SITUATION-BASED SCENARIOS FOR E-LEARNING

Fabrice Trillaud, Phuong Thao Pham, Mourad Rabah, Pascal Estraillier, Jamal Malki

SITUATION-BASED SCENARIOS FOR E-LEARNING

FabriceTrillaud, Phuong Thao Pham, MouradRabah, Pascal Estraillier and Jamal Malki
L3i Laboratory, University of La Rochelle, La Rochelle, France
{fabrice.trillaud, phuongthao.pp, mourad.rabah, pascal.estraillier, jamal.malki}@univ-lr.fr

ABSTRACT

Online Distance Learning (ODL) applications as other interactive systems need to be scenarized in order to represent and manage users' activities in an effective way. To help applications designers, a clear and flexible framework should guide scenario elaboration. We propose an approach based on scenario structuring using the notion of "situation". This framework takes place in our overall ODL management methodology, namely a before-throughout-after scenario creation, execution and evolution cycle. A "situation" in our approach is an elementary building block dividing system's actors' interactions into contextually independent scenes and allowing scenario description and adaptation. In this paper, we discussour methodology to manage ODL applications,introduce the situation-based scenarios and illustrate how they can be applied on a case study in order to enhance Online Distance Learning.

KEYWORDS

Situation-based model; adaptive scenario; online distance learning; scenario management; scenario execution process

1. INTRODUCTION

Online Distance Learning (ODL) is now becoming a mature research domain (Lionarakis, 2009). We observe the development of generic online meeting applications for training purpose, usedto create online virtual classrooms. But these are not well suited to a training context, and specialized ones can require too much learning time to master, slowing down their acceptance by the users (Murphy and Manzanares, 2008;Kinser, 2003;Jung et al., 2008;Pigliapoco et al., 2008). Moreover, users expect new learning possibilities through the use of these applications (Hay et al., 2004).

ODL is not a mere unfolding of a "video conference" or an "online meeting". A learning session can be seen as a "scenario" that organizes the educational activities. However, scenario organization for training purpose in online learning is still rarely used or studied.Furthermore, scenario elaboration methods for learning applications should be flexible enough to, on one hand, facilitate scenario conception and, on the other hand, allow its unavoidable modification and adaptation to a given learning context. Hence, the scenario management should be done through a scenario "lifecycle" wherethe scenario is created (before the learning session start), executed (during the learning session) and modified (after learning session feedback evaluation). We have defined a methodology that integrates this scenario lifecycle.

It is also important to structure and organize learning sessions in an effective way. A real learning session is a set of relatively independent activities. We thus propose to decompose the scenario into contextually independent blocks and introduce the notion of "situation". Each "situation"represents a kind of scenethat gatherssystem's actors'interactions within a given shared context (Pham et al., 2011). This allows us to structure the overall scenario according to users activities and give us flexibility for scenario creation (by choosing the activity "situations" blocks) and modification or adaptability (during or after the training session according to the context, the feedbacks or the results). Furthermore, we can associate to the scenario-based structuring interactive storytelling techniques as well as consistency management in dynamical execution to ensure robustness and interactivity for online learning.

Several authoring environments for the e-learning do exist (as LAMS andCopperCore)[1]. Their main aim is to provide the authors with a tool to structure the learning activities and to give to the learners a view on

[1] LAMS - http://www.lamsinternational.com/, CopperCore - http://coppercore.sourceforge.net/

their learning process. Our ODL system and methodology allow not only to structure the learning activities but also: i) to manage the overall learning sequence creation and improvement using feedbacks, ii) to control users (learners and trainers) interactions by contextualizing the activities and confining the interactions into "situations", and iii) to dynamically and automatically adjust the learning sequence according to the activities' progression and using pre-/post-conditions for activities chaining.

Our finalaim isto achieve a complete generic situation-based scenario management system, including authoring tool, execution environment, and scenario evolution control all along its