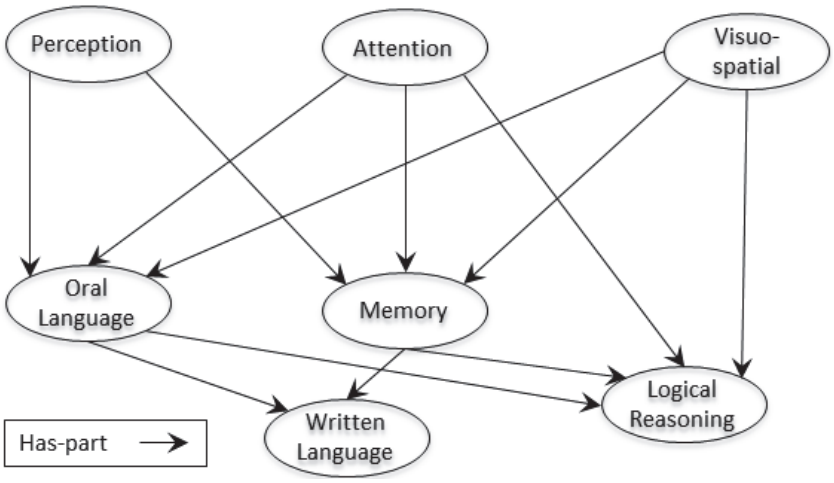


Figure 5.6: Modelling of the main eight cognitive concepts

The figure reads as follows: Perception, Attention, and Visuo-Spatial are basic concepts i.e. they do not require the competence of any other concept. The concepts Memory and Oral Language require sufficient knowledge of Perception, Attention and Visuo-Spatial. Written Language requires knowledge of Oral Language and Memory. Logical Reasoning requires Oral Language, Memory, Attention and Visuo-Spatial.

In this section, we present, firstly, the modelling of the eight cognitive functions and sub functions as main concepts and sub-concepts respectively. Secondly, we present the modelling of the mini-games as pedagogical resources. Finally, we present the modelling the objects of the Tom O'Connnor game as serious game resources.

5.5.1 Main Concept Modelling

Recall that, we have presented the modelling of the domain concept knowledge in chapter 3 (section 3.2.2). According to these models, the modelling of the eight cognitive functions of the project CLES is as follows:

Perception : $\text{Concept}_{\text{Perception}} = <\text{Perception,null}>$

Attention : $\text{Concept}_{\text{Attention}} = <\text{Attention,null}>$

Visuo-Spatial : $\text{Concept}_{\text{Visuo-Spatial}} = <\text{Visuo-Spatial,null}>$

Memory : $Concept_{Memory}$ = <Memory,R_{Memory}>

- R_{Memory} = <Memory, T_R, RC_1, RC_2, RC_3>
- T_R = <"Required", "Prerequisite relation", $F_{Required}$>
- RC_1: <Perception, null, 30>
- RC_2: <Attention, null, 30>
- RC_2: <Visuo-Spatial, null, 40>

Oral Language : $Concept_{OralLanguage}$ = <Oral Language,$R_{OralLanguage}$>

- $R_{OralLanguage}$ = <Oral Language, T_R, RC_1, RC_2, RC_3>
- T_R = <"Required", "Prerequisite relation", $F_{Required}$>
- RC_1: <Perception, null, 30>
- RC_2: <Attention, null, 30>
- RC_2: <Visuo-Spatial, null, 40>

Written Language : $Concept_{WrittenLanguage}$ = <Written Language,$R_{WrittenLanguage}$>

- $R_{WrittenLanguage}$ = <Written Language, T_R, RC_1, RC_2>
- T_R = <"Required", "Prerequisite relation", $F_{Required}$>
- RC_1: <Oral Language, null, 30>
- RC_2: <Memory, null, 30>

Logical Reasoning : $Concept_{LogicalReasoning}$ = <Logical Reasoning,$R_{LogicalReasoning}$>

- $R_{LogicalReasoning}$ = <Logical Reasoning, T_R, RC_1, RC_2, RC_3, RC_4>
- T_R = <"Required", "Prerequisite relation", $F_{Required}$>
- RC_1: <Oral Language, null, 30>
- RC_2: <Memory, null, 30>
- RC_1: <Attention, null, 30>
- RC_2: <Visuo-Spatial, null, 30>

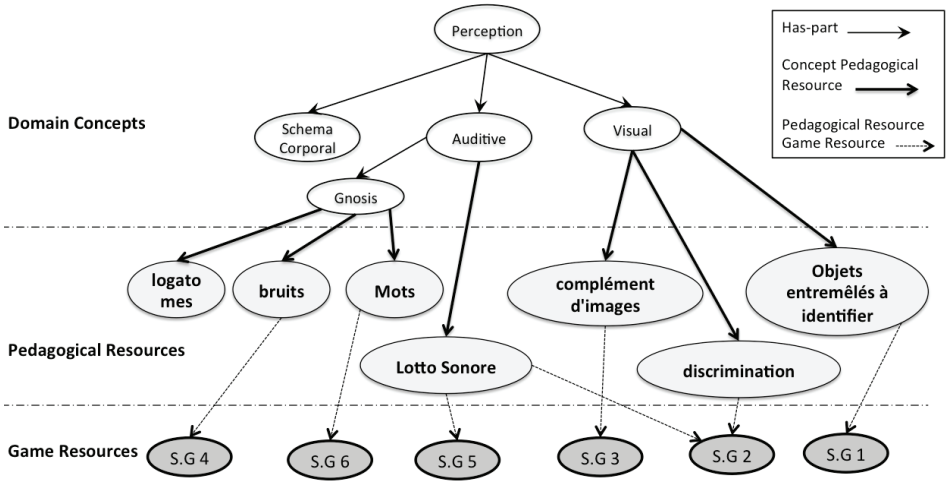


Figure 5.7: Complete model of Perception

5.5.2 Sub-Concepts Modelling

Each of the eight main concepts has some sub-concepts. These sub-concepts represent specific cognitive functions. Furthermore, some these concepts also have some pedagogical resources related to them. The modelling of some of these sub-concepts can be seen in the figures 5.7, 5.8, and 5.9. These figures show the model of Perception, Memory and Written Language, respectively.

Some of the sub concepts are in relation with one or more pedagogical resources. Moreover, the pedagogical resources are also in relation with one of the Game Resources. These resources represent the objects in the Tom O'Connor serious game environment. If the scenario generator selects a pedagogical resource, related to a game resource, to present to a learner, then the generator hides the resource behind the game resource. For example in figure 5.7, the resource "Lotto Sonore" is in relation with the serious game resources "S.G.2" & "S.G.5", therefore, in the serious game, "Lotto Sonore" can be put behind either "S.G.2" or "S.G.5". We present, here, the modelling of Perception.

5.5.2.1 Perception

Perception : $\text{Concept}_{\text{Perception}} = <\text{Perception}, \text{R}_{\text{Perception}}>$

- $\text{R}_{\text{Perception}} = <\text{Perception}, \text{T}_{\text{H.P}}, \text{RC}_1, \text{RC}_2, \text{RC}_3>$
- $\text{T}_{\text{H.P}} = <$"Has-Pars", "Has-Parts Relation", $\text{F}_{\text{H.P}}>$
- RC_1: <Visual, null, 30>
- RC_2: <Auditive, null, 30>
- RC_3: <Schema Corporal, null, 40>

Visual : $\text{Concept}_{\text{Visual}} = <\text{Visual}, \text{null}>$

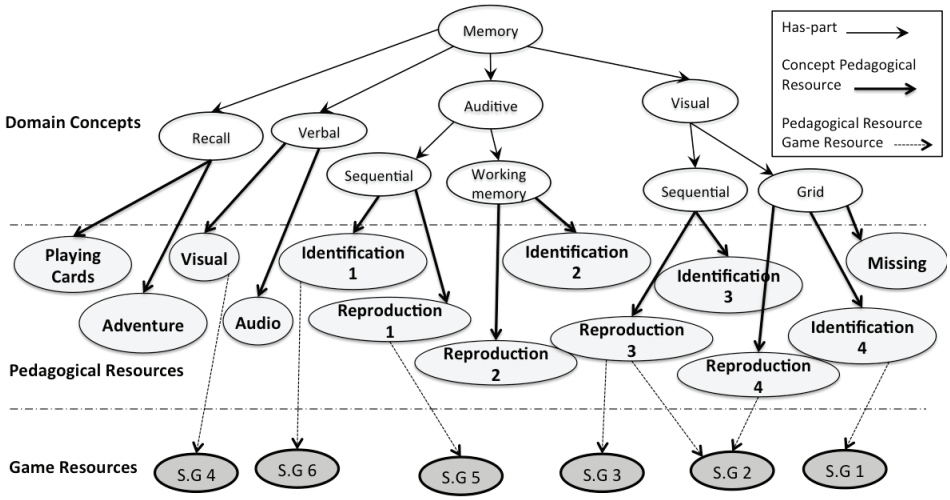


Figure 5.8: Complete model of Memory

Body Schema : Concept$_{\text{SchemaCorporal}}$ = <Schema Corporal, null>

Auditive : Concept$_{\text{Auditive}}$ = <Auditive, R$_{\text{Auditive}}$>

- R$_{\text{Auditive}}$ = <Auditive, T$_{\text{H.P}}$, RC$_{\text{A1}}$>
- T$_{\text{H.P}}$ = <"Has-Pars", "Has-Parts Relation", F$_{\text{H.P}}$>
- RC$_{\text{A1}}$: <Gnosis, null, 100>

Gnosis : Concept$_{\text{Gnosis}}$ = <Gnosis, null>

5.5.2.2 Pedagogical Resource Modelling

The modelling of the pedagogical resources is according to the model presented in chapter 3 (section 3.2.3). The *ImpFunc* denotes the *impact function*.

Identify Intermixed Objects : Identify Intermixed Objects : <

Id$_{\text{IdentifyIntermixedObjects}}$, "mini-game", < <"Types of images","" >,<"Number of images in the model","" >,<"Time","" >,<"Possible responses","" > >, null, null, <text,"Description of the mini-game">,< <ID$_{\text{Visual}}$,10,*ImpFunc*> > >

Complement of Image : Complement of Image : <Id$_{\text{ComplementofImage}}$, "mini-game", < <"Types of images","" >,<"Number of images in the model","" >,<"Time","" >,<"Possible responses","" > >, null, null, <text,"Description of the mini-gamese">,< <ID$_{\text{Visual}}$,10,*ImpFunc*> > >

Discriminate : Discriminate : <Id$_{\text{Discriminate}}$, "mini-game", < <"Types of images","" >,<"Number of images in the model","" >,<"Time","" >,<"Possible responses","" > >, null, null, <text,"Description of the mini-game">,< <ID$_{\text{Visual}}$,10,*ImpFunc*> > >

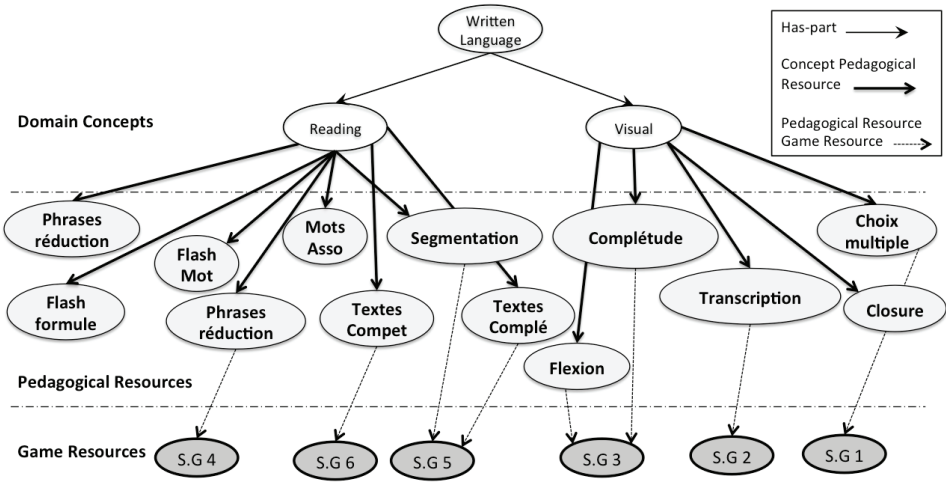


Figure 5.9: Complete model of Written Language

Lotto Sonore : Lotto Sonore : <Id$_{\text{LottoSonore}}$, "mini-game", < <"Types of images","" >,<"Number of images in the model","" >,<"Time","" >,<"Possible responses","" > >, null, null, <text,"Description of the mini-game">,< <ID$_{\text{Auditive}}$,10,$ImpFunc$> > >

Sound : Sound : <Id$_{\text{Sound}}$, "mini-game", < <"Types of images","" >,<"Number of images in the model","" >,<"Time","" >,<"Possible responses","" > >, null, null, <text,"Description of the mini-game">,< <ID$_{\text{Gnosis}}$,10,$ImpFunc$> > >

Word : Word : <Id$_{\text{Word}}$, "mini-game", < <"Types of images","" >,<"Number of images in the model","" >,<"Time","" >,<"Possible responses","" > >, null, null, <text,"Description of the mini-game">, < <ID$_{\text{Gnosis}}$,10,$ImpFunc$> > >

Logatome : Logatome : <Id$_{\text{Logatome}}$, "mini-game", < <"Types of images","" >,<"Number of images in the model","" >,<"Time","" >,<"Possible responses","" > >, null, null, <text,"Description of the mini-game">, < <ID$_{\text{Gnosis}}$,10,$ImpFunc$> > >

5.5.2.3 Serious Game Resource Modelling

The modelling of the serious game resources is according to the model presented in chapter 3 (section 3.2.4).

S.G.4 : <Ids.G.4, <"type","gaming object">, < <ID$_{\text{Sound}}$,"related"> > >

S.G.6 : <Ids.G.6, <"type","gaming object">, < <ID$_{\text{Word}}$,"related"> > >

S.G.1 : <Ids.G.1, <"type","gaming object">, < <IDIdentifyIntermixedObjects, "related"> > >

S.G.2 : <Ids.G.2, <"type","gaming object">, < <IDDiscrimination,"related"> , <IDLottoSonore,"related"> > >

5.6 Using GOALS for CLES

As we have mentioned earlier, the entire model of CLES is quite large and difficult to visualize on a single piece of paper. Therefore, we present just a small part of the Model to show the modelling of CLES in GOALS.

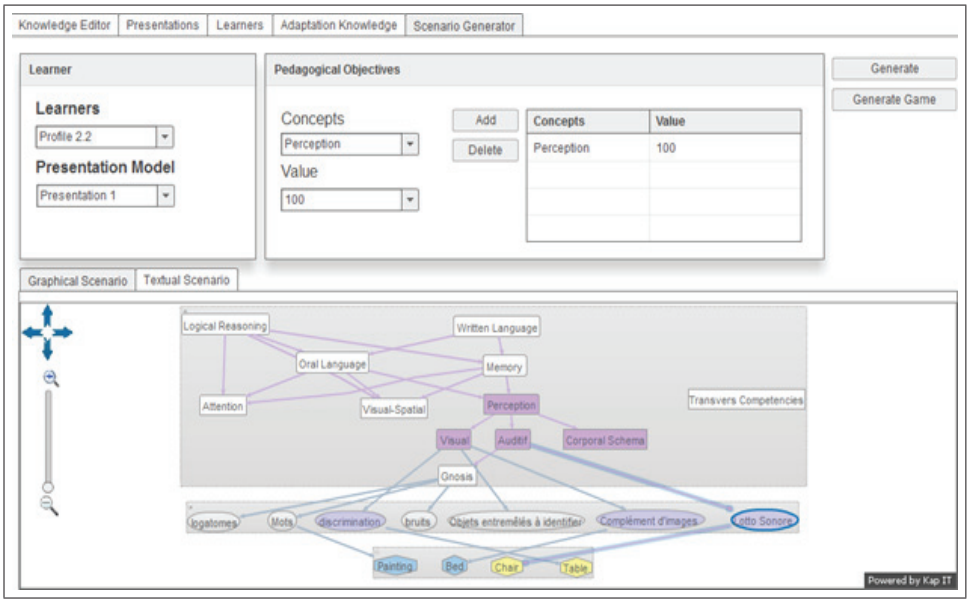


Figure 5.10: Modelling of CLES

This figure 5.10 shows the eight main concepts along with the complete sub-concepts of Perception. This figure also shows a generated scenario for a profile named "Profile 2.2" and pedagogical goals "Perception at 100". The generator has chosen the dark violet coloured concepts, the light violet coloured pedagogical resources and the yellow coloured in the generated scenario.

In the context of CLES, the modelling of its knowledge in GOALS was just one aspect of the problem. The modelled knowledge has to be tested to know whether the scenarios generated via this knowledge is good enough or not to be used in the actual system. We have performed this test via some experiments, presented in the next chapter.

Chapter 6

Evaluations

Contents

6.1 Introduction 122
6.2 State-Of-The-Art 122
6.3 Evaluation of Generator scenarios 126
6.3.1 Evaluation Protocol 127
6.3.2 Experiment and results 131
6.4 Study of the impact of serious games on learners 135
6.4.1 Evaluation Protocol 135
6.4.2 Experiment and results 136

The objective of this section is to present the different evaluations that we have conducted to validate our propositions. We start this chapter by a literature review of evaluation methods in section 6.2. Section 6.3 presents the protocol evaluation we conducted to study the validity of our scenario generator. For this, we present this protocol in section 6.3.1 and the actual experiment in section 6.3.2. Section 6.4 reports on the protocol validation we conducted to study the impact of adapted scenario on learners. For this, we present this protocol in section 6.4.1 and an experiment with real learners in section 6.4.2.

6.1 Introduction

In line with the research questions, presented in chapter 1 (section 1.4), we have answered, in terms of contributions (chapter 3), the first two research questions namely: knowledge models and a model of a generic generator of adaptive scenarios in serious games. This chapter presents two experimentations, which answer the third research questions. Recall that, these questions are:

> **Question 3 : How to validate the functioning of the scenario generator (the knowledge models and strategies used to generate the pedagogical scenarios)? and How to study the impact of the generated scenarios on the actual learning of the learner?**

We answered the first part of this question by conducting an experimentation with a domain expert to verify the quality of the scenarios generated by the scenario generator. We present the answer to the second part by conducting an experimentation with real-world learners, and studying the effect of the generated scenarios on learning.

In the next section, we present a literature review of the evaluation methods of similar approaches.

6.2 State-Of-The-Art

According to [Cooley 1976], "An evaluation is a process by which relevant data are collected and transformed into information for decision making". Much research has been taken place, in the field of Intelligent Tutoring Systems (ITS), to find out what to evaluate and how to evaluate. [Scriven 1967] divided the evaluation process into two broad phases. Formative Evaluation and Summative Evaluation.

According to [Mark 1993], "Formative evaluation occurs during design and early development of a project and is oriented to the immediate needs of developers who are concerned with improving the design and behaviour of a system". Whereas, summative evaluation is "concerned with the evaluation of completed systems and the making of formal claims about those systems" [Mark 1993].

While formative evaluation tests the relation between the architecture of the system with its behaviour [Littman 1988] (first part), summative evaluation allows the designer of the system to make formal claims about it at the end of the system's development [Littman 1988] (second part).

From an ITS's point of view, [Mark 1993] presents with an excellent review of the point-of-views an evaluator could take while evaluating an ITS. Some of these point-of-views are:

- Proofs of correctness
- Criterion-based evaluation

- Expert knowledge and behaviour
- Certification
- Sensitivity Analysis
- Pilot Testing
- Experimental Research

We have performed the evaluation to provide a proof of correctness of the knowledge models and scenario generation process and verify the generator's behaviour and the expert's knowledge.

Many Researches have tried to identify the methods for evaluating adaptive systems ([Arruabarrena 2006, Weibelzahl 2002, Van Velsen 2008, Brusilovsky 2001b, Gena 2005, Masthoff 2003, Paramythis 2009, Raibulet 2010], etc). They tried to identify the aspects that should be evaluated in an adaptive system and how the evaluation should be performed. However, looking at the evaluation process in approaches similar to ours, we have observed that these approaches have not followed the above mentioned evaluation methods to perform their evaluations.

Though, there are many approaches for generating courses or exercises. We have reviewed some of the strategies used by these approaches to evaluate their scenario generation process. [Ullrich 2010] presented a dynamic course generation approach called PAIGOS. PAIGOS generates dynamic and adaptive pedagogical courses according to the pedagogical goals of learners. The researchers behind PAIGOS have reported on the formative and summative evaluation. The aim of the formative evaluation was to find out whether the ".... an automatic selection of the educational resources would show an advantage over accessing the complete content within a single book." [Ullrich 2010]. The formative evaluation is performed by selecting two groups of learners, one group uses PAIGOS, the other uses a traditional book. After the experiment, each learner completed a questionnaire (based on LIKERT scale). This questionnaire evaluated the utility of the system.

The summative evaluation of PAIGOS uses a cooperative evaluation methodology [Monk 1993]. This evaluation takes place after PAIGOS was ready to use by learners. The objective was to evaluate the learning scenario generation process. The learner generates a scenario, interacts with it, and share his experience with the authors who stays with the learner during the whole process. This methodology allows researchers to guide the learner in real-time, and have feedback in real-time, as well.

[Kravcik 2004b] reports on the evaluation of the Web-based Intelligent Design and Tutoring System (WINDS). The purpose of WINDS is to facilitate the authoring process of adaptive pedagogical courses. The authors conducted an evaluation of WINDS to verify whether the courses created using the WINDS platform are up to the standard or not. The authors also wanted to find out whether the course creation process helps the course designers or not. They performed a formative evaluation

using four lecturers, each of them performed a certain set of tasks. During the experiments, the researchers carefully observed the lecturers. After the experiments, the lecturers answered a set of open-ended questionnaires to evaluate their experience. Furthermore, the observation of the lecturers by the authors also revealed some information that contributed in the amelioration of the system.

[Huang 2008a] presents an approach for generating automatically and intelligently auxiliary materials. The idea is to propose auxiliary learning materials, which provide more interactive and cooperative characteristics for the learning process. The authors conducted an evaluation to verify whether the algorithm can generate auxiliary materials, and achieve the expected convergence. The authors also asked the participants whether they liked the system's interaction, assistance, usability, and flexibility. They asked several users to use the proposed system, for the generation of auxiliary learning material and then presented the users with a questionnaire to record their opinion about the system.

[Zouaq 2008] showed an ontology-based approach for the dynamic generation of learning knowledge objects. This approach offers an alternative to static learning objects by generating dynamically learning resources called "Learning Knowledge Object" (LKO). The LKO is knowledge-based, theory-aware and dynamically generated. The purpose of the evaluation, conducted by the researchers was to confirm the validity and usability of their proposed approach. They also evaluated the semantic validity of the domain concept maps and the domain ontology. They compared the effectiveness of their approach with a similar tool for the same input using comparative evaluations. They also performed an empirical evaluation, the goal was to compare, on a given subject, LKO's and traditional learning objects with a set of learners.

Adaptive navigation support for parametrized questions for an object-oriented language (JAVA) (JavaGuide) [Hsiao 2009]. JavaGuide is an extension of QuizJET (Java Evaluation Toolkit). QuizJET supports authoring, delivery, and evaluation of parametrized questions for Java [Hsiao 2008]. JavaGuide adds the adaptive navigation support to QuizJET. The purpose of the evaluation was to verify whether the adaptive navigation in JavaGuide helps the learner in the learning process. The researchers analyzed the learners taking part in the evaluation both at an overall level and session by session level. They observed different statistical indicators and compared them. They also compared the impact of these tools on weak and strong learners. Finally, the learners answered a set of questionnaires to conduct a subjective evaluation of their experience with the system.

Another course and exercise sequencing approach for an adaptive hypermedia system and its evaluation could be seen in [Fischer 2001]. The authors explain how a knowledge library can be used to create exercises automatically. The evaluation conducted to demonstrate the advantages and drawbacks of the automatic course sequencing approach. This evaluation is based on a comparative method comparing between the output of the system with that of a human.

[Karampiperis 2005c, Karampiperis 2005a] used statistical techniques to generate a course most suitable to the learner. Instead of first selecting the concepts

and then for each concept selecting the educational resources, they first calculate all possible courses that reach a set of concepts and then select the best suited one, according to a utility function. The objective is to evaluate the quality of the generated courses for a learner by using comparative method. It consists in producing learning paths by using their approach, and then compared these paths with the learning paths produced by a simulated perfect rule-based AEHS, using a Domain Model and Media Space.

[Sangineto 2007] proposes an approach for a course generator LIA (Learning Intelligent Advisor). LIA generates a course by keeping into account the learning styles of a user. The objective of the evaluation here was to evaluate whether the learners using the course generator LIA perform better than those using a traditional system. The researchers conducted this evaluation by selecting a group of learners and testing them before the experimentation. Afterwards, they divided the learners into two groups. They train the first group by using traditional learning methods, and the second group by using LIA. Afterwards, the researchers presented both groups with a test, and then compared their performance on the test with their performances on the test before the experimentation. The researchers also conducted interviews with the learners' in-order to acquire their feedback on the usability of the system.

[Liu 2010] presented an approach for the automatic generation of review questions for a research paper. This approach first automatically extracts citations from students' compositions together with key content elements. Next, it classifies the citations using a rule-based approach, and then it generates questions based on a set of templates and the content elements. The objective of the approach's evaluation was to find out whether the questions generated by the approach are real enough or not. For this, they asked human judges (authors of research papers) to ascertain, whether a human (lecturer, tutor, generic) had asked these questions, or a system had generated them. Similar evaluations can be seen in [Masthoff 2002, Motiwalla 2007, Virvou 2001, Delozanne 2008].

All of the above mentioned works use some comparison to test the output of their approaches. The objective of our comparison is to validate or evaluate the utility of the proposed approach for its intended users. This allows the designer, using our approach, to verify whether the approach accomplishes the tasks that it needs to do. Normally, the researchers of an approach conduct these evaluations with the potential real-world users. This comparison usually includes comparing the output produced by an approach with some other traditional approach or system.. Throughout literature, these types of evaluations can be referred to as **Comparative Evaluation** [Vartiainen 2002], and we have applied this strategy to answer the research question (first part).

There are also evaluation techniques for verifying whether a proposed approach provides or helps in achieving learning. We have reviewed some of them that have performed similar evaluations.

As identified by [Grubišic 2006], to evaluate the effectiveness of an e-learning system, learners should take a pre-test, then they should be divided into two groups:

the control group and the experimental group. The *control group* uses the traditional learning and teaching process, and the *experimental group* uses the e-learning system. Afterwards, the researchers should conduct a post-test with the learners (also a check point test, if necessary), to measure the effectiveness of the approach.

[Papastergiou 2009] used the same principle to evaluate their approach. The aim of this study was to assess the learning effectiveness and motivational appeal of a computer game for learning computer memory concepts. The researchers have designed the game according to the curricular objectives and the subject matter of the Greek high school Computer Science (CS) curriculum, as compared to a similar application, encompassing identical learning objectives and content but lacking the gaming aspect. They used the process of presenting the learners with a pre-test, post-test and actual use with the game, to measure the learning gain. They divided the learners in a control group and an experimental group. Afterwards, the researchers compared the learners' performance to study the learning effects of the digital game.

[Martín-Gutiérrez 2010] reports on an AR-based application, which targets to improve spatial abilities among engineering students, thus enabling them to gain a better understanding of engineering graphics subjects. They wanted to evaluate the potential of Augmented Reality technology in university education. To conduct the evaluation, they asked the learners to do a pre-test, then they divided them into a control group and an experimental group. The experimental group uses the system while the control group uses traditional methods. Afterwards, they ask all the learners to conduct a post test. The researchers compared the learner's performances in the pre-test and the post-test to measure the effect of the system. In addition, they applied An analysis of covariance (ANCOVA). The ANCOVA method allows eliminating the difference of pre-test scores between groups, and then the adjusted post-test scores, revealing the real effects of the experimental treatment. Furthermore, [Lepp 2008, Fossati 2008, Beal 2010, Crowley 2007, VanLehn 2005, Kalloo 2011, Liao 2011, Villamañe 2001, Stankov 2004, Sykes 2005] use the same principles although their systems are quite different in purpose.

Considering, the successful use of pre and post test strategy to measure the effects of a system, we have answered, the second part of the research question, using an evaluation based on the same strategy.

In the next section, we present the experiment we conducted to evaluate the proposed scenario generator.

6.3 Evaluation of Generator scenarios

In order to answer the first part of the third research question, we have proposed an evaluation protocol and conducted an experimentation in the context of the Project CLES. More precisely, the objective of this evaluation is the validation of:

- The scenario generator's working: this means the validation of the concept selection strategy which we have defined for each type of relations, and

- The knowledge models: it means to validate the concepts and the relations that we've introduced into the system in the context of project CLES.

In the next section (6.3.1), we present the evaluation protocol. We have implemented this protocol as part of an experiment that we have conducted with an expert therapist in the context of the project CLES. The section 6.3.2 presents the results of this experimentation.

6.3.1 Evaluation Protocol

The evaluation protocol was our guide in the experimentation process.

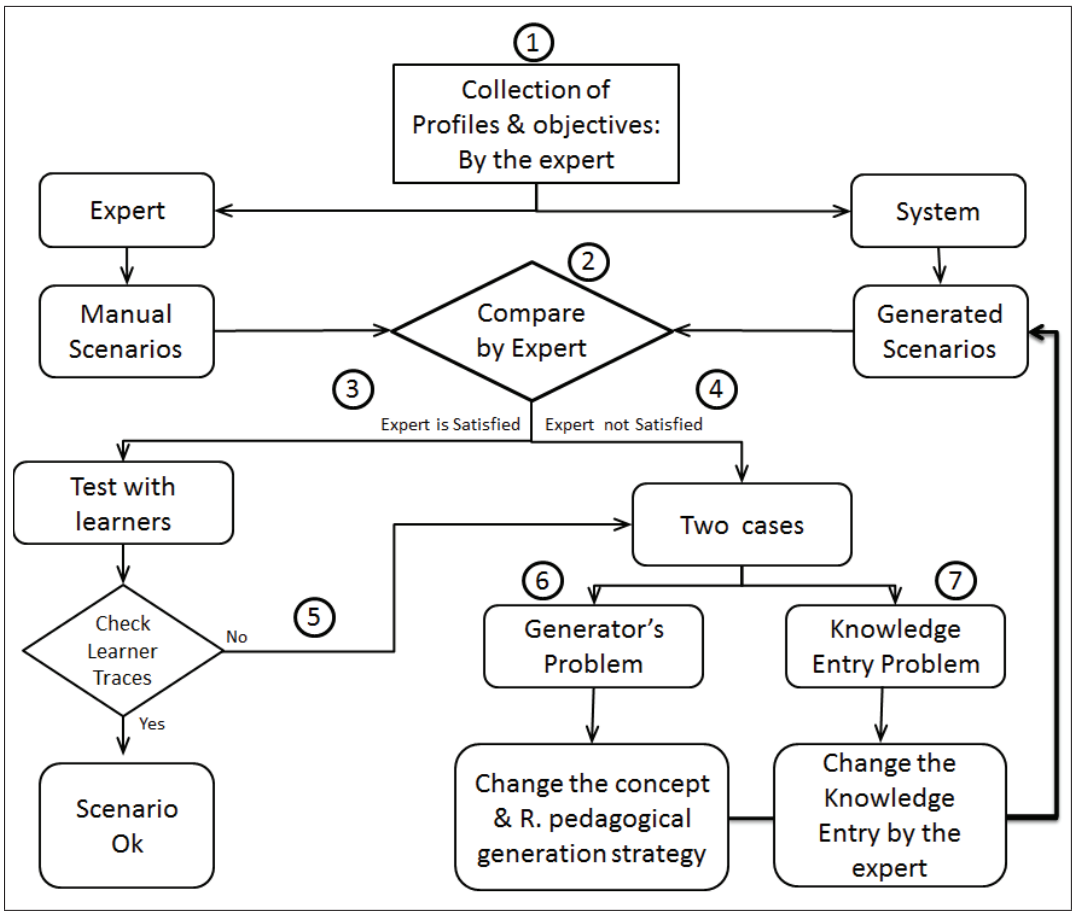


Figure 6.1: Evaluation Protocol

The flow chart of this protocol can be seen in the figure 6.1. The basic strategy that we have adapted is comparative evaluation [Vartiainen 2002] i.e. it consists in comparing the pedagogical scenarios created manually by the domain expert with the pedagogical scenarios generated automatically by the generator for the same input. This input corresponds to the domain knowledge and profile of some learners.

Furthermore, during the evaluation process we conduct an Elicitation Interview [Bull 1970] with the expert. The purpose of this interview is to help the expert in explicating his thinking process, how he reasons while creating a pedagogical scenario. The working of this protocol is as follows:

At first, the expert creates a certain number of learner profiles (1). As the expert has a vast experience in his/her respected field, s/he can give us the profiles that are much close to the reality.

The profiles should also be diverse i.e. different profiles should contain different competencies. This will help us in determining whether the generator can handle diverse cases or not.

Afterwards, the expert sets some pedagogical objectives for each of these profiles, then creates, for each case (one profile + pedagogical objectives), an adapted pedagogical scenario. Afterwards, we introduce all cases defined by the expert into the generator in order to automatically generate the pedagogical scenario for each case (profile + learning objective).

(2) Then the expert compares the two sets of scenarios (defined by the expert and generated by the generator). While the expert is performing the comparison, we ask him/her, via an explication interview, to verbalize the process of comparing the two scenarios. We film the expert during the whole evaluation process. To help the expert in his comparing process, we can ask him/her the following questions:

- Are the concepts in the two scenarios same? If no, the concepts that are different, are they just a matter of choice i.e. one concept can be replaced by the other? If no, will these concepts hinder the learner in achieving his/her pedagogical objectives?

- Are there some concepts missing that are necessary to achieve the pedagogical objectives?

- Are the selected pedagogical resources similar or not? If no, do the selected pedagogical resources are of the correct type and belong to the correct concept? Do the selected pedagogical resources hinder the learner in achieving his/her pedagogical objectives?

- Is the level of difficulty selected of the pedagogical resources good or not? If no, can the learner achieve his/her pedagogical objective with the current levels?

- What is the level of satisfaction of the expert for the generated scenario?

The result of this comparison will be either the expert will find the scenarios similar (3) or not similar (4). In the following sections, we describe the protocol, to be followed, for both of these cases.

6.3.1.1 Scenario Similar

If the expert is satisfied of the scenarios generated by the generator, then the real-world learners use the generator. Ideally these learners should have the same profiles as entered in the generator. If this is not the case, their profiles must be entered into the generator.

The learners should be asked how difficult are the scenarios. If possible this phase should be filmed for a posteriori analysis. The learners' interaction traces will also help us in answering this question. Indeed, the analysis of the traces can help us in identifying whether a learner is finding the scenario difficult, for example, if s/he is constantly failing the exercises, then we can assume that the exercise are not according to the learner. Similarly, if the learner is answering the exercises quickly, then we can conclude that the learner is finding the exercises easy to solve.

If the learners find the scenarios too easy or too difficult (5), then this will imply that either the knowledge entered in the system by the expert can be improved, or the system is not generating the scenarios properly. In these cases or if expert not finds the scenarios similar, (4) then we will do the following.

6.3.1.2 Scenarios Dissimilar

If the expert is not satisfied then two cases are possible:

1. The system's generator is not working properly (6)
2. The knowledge entered in the system by the Expert is not correct (7)

The system is not working properly if one or more of the following are true:

- The masteries of the selected concepts are not calculated correctly given the learner's profile and pedagogical objectives.
- The algorithms, used to calculate the concepts' masteries, are not doing the calculation according to the expert's understanding.
- Some of the concepts are not selected despite proper relations in the concept graph.
- The presentation model is not followed correctly.
- The presentation model is not complete enough i.e. the expert cannot do what s/he wants to do with the presentation model for example s/he cannot include pre-requisite concepts' resources etc.
- The pedagogical resources are not selected correctly i.e. some resources that should have been selected are not selected.
- The adaptation knowledge is not applied correctly.
- The relations are not sufficient to model the expert's needs.

Effectively, if any of the above mentioned points are true then we've to review the:

Concept selection strategy: This means we have to review the selection of concepts based on different relations and the calculation of masteries based on them. Currently we've four kinds of relations: Has-Parts, Required, Type-Of and Parallel.

Pedagogical Resource selection strategy: Here, we have to review the pedagogical resource selection strategy. Currently, we, according to the presentation model, select all the resources related to a concept. Then, we verify whether a learner has already seen or mastered a resource or not. If this is the case, we ignore that resource and proceed on the next one.

If none of the cases are applicable then maybe the expert has made some error in entering the knowledge in the system. We can identify this mistake by asking the expert some of the following questions:

- Ask the expert whether there is a concept missing in the scenario? If yes then ask him to create the concept. If the expert had not forgotten the creation of the concept then:
  - Ask him to check whether he had missed to link the related concepts? If he had linked them.
    - Ask him whether he linked the concepts with the correct type of relation or not?
- Ask the expert if there is a concept, which is present in the scenario that should not be presented? If yes:
  - Ask him to check the relations and the type of relations between the concepts?
- Ask the expert whether there is a pedagogical resource missing? If yes
  - Ask him to make sure that the pedagogical resources are correctly related with the concerned concepts.
- If the calculated masteries are not correct then ask the expert to re-verify the values between the concepts. If the values are correct then
  - Ask the expert to verify the values of concepts in the learner's profile.
- If the level of the pedagogical resources is not correctly selected then ask the learner to re-verify the adaptation knowledge.
- If the sequence of the selected resources in the scenario is not correct then ask the expert to re-verify the presentation model.

Following the above mentioned protocol an experiment with a speech therapist. This experiment is described in the next section.

6.3.2 Experiment and results

We have applied this evaluation protocol on the generator GOALS in the context of the project CLES (this project is described in the chapter 5). Recall that, through GOALS, we can generate pedagogical scenarios, keeping into account the specificities of serious games, adapted to the learners. The chapter 3 describes the working and the architecture of this generator.

The experiment took place in the presence of a domain expert. This expert has an experience of more than 20 years as a speech therapist. He has participated in the development of many computer based solutions for persons in the situation of cognitive disabilities. He is aware with the technological advancements in the field; hence, he is an ideal person for passing a judgement on the performance of the generator.

The idea of this experimentation is to ask the expert to generate some scenarios, given some pedagogical goals and learner profiles, and the generator will do the same, for the same input. Then, the expert will compare the scenarios. The process of comparison depends upon the expert as he the best person to judge whether to let the generator generate the scenarios according to the learner or not.

To start, we have used the domain knowledge of the project CLES (see chapter 5) for this experiment. Since, the domain model of CLES is quite large, and the expert could have found the generated scenarios a bit difficult to evaluate, therefore, we have decided to break the CLES's knowledge structure into three substructures. In each of these structures, we modelled the eight main concepts and the sub-concepts of only one of the main concept. Thus, the three main concepts we have detailed are:

- Written Language
- Perception
- Memory

The sub-structures are shown in the figures 6.2, 6.3, and 6.4.

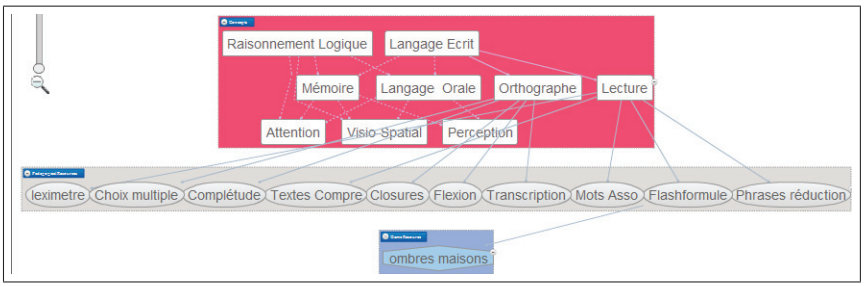


Figure 6.2: Written Language sub-structure

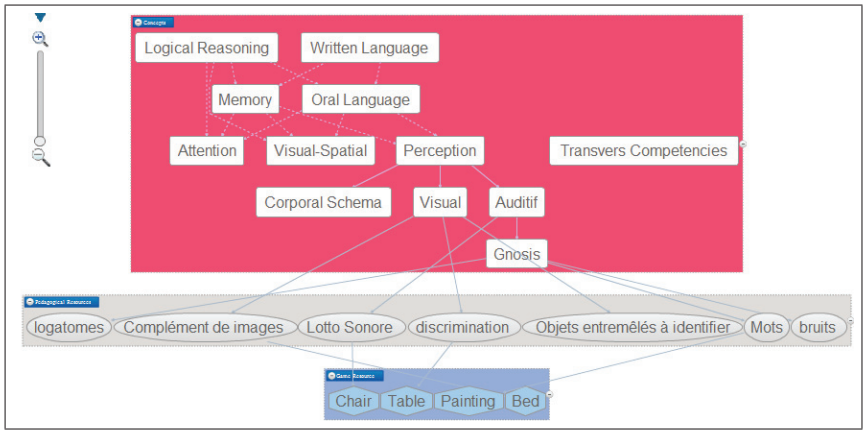


Figure 6.3: Perception sub-structure

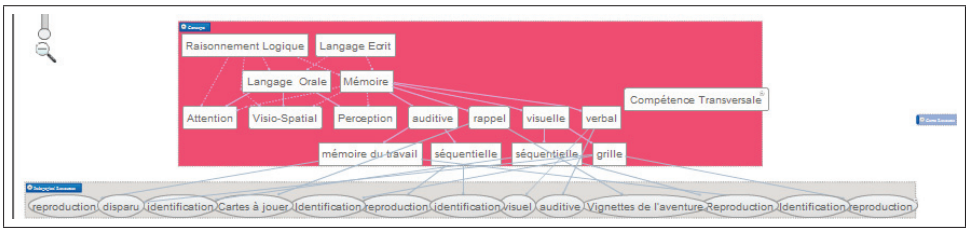


Figure 6.4: Memory sub-structure

In addition to these models, we created some learner profiles. In the project CLES, the initial value of a profile depends upon the age of the learner. We have applied the same principle in creating profiles based on ages of 8, 14 and 18 years. This selection of ages covers the whole range of ages of CLES. Furthermore, for each these ages we have created two types of profile : without disabilities and with disabilities. Therefore, we have created six profiles for each of the three selected substructures. The details of the profile for any concept are:

- Profile 1: 8 years, no disability in concept x
- Profile 2: 8 years, disability in concept x
- Profile 3: 14 years, no disability in concept x
- Profile 4: 14 years, disability in concept x
- Profile 5: 18 years, no disability in concept x
- Profile 6: 18 years, disability in concept x

The concept x is the divided concept in a substructure. This process gave us eighteen profiles in total. Profiles 1-6 for the written language substructure, profiles

7-12 for the perception substructure and the profiles 13-18 for the memory substructure. We have discussed the process of creating profiles with the expert, and he expressed his satisfaction on the process.

Afterwards, we asked the expert to give sufficient values to the profiles. With a vast amount of experience the expert had, he performed this task quite accurately. One of the profile created by the expert can be seen in the figure 6.5. The circles represent the domain concepts, and the red coloured numbers represent the profile values. For example, the value of the domain concept Memory in the learner profile is 80.

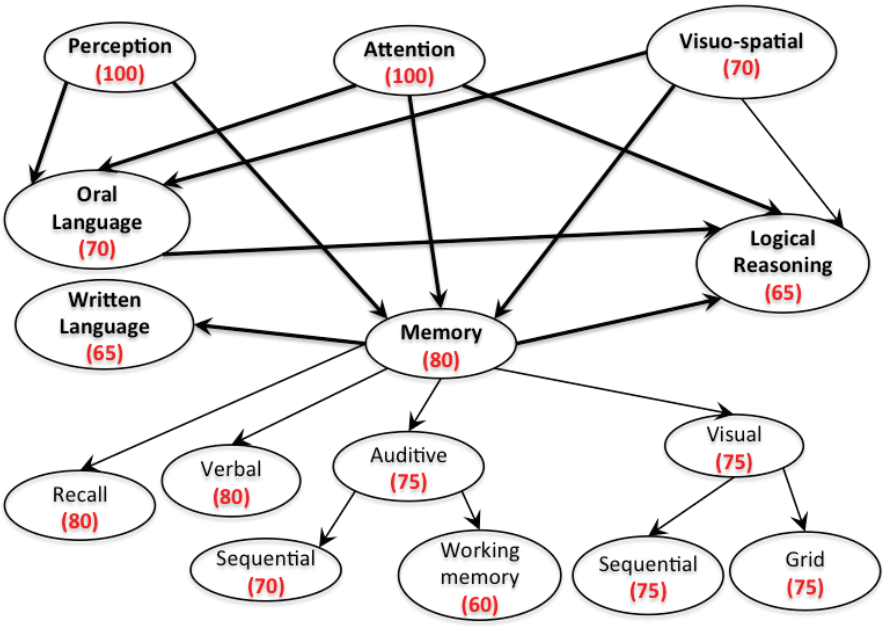


Figure 6.5: Profile of a child of 18 years having a disability in Memory

After the creation of profiles, we asked the expert to set appropriate pedagogical goals for each profile. These goals keep into account the specificities of every profile. For example, the expert gave higher values to the profiles with no cognitive disability, and a higher pedagogical goal/objective, then to the profiles with a cognitive disability. While the expert was fixing the goals, we asked him what factors he was taking into account while fixing a goal. This helped us in gaining valuable information about the domain modelling process and the fixing of pedagogical objectives according to a profile.

After the creation of the profiles, the expert created the pedagogical scenarios for every profile and their respective pedagogical objectives. In the meantime, we put, into the generator of GOALS, the values of profiles and the pedagogical objectives in order to generate automatically the adapted scenarios. While the expert was creating the scenarios, we conducted an elicitation interview with him, the objective being, helping the expert to explicit his thinking process. Afterwards, we asked the

expert to compare the two scenarios (expert and GOALS), created for the same input.

Throughout the experimentation, we filmed the expert in order to analyse his workings a posteriori. Thus, we have analysed about two hours of video. We performed the video analysis by using the tool ADVENE[1][Aubert 2004, Aubert 2005], and as a result of this analysis, we have detected some errors concerning the CLES's domain knowledge and the generator's functioning. ADVENE is a video analysis tool based on annotations (the figure 6.6 shows an interface of ADVENE).

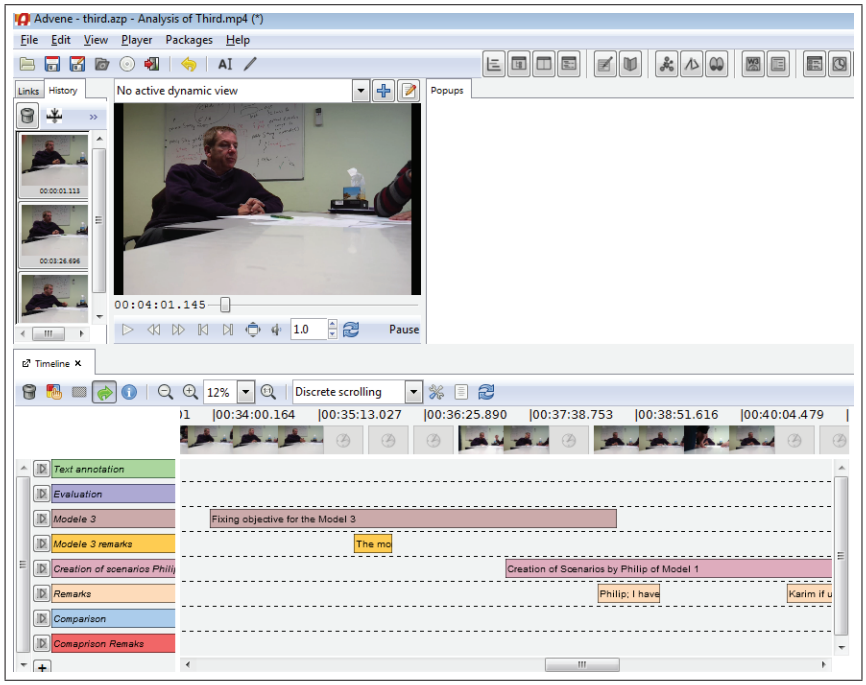


Figure 6.6: ADVENE: A tool for video annotation

Concerning the domain knowledge, we have added a new concept and five new relations between the concepts:

1. The concept Attention is a pre-requisite of the concept Visuo-Spatial.
2. The concept Memory is a pre-requisite of the concept Oral Language.
3. The concept Auditive Perception is a pre-requisite of the concept Oral Language.
4. The concept Visual Perception is a pre-requisite of the concept Written Language.
5. The concept Visual Memory is a pre-requisite of the concept Working-Memory.

[1]http://liris.cnrs.fr/advene

Concerning the functioning of the generator, the level of difficulty of some mini-games set by the generator did not match the level of mini-games set by the expert. The origin of this error was that the algorithm adopted by the generator was only taking into account the learner's profile to set this level, while the expert took into account the difference between the learner's profile and the session objectives.

This evaluation process helped us in validating the correctness of the generated scenarios. The expert was ready to use the scenarios in the project CLES. This experimentation also helped the expert to review the knowledge model and identify some more relations in the domain model, which were previously unidentified. We also detected a problem in the concept selection strategy and duly corrected it.

In the next section, we present another experiment,, which measures the learning impact of the generated scenarios on real-world learners.

6.4 Study of the impact of serious games on learners

In order to answer the second part of the third research question, we have proposed another evaluation protocol and conducted an experimentation in the context of the Project CLES. More precisely, the objective of this evaluation is:

- to verify that the serious game helps the learner in learning a subject/concept 'X' better than the traditional learning tools.
- to verify that the scenarios generated by our system are according to the learner profile.

In the next section 6.4.1, we present the evaluation protocol. We have implemented this protocol as part of an experiment, which we have conducted with learners in a situation of cognitive disabilities. The section 6.4.2 shows the results of this experiment.

6.4.1 Evaluation Protocol

As we want to measure the impact of adaptive learning scenarios on the learning of learners, we have followed the pre-test, post-test approach of [Grubišic 2006]. To conduct this evaluation, we have followed an evaluation protocol, which guides us during the experimentation process.

The flow of this protocol, shown in the figure 6.7, is as follows: we start with a group of learners. We divide these learners into two groups: Group A (Experimental Group) and Group B (Control Group). We give the learners of the two groups a pre-test, which is a questionnaire containing questions of type Multiple Choice Questions (MCQ). Then, the Group A uses the mini-games generated by GOALS and the Group B continues using the traditional application. Afterwards, we present both the Group A and B with a Post-test, which is also a MCQ. The learners, of the two groups, fill a questionnaire to express their feelings about their experience with

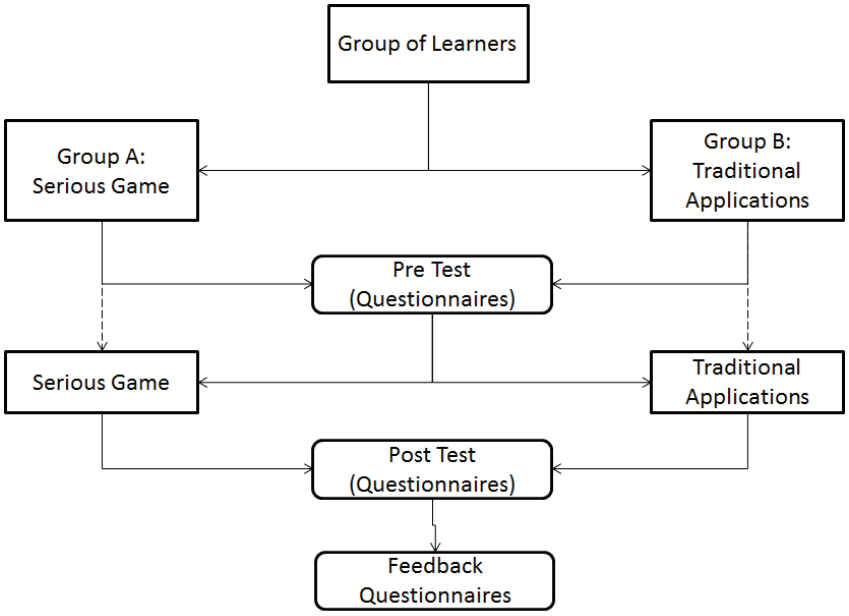


Figure 6.7: Evaluation protocol to study the impact of our system

the experiment. Finally, we compare the results of the pre-test and the post-test to study the learning gain of the serious game.

We followed this protocol to conduct an experiment, presented in the next section.

6.4.2 Experiment and results

We conducted this experiment with persons in a situation of cognitive disabilities. The objective of this experiment, is to identify whether the interaction with the pedagogical scenarios helps the learner or not. The experiment compared two different methods of improving cognitive disabilities. One method is to let a learner interact with the scenarios generated by the platform GOALS, and the other is to use traditional paper-and-pencil method. The two methods are identical in terms of learning objectives i.e. they both try to re-educate certain cognitive abilities of a learner. Consequently, if there are any real differences in the learning outcomes, then these differences could be attributed to the serious game and the pedagogical scenarios generated by GOALS.

We assigned the learners that participated in the experiment into two groups, one of which used the GOALS platform to generate scenarios (Group A) and the other one used the traditional paper-pencil supports (Group B).

The experiment included eight persons suffering from cognitive disabilities. These persons are undergoing therapy in an institution in Lyon, France. Among them, there were 6 boys and 2 girls. Table 6.1 gives an overview of different persons' ages and their cognitive disabilities. All of them possessed basic computer skills (e.g.

Web browsing skills), which was necessary to access the GOALS platform. We have used the knowledge models, created for the project CLES, to generate scenarios.

Person	Age	Disability Situation
Person 1	30	Dysphasie
Person 2	18	Dyspraxie
Person 3	21	Syndrome Asperger
Person 4	21	Physique
Person 5	21	Multi-disabilities
Person 6	16	Multi-disabilities, Attentional Troubles
Person 7	17	Syndrome Asperger
Person 8	26	Epilepsie

Table 6.1: Profile of different persons participating in this experiment

This experiment was focused upon three cognitive functions:

- Perception
- Memory
- Logical reasoning

Therefore, for each of these cognitive functions we have a pre-test, post-test and a test. In the first session, the learners solved a pre-test on paper. In the next session, the learners practised solving the problems of the same cognitive functions. For this, the experimental group (Group A) used the GOALS platform (see the figure 6.8), while the control group (Group B) used traditional paper-and-pencil method. In the last session, the learners solved a post-test on paper.

The pre-test and post-test were a paper based test. Each test contains multiple choice questions (MCQ) regarding a certain cognitive function. Some of questions regarding logical reasoning's pre-test and post-test can be seen in figure 6.9.

The result of the pre-test of the three cognitive functions for both group A and group B can be seen in the table 6.2. We have normalized these results on a scale of ten. The analysis of the pretest scores showed that there was no statistically significant difference in performance of the pre-test between learners of group A and learners of group B, which indicates that the two groups had similar mastery of the three cognitive functions.

After the pre-test, the learners in group A used the GOALS platform. As mentioned earlier, GOALS is an on-line platform; therefore, we created an account for each of the learner and created their respective profiles. Since, we were not aware of the actual profiles of these learners; therefore, we initialized there profiles as a function of their ages (just like in the project CLES). In the meantime, the learners in group B were using traditional support for the cognitive functions.



Figure 6.8: Learners of Group A using the GOALS platform

We video recorded (with their explicit permission) the persons of group A interacting with scenarios generated by GOALS. We analysed this video using ADVENE[2][Aubert 2004, Aubert 2005].

The tables 6.2 show the evolutions of learners' performances for both groups. We can by the results that both the groups showed some increase in the post-test results. The Group A showed a slightly better performance on Perception and Memory then Group B. However, we cannot associate with certainty the increase in performance to the generated scenarios, because the learners interacted for a very small period of time with GOALS. The activities done by the Group A were very similar to that of the post-test. This may be a cause of their better performance.

[2]http://liris.cnrs.fr/advene

Groups	Learners	Perception					
		Pre-Test	Average	Standard Deviation	Post-Test	Average	Standard Deviation
Group A	Person 1	4	6,25	2.63	10	8.875	1.03
	Person 2	5			7.5		
	Person 3	6			9		
	Person 4	10			9		
Group B	Person 5	7	8	1.41	10	8.5	1.22
	Person 6	7			7.5		
	Person 7	8			7.5		
	Person 8	10			9		

(a) Pre-Test and Post-Test results for the function Perception

Groups	Learners	Memory					
		Pre-Test	Average	Standard Deviation	Post-Test	Average	Standard Deviation
Group A	Person 1	3	5.375	2.32	5	6.25	0.96
	Person 2	8.5			6		
	Person 3	5.5			7		
	Person 4	5.5			7		
Group B	Person 5	3	2.25	1.5	5	4.25	2.22
	Person 6	1			2		
	Person 7	1			3		
	Person 8	4			7		

(b) Pre-Test and Post-Test results for the function Memory

Groups	Learners	Logical Reasoning					
		Pre-Test	Average	Standard Deviation	Post-Test	Average	Standard Deviation
Group A	Person 1	0	1.25	1.44	5	7,5	2,90
	Person 2	0			5		
	Person 3	2.5			10		
	Person 4	2.5			10		
Group B	Person 5	0	1.875	1.39	5	7.5	2.90
	Person 6	2.5			5		
	Person 7	5			10		
	Person 8	0			10		

(c) Pre-Test and Post-Test results for the function Logical Reasoning

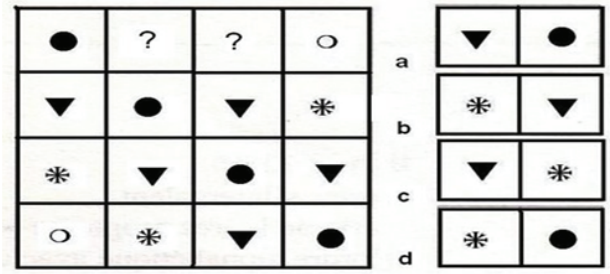
Table 6.2: Pre-Test and Post-Test scores of on all the tests

Test N°1

Il s'agit de séries qui se situent à l'intérieur de figures géométriques, souvent un carré de quadrillé de 4 lignes. Elles peuvent être composées de chiffres, de lettres ou de signes.

Il vous faut découvrir les figures manquantes matérialisées par les points d'interrogations.

1. Les figures manquantes :

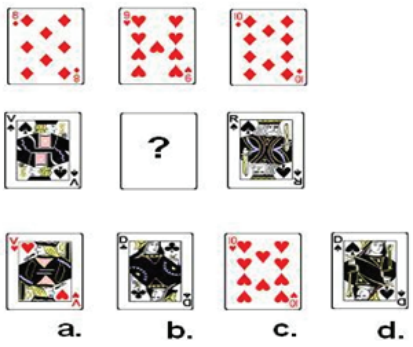


- A
- B
- C
- D

(a) Logical Reasoning's pre-test

Plusieurs cartes sont disposées selon une certaine logique et l'une des cartes est retournée. Vous devez retrouver la logique pour déterminer la carte manquante. Il convient de traiter la valeur et la couleur de la carte séparément.

Test N°1



(b) Logical Reasoning's post-test 9

Figure 6.9: Logical Reasoning tests

Chapter 7

Conclusions and Perspectives

The research took place in the context of adaptive learning systems. We explored the adaptation of the learning experience in serious games. We worked on a project CLES (Cognitive Linguistic Elements Stimulation). The objective of this project was to develop an adaptive serious game, available on-line, for the evaluation and training of cognitive functions of persons with cognitive disabilities. This project aims, on the one hand, to create for each cognitive disorder a mini-game, which targets an aspect of the disorder, while optimizing, through techniques of video games, their cognitive ergonomics. On the other hand, it aims to develop a module to generate, for each patient, personalized paths through the game keeping into account the patient's difficulties and progress. This project considers the following cognitive disorders: perception, attention, memory, oral language, written language, logical reasoning, visuo-spatial and transversal skills.

In this context, the objective of our research was to propose models and processes to allow the generation of pedagogical scenarios that can be used in serious games. By scenario, we mean a suite of pedagogical activities generated by the system for a learner keeping into account the learner's profile to achieve a pedagogical goal in Games-Based Learning Environment. The approach we proposed for scenarios generator took into account two characteristics:

1. **Generic :** The knowledge should be represented in such a way, so that it can be re-utilized with different pedagogical domains and serious games.

2. **Scalable :** The approach should be able to adapt by continuously acquiring knowledge. This means keeping into account the interaction of the learners to update their profiles, adapting the pedagogical scenarios and modifying the domain knowledge.

To archive this research objective, we have identified three research questions. The first question deals with the identification and the representation of the knowledge necessary to adapt the pedagogical scenarios according to a learner. The second question deals with inference process for the exploitation of this knowledge. The third question deals with the evaluation and validation of knowledge and the inference process, and verification of the impact of the pedagogical scenarios on the actual learning.

To respond to the first research question, we proposed different types of knowledge models, which are necessary to generate the pedagogical scenarios. This includes the domain concept knowledge, the pedagogical resource knowledge and the

serious game resource knowledge. In addition, we also proposed the organization of this knowledge in a multilayer architecture to make sure that the knowledge of a given layer remains independent of other layers. This independence allows the approach to be used with different pedagogical domains and different serious games, hence, achieving the generic characteristics. As the objective targets a personalized scenario, we have modelled the learner profile. This profile is used by the inference process to adapt the scenarios. Furthermore, we model and store the learner's interaction with the system. These traces are used for updating the learner profile, and adapting the scenarios. Moreover, the scenario generation also needs other models as well to organize the scenario. These models include the presentation model (used to organize the pedagogical activities) and the adaptation knowledge (used to adapt pedagogical resources according to a learner).

To answer the second question, we proposed an adaptive pedagogical scenario generator. This generator takes into account the pedagogical objectives of the learner and his profile to generate adaptive scenarios. These scenarios get generated in three steps, firstly, all the domain concepts, which are necessary for the learner to achieve his pedagogical goals, get selected. Secondly, for each selected concept, appropriate pedagogical resources get selected for the learner, and then these resources get adapted according to the adaptation knowledge. Thirdly, we associate the pedagogical resources with the serious game resources, so that the learner can interact with them via a serious game.

In order to test the proposed models with real-world pedagogical domains and serious games, we developed the platform GOALS (Generator Of Adaptive Learning Scenarios). In this platform, we allowed course designers the possibility to create knowledge related to a pedagogical domain. They can create and manage the different knowledge models required to generate a scenario, and manage different learners. They can also test the pedagogical scenarios by generating them for a learner and his pedagogical goals. GOALS is an on-line platform based on client/server architecture. The interface of GOALS uses Adobe Flash technologies the core of the platform uses JAVA J2EE technologies, and a My-SQL database stores the data.

We used GOALS to respond to the third question, i.e. we conducted an evaluation with an expert in the context of the project CLES. Thus, we modelled the CLES's knowledge via the proposed models and tested this modelling according to an evaluation protocol. The protocol uses the comparative evalua- tion strategy. The idea is to compare the scenarios generated by GOALS with the scenarios created manually by the learner for the same input (learners' profiles + pedagogical goals). The expert expressed his satisfaction for the generated scenarios in general. This evaluation allowed us to make some improvements to the knowledge of the project CLES.

In order to study the impact of the adaptive learning scenario on the learners, we conducted an experimentation with real-world learners in the context of the project CLES. For this experiment, we divided the learners in an experimental group and a control group. Afterwards, we asked them to conduct a pre-test on some cognitive functions. Next, the control group trained with traditional learning techniques and

the experimental group used the GOALS platform. Finally, all of the learners had a post-test. We compared the performance of learners in the pre-tests and post-tests to measure the learning gain.

The main difficulty we could have faced while following the evaluation protocol was to identify whether the problem exists in the expert's knowledge introduction into the system or in the generation of the scenario, when the expert expresses dissatisfaction with the generated scenarios. Moreover, there is also the possibility that the problem exists in both the expert's knowledge introduction and the scenario generator. However, we did not face this problem as we were fortunate enough in pinpointing the problem. However, we can face this problem with future evaluations.

Another difficulty we faced while designing the interface for the visual creation of the domain knowledge in GOALS was to manage the large amount of elements. In CLES, there are about 41 domain concepts, 44 relations between the concepts, 91 mini-games, and many serious game resources, however, the screen area to display all these elements is quite limited. Consequently, when we try to display all the elements simultaneously the knowledge graph difficult to visualize. To counter this problem, when the users first visualize the domain knowledge, we only show the concepts that are not sub-concepts of any other concept. The user can click, on a concept, to show the sub-concepts of that concept. In this way, the amount of knowledge displayed is limited, hence, easy to visualize and manipulate.

7.1 Perspectives

In the current state of the generator, we provide the adaptation before the beginning of the gaming session. However, we do not adapt the pedagogical scenarios in real-time i.e during the gaming session. For our future works, we would like to personalize the interaction i.e. to be able to adapt the scenarios in real-time. This requires methods that allows to analyse the learner's interaction traces to detect the cases of incoherence (pedagogical activities that are maladapted to the learner's situation) and modify the scenario, during the interaction, according to the learner's performance. We can adjust the difficulty levels of the pedagogical resources, select different pedagogical resources or select another conceptual path.

For our future works, we would also like to make use of the learner's interaction traces in an "off-line" manner. These traces contain the learner's interaction knowledge. We can use this knowledge to propose, the domain experts, possible modifications in the knowledge models. We can propose to adjust the difficulty levels of pedagogical resources, discover new concepts, combine similar concepts and add a relation between a previously unrelated pedagogical resource and domain concept. This process is semi-automatic i.e. we will only propose the modifications to the expert and let him decide whether to make the modifications or not. We propose to use machine learning techniques to extract this knowledge from the traces. We can use unsupervised learning (clustering) to data-mine the traces. We can show the expert an interface, which will help him in setting the parameters of

the machine-learning method. The parameters include the clustering method, the distance calculation method and the number of clusters.

We would also like to study the impact of adaptive learning over a long period of time. This would require to conduct a series of tests over a period of time with real-world learners.

Bibliography

[Ahanger 1997] G Ahanger and T.D.C. Little. *Easy Ed: An integration of technologies for multimedia education.* In Proc. of WebNet, volume 97, pages 15–20. Citeseer, 1997. (Cited on page 34.)

[Ahmad 2007] Faisal Ahmad, S. de la Chica, Kirsten Butcher, Tamara Sumner and J.H. Martin. *Towards automatic conceptual personalization tools.* In Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, pages 452–461. ACM, 2007. (Cited on page 47.)

[Albert 2009] Dietrich Albert, Alexander Nussbaumer, C.M. Steiner, M. Hendrix and Alexandra Cristea. *Design and Development of an Authoring Tool for Pedagogical Relationship Types between Concepts.* In International Conference on Computers in Education (ICCE 2009), pages 194–196, 2009. (Cited on pages 43 and 48.)

[Aldrich 2005] Clark Aldrich. Learning by doing: A comprehensive guide to simulations, computer games, and pedagogy in e-learning and other educational experiences. Numeéro 3. Pfeiffer & Co, 2005. (Cited on page 3.)

[Amoia 2012] Marilisa Amoia, Treveur BRETAUDIERE, Alexandre DENIS, Claire GARDENT and Laura PEREZ-BELTRACHINI. *A Serious Game for Second Language Acquisition in a Virtual Environment.* JSCI, Journal on Systemics, Cybernetics and Informatics, vol. 10, no. 1, 2012. (Cited on page 19.)

[Anderson 2004] John R. Anderson. Cognitive Psychology and its Implications. Worth Publishers, 6th édition, 2004. (Cited on page 107.)

[Arruabarrena 2006] Rosa Arruabarrena, Tomás A Pérez, J. López-Cuadrado, J. Gutiérrez and J.A. Vadillo. *On evaluating adaptive systems for education.* AH2002, 2nd. International Conference on Adaptive Hypermedia and Adaptive Web Based Systems. 2002, vol. 2347, pages 363–367, 2006. (Cited on page 123.)

[Astle 2009] Duncan Astle and G Scerif. *Using developmental cognitive neuroscience to study behavioral and attentional control.* Developmental Psychobiology, vol. 57, pages 107 – 118, 2009. (Cited on page 108.)

[Aubert 2004] Olivier Aubert, PA Champin and Y Prié. *The advene model for hypervideo document engineering.* Research Report RR-2004022, LIRIS, pages 1–19, 2004. (Cited on pages 134 and 138.)

[Aubert 2005] Olivier Aubert and Yannick Prié. *Advene: active reading through hypervideo.* Proceedings of the sixteenth ACM conference on . . . , 2005. (Cited on pages 134 and 138.)

[Bakkes 2008] Sander Bakkes, Pieter Spronck and Jaap van Den Herik. *Rapid adaptation of video game AI.* 2008 IEEE Symposium On Computational Intelligence and Games, pages 79–86, December 2008. (Cited on page 25.)

[Baldoni 2005] Matteo Baldoni, Cristina Baroglio and Nicola Henze. *Personalization for the semantic web.* Reasoning Web, vol. 3564/2005, no. 95, pages 173–212, 2005. (Cited on page 15.)

[Barr 1976] Avron Barr, Marian Beard and RC Atkinson. *The computer as a tutorial laboratory: the Stanford BIP project.* International Journal of Man-Machine Studies, vol. 8, no. 5, pages 567–582, 1976. (Cited on page 29.)

[Beal 2010] Carole R. Beal, Ivon Arroyo and Paurl R. Cohen. *Evaluation of AnimalWatch: An intelligent tutoring system for arithmetic and fractions.* Journal of Interactive . . . , 2010. (Cited on page 126.)

[Bergeron 2006] Bryan Bergeron. Developing Serious Games. Cengage Learning, 2006. (Cited on page 17.)

[Bibeau 2004] Robert Bibeau. *Scénarios pédagogiques, propositions éducatives, activités d'apprentissage avec les TIC.* Rapport technique, 2004. (Cited on page 16.)

[Bieliková 2006] Mária Bieliková. *An adaptive web-based system for learning programming.* International Journal of Continuing Engineering Education and Life Long Learning, vol. 16, no. 1, pages 122–136, 2006. (Cited on page 47.)

[Bieliková 2008] Mária Bieliková, Marko Divéky, Peter Jurnečka, Rudolf Kajan and L. Omelina. *Automatic generation of adaptive, educational and multimedia computer games.* Signal, Image and Video Processing, vol. 2, no. 4, pages 371–384, 2008. (Cited on pages 21, 27 and 43.)

[Bikovska 2007] Jana Bikovska and Galina Merkuryeva. *Scenario-based planning and management of simulation game: a review.* In 21st European Conference on Modelling and Simulation, volume 4, 2007. (Cited on pages 23 and 27.)

[Botella 2000] C Botella, R Banos, H Villa, C Perpina and A Garciapalacios. *Virtual reality in the treatment of claustrophobic fear: A controlled, multiple-baseline design.* Behavior Therapy, vol. 31, no. 3, pages 583–595, 2000. (Cited on pages 4 and 104.)

[Bouzeghoub 2005] Amel Bouzeghoub, Claire Carpentier, Bruno Defude and Freddy Duitama. *A model of reusable educational components for the generation of adaptive courses.* In Proc. First International Workshop on Semantic Web for Web-Based Learning in conjunction with CAISE, volume 3. Citeseer, 2005. (Cited on pages 35, 36 and 40.)

[Bra 1998] Paul De Bra and Licia Calvi. *AHA! An open adaptive hypermedia architecture.* New Review of Hypermedia and Multimedia, pages 1–18, 1998. (Cited on pages 31 and 40.)

[Bra 2001] Paul De Bra and JP Ruiter. *AHA! Adaptive hypermedia for all.* In proceedings of WebNet, 2001. (Cited on pages 31 and 40.)

[Broomfield 2004] Jan Broomfield and Barbara Dodd. *Children with speech and language disability: caseload characteristics.* International journal of language & communication disorders / Royal College of Speech & Language Therapists, vol. 39, no. 3, pages 303–24, 2004. (Cited on page 104.)

[Brown 2009] D.J. Brown, Nicholas Shopland, Steven Battersby, Alex Tully and S. Richardson. *Game On: accessible serious games for offenders and those at risk of offending.* Journal of Assistive Technologies, vol. 3, no. 2, pages 13–25, 2009. (Cited on pages 23 and 27.)

[Brusilovsky 1992] Peter Brusilovsky. *A framework for intelligent knowledge sequencing and task sequencing.* Intelligent tutoring systems, pages 499–506, 1992. (Cited on page 29.)

[Brusilovsky 1993] Peter Brusilovsky. *Task sequencing in an intelligent learning environment for calculus.* In Seventh International PEG Conference, pages 57–62, 1993. (Cited on page 29.)

[Brusilovsky 1994] Peter Brusilovsky. *ILEARN: an intelligent system for teaching and learning about UNIX.* In Proc. of SUUG International Open Systems Conference, Moscow, Russia, ICSTI, pages 35–41, 1994. (Cited on page 29.)

[Brusilovsky 1996] Peter Brusilovsky, E Schwarz and Gerhard Weber. *A tool for developing hypermedia-based ITS on WWW.* In Intelligent Tutoring Systems - Proceedings of the Third International Conference, ITS '96, pages 261–269, 1996. (Cited on page 32.)

[Brusilovsky 2001a] Peter Brusilovsky. *Adaptive hypermedia.* User modeling and user-adapted interaction, vol. 11, no. 1, pages 87–110, 2001. (Cited on page 28.)

[Brusilovsky 2001b] Peter Brusilovsky and Charalampos Karagiannidis. *The benefits of layered evaluation of adaptive applications and services.* Evaluation of Adaptive, 2001. (Cited on page 123.)

[Brusilovsky 2003a] Peter Brusilovsky and Julita Vassileva. *Course sequencing techniques for large-scale web-based education.* International Journal of Continuing Engineering Education and Lifelong Learning, vol. 13, no. 1/2, pages 75–94, 2003. (Cited on pages 29 and 46.)

[Brusilovsky 2003b] Peter Brusilovsky and Julita Vassileva. *Course sequencing techniques for large-scale web-based education.* Engineering Education and Lifelong Learning, vol. 13, no. 1/2, pages 75–94, 2003. (Cited on page 58.)

[Brusilovsky 2007] Peter Brusilovsky and Eva Millán. *User models for adaptive hypermedia and adaptive educational systems.* Lecture Notes in Computer Science, vol. 4321, page 3, 2007. (Cited on page 54.)

[Bull 1970] George G Bull. *The Elicitation Interview.* Studies in Intelligence, vol. 14, no. 2, pages 115–122, 1970. (Cited on page 128.)

[Burgos 2008] Daniel Burgos and Pablo Moreno-Ger. *Building adaptive game-based learning resources: The integration of IMS Learning Design and< e-Adventure>.* Simulation & . . . , pages 1–12, 2008. (Cited on pages 21 and 43.)

[Campos 2004] Eduardo Campos, Ana Granados, Sergio Jiménez and Javier Garrido. *Tutor Informatico: Increasing the Selfteaching in Down Syndrome People.* Computers Helping People with Special Needs, pages 629–629, 2004. (Cited on page 104.)

[Capell 1993] Peter Capell and Roger B Dannenberg. *Instructional design and intelligent tutoring: Theory and the precision of design.* Journal of Artificial Intelligence in Education, vol. 4, no. 1, pages 95–121, 1993. (Cited on page 29.)

[Capuano 2002] Nicola Capuano, Matteo Gaeta, Alessandro Micarelli and Enver Sangineto. *An integrated architecture for automatic course generation.* In Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 02), numéro Section 4, pages 322–326. Citeseer, 2002. (Cited on pages 36 and 40.)

[Carro 2003] Rosa María Carro, Alvaro Ortigosa, E Mart\'\in and Johann Schlichter. *Dynamic generation of adaptive web-based collaborative courses.* Groupware: Design, Implementation, and Use, pages 191–198, 2003. (Cited on pages 37 and 40.)

[Carro 2006] Rosa María Carro, Ana M. Breda, Gladys Castillo and Antonio L. Bajuelos. *A methodology for developing adaptive educational-game environments.* In Adaptive Hypermedia and Adaptive Web-Based Systems, pages 90–99. Springer, 2006. (Cited on pages 21 and 27.)

[Carron 2007] Thibault Carron, Jean-Charles Marty and Jean-Mathias Heraud. *Teaching with game-based learning management systems: Exploring a pedagogical dungeon.* Simulation & Gaming, vol. 39, no. 3, pages 353–378, July 2007. (Cited on pages 23, 27 and 43.)

[Castel 2005] Alan D Castel, Jay Pratt and Emily Drummond. *The effects of action video game experience on the time course of inhibition of return and the*

efficiency of visual search. Acta psychologica, vol. 119, no. 2, pages 217–30, June 2005. (Cited on page 4.)

[Caumanns 1998] J. Caumanns. *A bottom-up approach to multimedia teachware.* In Intelligent Tutoring Systems, pages 116–125. Springer, 1998. (Cited on page 34.)

[Chang 2008] Wen-Cih Chang and Yu-Min Chou. *Introductory C Programming Language Learning with Game-Based Digital Learning.* Advances in Web Based Learning-ICWL 2008, pages 221 – 231, 2008. (Cited on pages 23 and 27.)

[Cho 2002] Baek-Hwan Cho, Jeonghun Ku, Dong Pyo Jang, Saebyul Kim, Yong Hee Lee, In Young Kim, Jang Han Lee and Sun I Kim. *The effect of virtual reality cognitive training for attention enhancement.* Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society, vol. 5, no. 2, pages 129–37, April 2002. (Cited on page 4.)

[Clauzel 2011] Damien Clauzel, Karim Sehaba and Yannick Prié. *Enhancing synchronous collaboration by using interactive visualisation of modelled traces.* Simulation Modelling Practice and Theory, vol. 19, no. 1, pages 84–97, January 2011. (Cited on page 55.)

[Committee 2002] IEEE Learning Technology Standards Committee. *Draft standard for learning object metadata.* Rapport technique July, 2002. (Cited on page 50.)

[Conde 2009] Angel Conde, K. de Ipiña, Mikel Larrañaga, N. Garay-Vitoria, E. Irigoyen, A. Ezeiza and J. Rubio. *LAGUNTXO: a rule-based intelligent tutoring system oriented to people with intellectual disabilities.* Visioning and Engineering the Knowledge Society. A Web Science Perspective, pages 186–195, 2009. (Cited on page 104.)

[Cooley 1976] William W Cooley and Paul R. Lohnes. Evaluation research in education. New York : Irvington Publishers : distributed by Halsted Press, 1976. (Cited on page 122.)

[Coyne 2003] Richard Coyne. *Mindless repetition: Learning from computer games.* Design Studies, vol. 24, no. 3, pages 199–212, 2003. (Cited on page 3.)

[Cristea 2003] Alexandra Cristea and Arnout de Mooij. *LAOS: Layered WWW AHS authoring model and their corresponding algebraic operators.* WWW03 (The Twelfth International World Wide . . . , 2003. (Cited on page 43.)

[Crowley 2007] Rebecca S. Crowley, Elizabeth Legowski, Olga Medvedeva, Eugene Tseytlin, Ellen Roh and Drazen Jukic. *Evaluation of an intelligent tutoring system in pathology: Effects of external representation on performance gains, metacognition, and acceptance.* JAMIA, pages 182–190, 2007. (Cited on page 126.)

[Dagger 2005] Declan Dagger, Vincent Wade and Owen Conlan. *Personalisation for all: Making adaptive course composition easy.* Educational Technology \& Society, vol. 8, no. 3, pages 9–25, 2005. (Cited on pages 43 and 47.)

[De Bra 1999] Paul De Bra, G.J. Houben and Hongjing Wu. *AHAM: a Dexter-based reference model for adaptive hypermedia.* In Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots, page 156, New York, New York, USA, 1999. ACM. (Cited on pages 43 and 46.)

[De Bra 2002] Paul De Bra, Ad Aerts and Brendan Rousseau. *Concept relationship Types for AHA! 2.0.* In Proceedings of the AACE ELearn'2002 conference, pages 1386–1389. Citeseer, 2002. (Cited on page 31.)

[De Bra 2006] Paul De Bra, David Smits and Natalia Stash. *Creating and delivering adaptive courses with AHA!* Innovative Approaches for Learning and ..., 2006. (Cited on page 31.)

[De Lisia 2002] Richard De Lisia and Jennifer L. Wolford. *Improving Children's Mental Rotation Accuracy With Computer Game Playing.* The Journal of Genetic Psychology: Research and Theory on Human Development, vol. 163, no. 3, pages 272–282, 2002. (Cited on page 3.)

[De-Marcos 2008] Luis De-Marcos, José-Javier Martínez and José-Antonio Gutiérrez. *Particle Swarms for Competency-Based Curriculum Sequencing.* Emerging Technologies and Information Systems for the Knowledge Society, pages 243–252, 2008. (Cited on page 34.)

[Delozanne 2008] Élisabeth Delozanne, Dominique Prévit, Brigitte Grugeon and F. Chenevotot. *Automatic Multi-criteria Assessment of Open-Ended Questions: A Case Study in School Algebra.* In Intelligent Tutoring Systems, pages 101–110. Springer, 2008. (Cited on page 125.)

[Diamond 1989] A Diamond and P.S. Goldman-Rakic. *Comparison of human infants and rhesus monkeys on Piaget's AB task: Evidence for dependence on dorsolateral prefrontal cortex.* Experimental Brain Research, vol. 74, no. 1, pages 24–40, 1989. (Cited on page 104.)

[Dijkstra 1974] EW Dijkstra. *On the role of scientific thought.* pages 60–66, 1974. (Cited on page 45.)

[Drivera 1991] Jon Drivera and Peter W Halligan. *Can Visual Neglect Operate in Object-centred Co-ordinates? An Affirmative Single-case Study.* Cognitive Neuropsychology, vol. 8, no. 6, pages 475 – 496, 1991. (Cited on page 4.)

[Duitama 2005] Freddy Duitama, Bruno Defude, Amel Bouzeghoub and C. Lecocq. *A framework for the generation of adaptive courses based on semantic metadata.* Multimedia Tools and Applications, vol. 25, no. 3, pages 377–390, 2005. (Cited on pages 35, 36, 40, 43, 47 and 48.)

[Dung 2010] Tran Chi Dung, Sébastien George and Iza Marfisi-Schottman. *EDoS: An authoring environment for serious games design based on three models.* In 4th Europeen Conference on Games Based Learning ECGBL2010, numéro October, pages 393–402, 2010. (Cited on pages 22 and 43.)

[Eliot 1997] Christopher Rhodes Eliot, Daniel E. Neiman and Michelle M. Lamar. *Medtec: A Web-Based Intelligent Tutor for Basic Anatomy.* In Proc. of WebNet'97, World Conference of the WWW, Internet and Intranet, pages 167–165, 1997. (Cited on page 29.)

[ELSPA 2006] ELSPA. *Unlimited learning Computer and video games in the learning landscape.* Rapport technique, 2006. (Cited on page 3.)

[Emin 2008] Valérie Emin, Jean-Philippe Pernin and Viviane Guéraud. *Goal-oriented authoring approach and design of learning systems.* Advances in Conceptual Modeling – Challenges and Opportunities, vol. 5232, no. Lecture Notes in Computer Science, pages 292 – 301, 2008. (Cited on page 17.)

[Enochsson 2004] Lars Enochsson, Bengt Isaksson, René Tour, Ann Kjellin, Leif Hedman, Torsten Wredmark and Li Tsai-Felländer. *Visuospatial skills and computer game experience influence the performance of virtual endoscopy.* Journal of gastrointestinal surgery : official journal of the Society for Surgery of the Alimentary Tract, vol. 8, no. 7, pages 876–82; discussion 882, November 2004. (Cited on pages 3 and 4.)

[Farrell 2004] Robert G. Farrell, Soyini D. Liburd and John C. Thomas. *Dynamic assembly of learning objects.* Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters - WWW Alt. '04, vol. 01, no. 914, page 162, 2004. (Cited on page 47.)

[Felder 1988] Richard M. Felder and Linda K. Silverman. *Learning and Teaching Styles in Engineering Education.* Engineering Education, vol. 78, no. 7, pages 674–681, 1988. (Cited on page 55.)

[Ferguson 2007] Christopher J. Ferguson, Amanda M. Cruz and Stephanie M. Rueda. *Gender, Video Game Playing Habits and Visual Memory Tasks.* Sex Roles, vol. 58, no. 3-4, pages 279–286, October 2007. (Cited on page 4.)

[Fischer 2001] Stephan Fischer. *Course and exercise sequencing using metadata in adaptive hypermedia learning systems.* Journal on Educational Resources in Computing (JERIC), vol. 1, no. 1es, page 5, 2001. (Cited on pages 47 and 124.)

[Fossati 2008] Davide Fossati, B. Di Eugenio, Christopher Brown and S. Ohlsson. *Learning linked lists: Experiments with the iList system.* In Intelligent Tutoring Systems, pages 80–89. Springer, 2008. (Cited on page 126.)

[Fu 2009] Fong-Ling Fu, Rong-Chang Su and Sheng-Chin Yu. *EGameFlow: A scale to measure learners' enjoyment of e-learning games.* Computers & Education, vol. 52, no. 1, pages 101–112, January 2009. (Cited on page 19.)

[Gena 2005] Cristina Gena. *Methods and techniques for the evaluation of user-adaptive systems.* The Knowledge Engineering Review, vol. 20, no. 01, page 1, December 2005. (Cited on page 123.)

[George 2010] Sébastien George. *Interactions et communications contextuelles dans les environnements informatiques pour l'apprentissage humain.* PhD thesis, Institut National des Sciences Appliquées de Lyon et Université Claude Bernard Lyon 1, 2010. (Cited on pages 18 and 19.)

[Germanakos 2006] Panagiotis Germanakos and Constantinos Mourlas. *Adaptation and Personalization of Web-Based Multimedia Content.* In & S. Chen G. Ghinea, editeur, Digital Multimedia Perception and Design, volume 29, chapitre 014, pages 284–304. Hershey, PA: Idea Group Publishing, 2006. (Cited on page 15.)

[Green 2003] C Shawn Green and Daphne Bavelier. *Action video game modifies visual selective attention.* Nature, vol. 423, no. 6939, pages 534–7, May 2003. (Cited on page 4.)

[Green 2010] C. Shawn Green, Renjie Li and Daphne Bavelier. *Perceptual Learning During Action Video Game Playing.* Topics in Cognitive Science, vol. 2, no. 2, pages 202–216, April 2010. (Cited on page 4.)

[Grubišic 2006] Ani Grubišic, Slavomir Stankov and Branko Žitko. *An approach to automatic evaluation of educational influence.* Proceedings of the 6th WSEAS . . . , 2006. (Cited on pages 125 and 135.)

[Guéraud 2006] Viviane Guéraud and Jean-Michel Cagnat. *Automatic semantic activity monitoring of distance learners guided by pedagogical scenarios.* In EC-TEL'06 Proceedings of the First European conference on Technology Enhanced Learning: innovative Approaches for Learning and Knowledge Sharing, pages 476 – 481, 2006. (Cited on page 16.)

[Henze 2004] Nicola Henze and Wolfgang Nejdl. *A logical characterization of adaptive educational hypermedia.* New Review of Hypermedia and Multimedia, vol. 10, no. 1, pages 77–113, June 2004. (Cited on page 28.)

[Heraud 2000] JM Heraud and Alain Mille. *Pixed: vers le partage et la réutilisation d'expériences pour assister l'apprentissage.* Proceedings of international symposium TICE, 2000. (Cited on pages 32, 33 and 40.)

[Heraud 2004] Jean-mathias Heraud, Laure France and Alain Mille. *Pixed : An ITS that guides students with the help of learners ' interaction logs.* In 7th International Conference on Intelligent Tutoring Systems, pages 57—-64. 2004. (Cited on pages 32, 33 and 40.)

[Hodhod 2009] Rania Hodhod, Daniel Kudenko and Paul Cairns. *Serious Games to Teach Ethics*. In proceedings of AISB, volume 9, pages 6–9, 2009. (Cited on pages 23, 24 and 27.)

[Hsiao 2008] I-han Hsiao, Peter Brusilovsky and Sergey Sosnovsky. *Web-based parameterized questions for object-oriented programming*. World Conf. on ELearning in . . . , 2008. (Cited on page 124.)

[Hsiao 2009] I-han Hsiao, Sergey Sosnovsky and Peter Brusilovsky. *Adaptive navigation support for parameterized questions in object-oriented programming*. Learning in the Synergy of Multiple . . . , pages 88–98, 2009. (Cited on page 124.)

[Hsieh 2010] Tung-Cheng Hsieh and Tzone-I Wang. *A mining-based approach on discovering courses pattern for constructing suitable learning path*. Expert Systems with Applications, vol. 37, no. 6, pages 4156–4167, June 2010. (Cited on page 34.)

[Huang 2008a] Tien-Chi Huang, Yueh-Min Huang and Shu-Chen Cheng. *Automatic and interactive e-Learning auxiliary material generation utilizing particle swarm optimization*. Expert Systems with Applications, vol. 35, no. 4, pages 2113–2122, November 2008. (Cited on page 124.)

[Huang 2008b] Yueh-Min Huang, Juei-Nan Chen, Tien-Chi Huang, Yu-Lin Jeng and Yen-Hung Kuo. *Standardized course generation process using Dynamic Fuzzy Petri Nets*. Expert Systems with Applications, vol. 34, no. 1, pages 72–86, January 2008. (Cited on page 34.)

[Hunicke 2004] Robin Hunicke and Vernell Chapman. *AI for dynamic difficulty adjustment in games*. In Challenges in Game Artificial Intelligence AAAI Workshop, pages 91–96, 2004. (Cited on pages 24, 25 and 27.)

[Hussaan 2011] AM Hussaan, Karim Sehaba and Alain Mille. *Helping children with cognitive disabilities through serious games: project CLES*. In The proceedings of the 13th . . . , pages 2–3, 2011. (Cited on page 5.)

[Idris 2009] Norsham Idris, Norazah Yusof and Puteh Saad. *Adaptive Course Sequencing for Personalization of Learning Path Using Neural Network*. Int. J. Advance. Soft Comput. Appl, vol. 1, no. 1, 2009. (Cited on page 34.)

[Jennings-Teats 2010] Martin Jennings-Teats, Gillian Smith and Noah Wardip-Fruin. *Polymorph: dynamic difficulty adjustment through level generation*. In PCGames '10 Proceedings of the 2010 Workshop on Procedural Content Generation in Games, 2010. (Cited on pages 24, 25 and 27.)

[Kalloo 2011] Vani Kalloo and Permanand Mohan. *Correlation between Student Performance and Use of an mLearning Application for High School Mathematics*. 2011 IEEE 11th International Conference on Advanced Learning Technologies, pages 174–178, July 2011. (Cited on page 126.)

[Karampiperis 2005a] Pythagoras Karampiperis and Demetrios Sampson. *Adaptive learning resources sequencing in educational hypermedia systems.* Educational Technology & Society, vol. 8, no. 4, pages 128–147, 2005. (Cited on pages 34, 40, 48 and 124.)

[Karampiperis 2005b] Pythagoras Karampiperis and Demetrios Sampson. *Adaptive learning resources sequencing in educational hypermedia systems.* Educational Technology & Society, vol. 8, no. 4, pages 128–147, 2005. (Cited on page 43.)

[Karampiperis 2005c] Pythagoras Karampiperis and Demetrios Sampson. *Designing learning services for open learning systems utilizing IMS learning design.* In Proceedings of the IASTED International Conference WEB-BASED EDUCATION, pages 279–284, 2005. (Cited on pages 34, 35, 40, 43 and 124.)

[Keenoy 2004] Kevin Keenoy, Mark Levene and Don Peterson. *Personalisation and Trails in Self e-Learning Networks, project: SeLeNe–Self E-Learning Networks.* Deliverable, pages 1–51, 2004. (Cited on pages 38 and 40.)

[Kettel 2000] Lori Kettel, Judi Thomson and Jim Greer. *Generating individualized hypermedia applications.* In Proceedings of ITS-2000 workshop on adaptive and intelligent webbased education systems, pages 28–36, 2000. (Cited on page 34.)

[Khuwaja 1996] Ramzan Khuwaja, Michel Desmarais and Richard Cheng. *Intelligent Guide: Combining user knowledge assessment with pedagogical guidance.* In G. Gauthier C. Frasson and A. Lesgold, editeurs, Intelligent Tutoring Systems, Lecture Notes in Computer Science, volume 1086, pages 225–233. Springer-Verlag, Berlin, 1996. (Cited on page 29.)

[Kickmeier-Rust 2006] Michael D Kickmeier-Rust, Daniel Schwarz and Dietrich Albert. *The ELEKTRA project: Towards a new learning experience.* In M3 – INTERDISCIPLINARY ASPECTS ON DIGITAL MEDIA & EDUCATION, 2006. (Cited on page 22.)

[Kirriemuir 2004] John Kirriemuir and Angela McFarlane. *Literature review in games and learning.* Nesta Futurelab, 2004. (Cited on page 3.)

[Klopfer 2009] Eric Klopfer, Scot Osterweil and Katie Salen. *Moving learning games forward.* Rapport technique, 2009. (Cited on page 3.)

[Knutova 2009] Evgeny Knutova, Paul De Bra and Mykola Pechenizkiya. *AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques.* New Review of Hypermedia and Multimedia, vol. 15, no. 1, pages 5–38, 2009. (Cited on page 28.)

[Kontopoulos 2008] E Kontopoulos, D Vrakas, F Kokkoras and N. *An ontology-based planning system for e-course generation.* Expert Systems with Applications:

An International Journal, vol. 35, no. 1-2, pages 398–406, July 2008. (Cited on page 43.)

[Koper 2000] EJR Koper. *From change to renewal: Educational technology foundations of electronic learning environments, Inaugural address*. Open University of the Netherlands, Heerlen, pages 1–41, 2000. (Cited on page 50.)

[Kravcik 2004a] Milos Kravcik and Marcus Specht. *Flexible navigation support in the winds learning environment for architecture and design*. In Third International Adaptive Hypermedia and Adaptive Webbased Systems Conference,, volume 3137, pages 156–165, 2004. (Cited on pages 34 and 40.)

[Kravcik 2004b] Milos Kravcik, Marcus Specht and Reinhard Oppermann. *Evaluation of WINDS authoring environment*. In Adaptive Hypermedia and Adaptive Web-Based Systems, pages 166–175. Springer, 2004. (Cited on page 123.)

[Law 2008] E.L.C. Law and M. Rust-Kickmeier. *80Days: Immersive digital educational games with adaptive storytelling*. In First International Workshop on Story-Telling and Educational Games (STEG¿ 08), Maastricht, The Netherlands, numéro iii, 2008. (Cited on page 3.)

[Leinhardt 1998] G Leinhardt. *Situated knowledge and expertise in teaching*. Teachers' Professional Training, pages 146 – 168, 1998. (Cited on page 46.)

[Lepp 2008] Marina Lepp. *How Does an Intelligent Learning Environment with Novel Design Affect the Students' Learning Results?* In Beverley P. Woolf, Esma Aïmeur, Roger Nkambou and Susanne Lajoie, editeurs, ITS '08 Proceedings of the 9th international conference on Intelligent Tutoring Systems, volume 5091 of *Lecture Notes in Computer Science*, pages 70–79, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. (Cited on page 126.)

[Liao 2011] Wen-Wei Liao and Rong-Guey Ho. *Applying Observational Learning in the Cloud Education System of Art Education in an Elementary School*. 2011 IEEE 11th International Conference on Advanced Learning Technologies, pages 131–135, July 2011. (Cited on page 126.)

[Libbrecht 2001a] Paul Libbrecht, Erica Melis and Carsten Ullrich. *Generating personalized documents using a presentation planner*. In ED-MEDIA 2001-World Conference on Educational Multimedia, Hypermedia and Telecommunications, 2001. MEDIA, 2001. (Cited on page 34.)

[Libbrecht 2001b] Paul Libbrecht, Erica Melis and Carsten Ullrich. *Generating personalized documents using a presentation planner*. Proceedings of World Conference . . . , 2001. (Cited on page 40.)

[Limongelli 2008] Carla Limongelli, Filippo Sciarrone and Giulia Vaste. *LS-Plan: An Effective Combination of Dynamic Courseware Generation and Learning*

Styles in Web-Based Education. In Adaptive Hypermedia and Adaptive Web-Based Systems: 5th International Conference, AH 2008, Hannover, Germany, July 29-August 1, 2008, Proceedings, page 133. Springer-Verlag New York Inc, 2008. (Cited on page 37.)

[Littman 1988] D. Littman and E. Soloway. *Evaluating ITSs: The cognitive science perspective.* In M.C. Polson and & J.J. Richardson, editeurs, Foundations of intelligent tutoring systems. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1988. (Cited on page 122.)

[Liu 2010] Ming Liu, R. Calvo and Vasile Rus. *Automatic question generation for literature review writing support.* Intelligent Tutoring Systems, pages 45–54, 2010. (Cited on page 125.)

[Lo 2008] JJ Lo, NW Ji, YH Syu, WJ You and YT Chen. *Developing a digital game-based situated learning system for ocean ecology.* Transactions on edutainment I, no. 2006, pages 51–61, 2008. (Cited on pages 23 and 27.)

[Malan 1999] Ruth Malan and Dana Bredemeyer. *Functional requirements and use cases.* Rapport technique, 1999. (Cited on page 76.)

[Manly 2001] Tom Manly, Vicki Anderson and Ian Nimmo-Smith. *The differential assessment of children's attention: The Test of Everyday Attention for Children (TEA-Ch), normative sample and ADHD performance.* Journal of Child Psychology and Psychiatry, vol. 42, no. 8, pages 1065–1081, November 2001. (Cited on pages 4 and 104.)

[Mark 1993] M.A. Mark and J.E. Greer. *Evaluation methodologies for intelligent tutoring systems.* Journal of Artificial Intelligence in Education, vol. 4, no. 306, pages 129–129, 1993. (Cited on page 122.)

[Martín-Gutiérrez 2010] Jorge Martín-Gutiérrez, Manuel Contero and Mariano Alcañiz. *Evaluating the usability of an augmented reality based educational application.* Intelligent Tutoring Systems, pages 296–306, 2010. (Cited on page 126.)

[Masthoff 2002] Judith Masthoff. *Automatic generation of a navigation structure for adaptive web-based instruction.* In Adaptive Systems for Web-based Education. Citeseer, 2002. (Cited on pages 34 and 125.)

[Masthoff 2003] Judith Masthoff. *The evaluation of adaptive systems.* Adaptive evolutionary information systems, pages 329–347, 2003. (Cited on page 123.)

[McArthur 1988] David McArthur, Cathy Stasz, John Hotta, Oril Peter and Christopher Burdorf. *Skill-oriented task sequencing in an intelligent tutor for basic algebra.* Instructional Science, vol. 17, no. 4, pages 281 – 307, 1988. (Cited on page 29.)

[McNamara 2010] Danielle S McNamara, G. Tanner Jackson and Art Graesser. *Intelligent Tutoring and Games (/ TaG).* In Gaming for Classroom-Based Learning: Digital Role Playing as a Motivator of Study, chapitre 003. 2010. (Cited on pages 20 and 43.)

[Melis 2001] Erica Melis, E. Andres, J. B\\"udenbender, Adrian Frischauf, George Goguadze, Paul Libbrecht, Martin Pollet and Carsten Ullrich. *ActiveMath: A generic and adaptive web-based learning environment.* International Journal of Artificial Intelligence in Education, vol. 12, no. 4, pages 385–407, 2001. (Cited on page 34.)

[Melis 2006] Erica Melis, Giorgi Goguadze, Martin Homik, Paul Libbrecht, Carsten Ullrich and Stefan Winterstein. *Semantic-Aware Components and Services of ActiveMath.* British Journal of Educational Technology, vol. 37, no. 3, pages 405–423, 2006. (Cited on page 34.)

[Michael, David R. And Chen 2005] Sandra L. Michael, David R. And Chen. Serious Games: Games that educate, train, and inform. Muska & Lipman/Premier-Trade, 2005. (Cited on page 18.)

[Michael 2005] David Michael and Sande Chen. Serious games: Games that educate, train, and inform. 2005. (Cited on page 17.)

[Mikael 2009] Lebram Mikael, Per Backlund, Henrik Engström and Mikael Johannesson. *Design and Architecture of Sidh – a Cave Based Firefighter Training Game.* In Design and Use of Serious Games, pages 19—-31. 2009. (Cited on page 17.)

[Mills 2007] Chris Mills and Barney Dalgarno. *A conceptual model for game based intelligent tutoring systems.* Proc. Ascilite, http://www. ascilite. org. au/ . . . , pages 692–702, 2007. (Cited on pages 20 and 43.)

[Mitchell 2004] Alice Mitchell and Carol Savill-Smith. *The use of computer and video games for learning: A review of the literature.* Rapport technique, 2004. (Cited on page 3.)

[Mitrovic 1996] Antonija Mitrovic, Slobodanka Djordjevic-kajan and Leonid Stoimenov. *INSTRUCT: Modeling Students by Asking Questions.* User Modeling and User-adapted Interaction - UMUAI, vol. 6, no. 4, pages 273–302, 1996. (Cited on page 46.)

[Mody 1997] M Mody, M Studdert-Kennedy and S Brady. *Speech perception deficits in poor readers: auditory processing or phonological coding?* Journal of experimental child psychology, vol. 64, no. 2, pages 199–231, February 1997. (Cited on pages 4 and 104.)

[Monk 1993] Andrew Monk, L Davenport, J Haber and Peter Wright. Improving your human-computer interface: A practical technique, volume i. 1993. (Cited on page 123.)

[Moreno-Ger 2007a] Pablo Moreno-Ger, Daniel Burgos and José Luis Sierra. *A game-based adaptive unit of learning with ims learning design and.* In Second European Conference on Technology Enhanced Learning, EC-TEL 2007, pages 247–261, 2007. (Cited on pages 21, 22 and 27.)

[Moreno-Ger 2007b] Pablo Moreno-Ger, José Luis Sierra, I Martinezortiz and B Fernandezmanjon. *A documental approach to adventure game development.* Science of Computer Programming, vol. 67, no. 1, pages 3–31, 2007. (Cited on pages 20 and 21.)

[Moreno-Ger 2008a] Pablo Moreno-Ger, Daniel Burgos, Iván Martínez-Ortiz, José Luis Sierra and Baltasar Fernández-Manjón. *Educational game design for online education.* Computers in Human Behavior, vol. 24, no. 6, pages 2530–2540, September 2008. (Cited on pages 21 and 27.)

[Moreno-Ger 2008b] Pablo Moreno-Ger, Daniel Burgos, Iván Martínez-Ortiz, José Luis Sierra and Baltasar Fernández-Manjón. *Educational game design for online education.* Computers in Human Behavior, vol. 24, no. 6, pages 2530–2540, September 2008. (Cited on page 43.)

[Motiwalla 2007] L Motiwalla. *Mobile learning: A framework and evaluation.* Computers & Education, vol. 49, no. 3, pages 581–596, November 2007. (Cited on page 125.)

[Mulwa 2010] Catherine Mulwa and Seamus Lawless. *Adaptive educational hypermedia systems in technology enhanced learning: a literature review.* In SIGITE '10 Proceedings of the 2010 ACM conference on Information technology education, pages 73–84, 2010. (Cited on page 28.)

[Nilsson 1971] Nils J. Nilsson. Problem-Solving Methods in Artificial Intelligence. McGraw-Hill, 1971. (Cited on page 46.)

[Oppermann 1994] Reinhard Oppermann. Adaptive user support: ergonomic design of manually and automatically adaptable software. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1994. (Cited on page 15.)

[Papastergiou 2009] Marina Papastergiou. *Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation.* Computers & Education, vol. 52, no. 1, pages 1–12, January 2009. (Cited on pages 3 and 126.)

[Paramythis 2009] Alexandros Paramythis. *Adaptive Systems : Development , Evaluation and Evolution Doktor.* PhD thesis, 2009. (Cited on page 123.)

[Parfitt 1998] Lynne Parfitt, Jun Jo and Anne Nguyen. *Multimedia in Distance Learning for Tertiary Students With Special Needs.* In ASCILITE, Australasian Society for Computers in Learning in Tertiary Education, volume 98, pages 561–569. Citeseer, 1998. (Cited on page 104.)

[Peachey 1986] Darwyn R. Peachey and Gordon I. McCalla. *Using Planning Techniques in Intelligent Tutoring Systems.* International Journal of Man-Machine Studies, vol. 24, no. 1, pages 77–98, 1986. (Cited on page 46.)

[Pernin 2006] Jean-Philippe Pernin and Anne Lejeune. *Models for the re-use of scenarios of training.* Rapport technique, 2006. (Cited on page 16.)

[Peter 2005] Yvan Peter and Thomas Vantroys. *Platform support for pedagogical scenarios.* JOURNAL OF EDUCATIONAL TECHNOLOGYAND SOCIETY, vol. 8, no. 3, page 122, 2005. (Cited on page 16.)

[Pintrich 1999] Paul R Pintrich. *The role of motivation in promoting and sustaining self-regulated learning.* International Journal of Educational Research, vol. 31, no. 6, pages 459–470, January 1999. (Cited on page 55.)

[Radford 2000] Antony Radford. *Games and learning about form in architecture.* Automation in Construction, vol. 9, no. 4, pages 379–385, 2000. (Cited on page 3.)

[Raibulet 2010] Claudia Raibulet, Laura Masciadri and Informatica Sistemistica. *Metrics for the Evaluation of Adaptivity Aspects in Software Systems.* International Journal, vol. 3, no. 1, pages 238–251, 2010. (Cited on page 123.)

[Ram 2007] Ashwin Ram, Santiago Onta and Manish Mehta. *Artificial Intelligence for Adaptive Computer Games.* Twentieth International FLAIRS Conference on Artificial Intelligence (FLAIRS-2007),, no. Ccl, 2007. (Cited on pages 25 and 27.)

[Rieber 1996] Lloyd P. Rieber. *Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games.* Educational Technology Research & Development, vol. 44, no. 2, pages 43–58, 1996. (Cited on page 3.)

[Rios 1999] Antonia Rios, Eva Millán, Mónica Trella, José-Luis Pérez-de-la Cruz and Ricardo Conejo. *Internet based evaluation system.* Artificial Intelligence in Education:, vol. 64, no. 18, pages 1896, 1898, September 1999. (Cited on page 29.)

[Sangineto 2007] Enver Sangineto, Nicola Capuano, Matteo Gaeta and Alessandro Micarelli. *Adaptive course generation through learning styles representation.* Universal Access in the Information Society, vol. 7, no. 1-2, pages 1–23, October 2007. (Cited on pages 36, 40, 58 and 125.)

[Schacter 2010] Daniel L. Schacter, Daniel T. Gilbert and Daniel M. Wegner. Psychology. Worth Publishers, 2010. (Cited on pages 107 and 108.)

[Schmeichel 2003] Brandon J. Schmeichel, Kathleen D. Vohs and Roy F. Baumeister. *Intellectual performance and ego depletion: Role of the self in logical reasoning and other information processing.* Journal of Personality and Social Psychology, vol. 85, no. 1, pages 33–46, 2003. (Cited on page 108.)

[Schneider 2003] Daniel K. Schneider, Paraskevi Synteta, Catherine Frété, Fabien Girardin and Stéphane Morand. *Conception and implementation of rich pedagogical scenarios through collaborative portal sites: clear focus and fuzzy edges.* In International Conference on Open and Online Learning, pages 1–40. Citeseer, 2003. (Cited on page 16.)

[Scriven 1967] M. Scriven. *The methodology of evaluation.* In R. W. Tyler, R. M. Gagné and M. Scriven, editeurs, Perspectives of curriculum evaluation, pages 39–83. Chicago, IL: Rand McNally, 1967. (Cited on page 122.)

[Sehaba 2005a] Karim Sehaba. *Exécution adaptative par observation et analyse de comportements Application à des logiciels interactifs pour des enfants autistes.* PhD thesis, Université de La Rochelle, 2005. (Cited on pages 4 and 104.)

[Sehaba 2005b] Karim Sehaba, Pascal Estraillier and Didier Lambert. *Interactive educational games for autistic children with agent-based system.* 4th International Conference on Entertainment Computing (ICEC?05), pages 422–432, 2005. (Cited on page 104.)

[Seridi 2004] H. Seridi, T. Sari and M. Sellami. *Adaptive Instructional Planning Using Neural Networks in Intelligent Learning Systems.* 2004. (Cited on page 34.)

[Shahin 2008] Reem Shahin, Lina Barakat, Samhar Mahmoud and Mohammad Alkassar. *Dynamic Generation of Adaptive Courses.* In Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on, pages 1–4. IEEE, April 2008. (Cited on pages 34 and 43.)

[Specht 1998] Marcus Specht and R. Oppermann. *ACE-adaptive courseware environment.* New Review of Hypermedia and Multimedia, vol. 4, no. 1, pages 141–161, 1998. (Cited on pages 32 and 40.)

[Specht 2001] Marcus Specht, Milos Kravcik, Leonid Pesin and Roland Klemke. *Authoring adaptive educational hypermedia in WINDS.* Proceedings of ABIS2001, Dortmund, Germany, vol. 3, no. 3, pages 1–8, 2001. (Cited on pages 34 and 40.)

[Stankov 2004] Slavomir Stankov, Vlado GaliniÄ‡ and Ani GrubišiÄ‡. *What is our effect size: Evaluating the educational influence of a web-based intelligent authoring shell.* ... International Conference on Intelligent ..., pages 1–6, 2004. (Cited on page 126.)

[Steiner 2009] Christina M Steiner, Michael D Kickmeier-Rust, Elke Mattheiss and Dietrich Albert. *Undercover : Non-Invasive , Adaptive Interventions in Educational Games.* In 1st international open workshop on intelligent personalization and adaptation in digital educational games, pages 55–65, 2009. (Cited on page 22.)

[Susi 2007] Tarja Susi. *Serious games–An overview.* Rapport technique, 2007. (Cited on pages 4 and 18.)

[Sykes 2005] E Sykes. *Qualitative Evaluation of the Java Intelligent Tutoring System.* Journal of Systemics, Cybernetics and Informatics, vol. 3, no. 5, pages 49–60, 2005. (Cited on page 126.)

[Tashiro 2009] Jayashi Tashiro. *What Really Works in Serious Games for Healthcare Education.* In Future Play '09 Proceedings of the 2009 Conference on Future Play on, 2009. (Cited on pages 3, 23 and 27.)

[Tetchueng 2008] J.L. Tetchueng, Serge Garlatti and Sylvain Laube. *A Context-Aware Learning System based on generic scenarios and the theory in didactic anthropology of knowledge.* International Journal of Computer Science and Applications, vol. 5, no. 1, pages 71–87, 2008. (Cited on pages 16 and 17.)

[Togelius 2007] Julian Togelius, Renzo De Nardi and Simon M. Lucas. *Towards automatic personalised content creation for racing games.* 2007 IEEE Symposium on Computational Intelligence and Games, pages 252–259, April 2007. (Cited on pages 24, 25 and 27.)

[Torrente 2009] Javier Torrente, Pablo Moreno-Ger, B. Fernández-Manjón and A. del Blanco. *Game-like Simulations for Online Adaptive Learning: A Case Study.* In Proceedings of the 4th International Conference on E-Learning and Games: Learning by Playing. Game-based Education System Design and Development, page 173. Springer, 2009. (Cited on pages 23 and 27.)

[Ullrich 2007] Carsten Ullrich. *Course Generation as a Hierarchical Task Network Planning Problem.* PhD thesis, 2007. (Cited on pages 37, 38, 40, 50 and 51.)

[Ullrich 2008] Carsten Ullrich. *Course Generation in Practice: Formalized Scenarios.* Pedagogically Founded Courseware Generation for Web-Based Learning, pages 111–167, 2008. (Cited on pages 37 and 40.)

[Ullrich 2009a] Carsten Ullrich and Erica Melis. *Pedagogically founded courseware generation based on HTN-planning.* Expert Systems with Applications, vol. 36, no. 5, pages 9319–9332, 2009. (Cited on pages 17 and 43.)

[Ullrich 2009b] Carsten Ullrich and Erica Melis. *Pedagogically founded courseware generation based on HTN-planning.* Expert Systems with Applications, vol. 36, no. 5, pages 9319–9332, 2009. (Cited on page 37.)

[Ullrich 2010] Carsten Ullrich and Erica Melis. *Complex Course Generation Adapted to Pedagogical Scenarios and its Evaluation.* Educational Technology & Society, vol. 13, no. 2, pages 102–115, 2010. (Cited on pages 33, 37, 58 and 123.)

[Van Marcke 1990] Kris Van Marcke. *A generic tutoring environment.* In L Aiello, editeur, Proceedings of the 9th European Conference on Artificial Intelligence,, pages 655–660, Stockholm, Sweden, 1990. Pitman , London. (Cited on pages 29 and 40.)

[Van Marcke 1992] Kris Van Marcke. *A generic task model for instruction.* In Sanne Dijkstra, editeur, Instructional Models for Computer-Based Learning Environments, pages 234–243. Springer-Verlag, Berlin, Berlin, Heidelberg, nato asi s édition, 1992. (Cited on pages 29 and 58.)

[Van Marcke 1998] Kris Van Marcke. *GTE: An Epistemological Approach to Instructional Modelling.* Instructional Science, vol. 16, no. 3-4, pages 91–147, 1998. (Cited on page 29.)

[Van Velsen 2008] Lex Van Velsen, Thea Van Der Geest, Rob Klaassen and Michaël Steehouder. *User-centered evaluation of adaptive and adaptable systems: a literature review.* The Knowledge Engineering Review, vol. 23, no. 03, pages 261–281, September 2008. (Cited on page 123.)

[Vanlehn 1987] Kurt Vanlehn. *Student Modelling.* In Foundations of intelligent tutoring systems. 1987. (Cited on page 55.)

[VanLehn 2005] K VanLehn, Collin Lynch and Kay Schulze. *The Andes physics tutoring system: Five years of evaluations.* ... the 2005 conference on ..., 2005. (Cited on page 126.)

[Vanlehn 2007] Kurt Vanlehn, Arthur C Graesser, G Tanner Jackson, Pamela Jordan, Andrew Olney and Carolyn P Rosé. *When are tutorial dialogues more effective than reading?* Cognitive science, vol. 31, no. 1, pages 3–62, February 2007. (Cited on page 19.)

[Vartiainen 2002] Pirkko Vartiainen. *On the Principles of Comparative Evaluation.* Evaluation, vol. 8, no. 3, pages 459–371, 2002. (Cited on pages 125 and 127.)

[Vassileva 1990] Julita Vassileva. *A classification and synthesis of student modelling techniques in intelligent computer-assisted instruction.* Computer Assisted Learning, vol. 438, pages 202 – 213, 1990. (Cited on page 46.)

[Vassileva 1992] Julita Vassileva. *Dynamic CAL-courseware generation within an ITS-shell architecture.* Computer assisted learning, vol. 602, pages 581–591, 1992. (Cited on pages 29 and 58.)

[Vassileva 1995] Julita Vassileva. *Dynamic courseware generation: at the cross point of CAL, ITS and authoring.* In Proceedings of ICCE, volume 95, pages 290–297, 1995. (Cited on pages 30 and 40.)

[Vassileva 1996] Julita Vassileva. *Instructional planning approaches: from tutoring towards free learning.* Proc. EuroAIED, no. 1966, pages 1–8, 1996. (Cited on pages 43 and 46.)

[Vassileva 1997] Julita Vassileva. *Dynamic Courseware Generation.* Communication and Information Technologies, vol. 5, no. 2, pages 87–102, 1997. (Cited on page 30.)

[Vassileva 1998a] Julita Vassileva. *DCG + GTE: Dynamic Courseware Generation with teaching expertise.* Instructional Science, vol. 26, pages 317—-332, 1998. (Cited on page 30.)

[Vassileva 1998b] Julita Vassileva and R Deters. *Dynamic courseware generation on the WWW.* British Journal of Educational Technology, vol. 29, no. 1, pages 5–14, 1998. (Cited on page 30.)

[Viet 2006] Anh Viet and D.H. Si. *Acgs: Adaptive course generation system-an efficient approach to build e-learning course.* In Computer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on, pages 259–259. Ieee, September 2006. (Cited on pages 37 and 40.)

[Villamañe 2001] Mikel Villamañe, Julián Gutiérrez, Rosa Arruabarrena, Tomás A. Pérez, Sara Sanz, Silvia Sanz, Javier López and José A. Vadillo. *Use and evaluation of HEZINET; A system for basque language learning.* Proceedings of the ICCE, pages 93–101, 2001. (Cited on page 126.)

[Virvou 2001] Maria Virvou and K.aterina Katerina. *Evaluation of the advice generator of an intelligent learning environment.* Advanced Learning Technologies, 2001. . . . , pages 339–342, 2001. (Cited on page 125.)

[Wasson 1990] B. Wasson. *Determining the Focus of Instruction: Content planning for intelligent tutoring systems.* Research report, University of Saskatchewan, 1990. (Cited on page 30.)

[Weber 1997] Gerhard Weber and Marcus Specht. *User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems.* History, 1997. (Cited on page 32.)

[Weibelzahl 2002] Stephan Weibelzahl and G Weber. *Advantages, opportunities and limits of empirical evaluations: Evaluating adaptive systems.* KI, 2002. (Cited on page 123.)

[Win 2002] Bart De Win and Frank Piessens. *On the importance of the separation-of-concerns principle in secure software engineering. . . .* of Engineering Principles . . . , pages 1–10, 2002. (Cited on page 45.)

[Wong 2007] Wee Ling Wong, Cuihua Shen, Luciano Nocera, Eduardo Carriazo, Fei Tang, Shiyamvar Bugga, Harishkumar Narayanan, Hua Wang and Ute Ritterfeld. *Serious video game effectiveness*. Proceedings of the international conference on Advances in computer entertainment technology - ACE '07, page 49, 2007. (Cited on page 3.)

[Wu 1998] Hongjing Wu, GJ Houben and Paul De Bra. *Aham: A reference model to support adaptive hypermedia authoring*. Proceedings of the Conference on . . . , pages 1–19, 1998. (Cited on page 48.)

[Yang 2007] Jongyeol Yang, Seungki Min, CO Wong and Jongin Kim. *Dynamic game level generation using on-line learning*. In Technologies for E-Learning, pages 916–924, 2007. (Cited on pages 24, 25 and 27.)

[Zouaq 2008] Amal Zouaq, Roger Nkambou and Claude Frasson. *Bridging the Gap between ITS and eLearning: Towards Learning Knowledge Objects*. Intelligent Tutoring Systems, pages 448–458, 2008. (Cited on page 124.)

[Zyda 2005] Michael Zyda. *From visual simulation to virtual reality to games*. Computer, vol. 38, no. 9, pages 25–32, 2005. (Cited on page 17.)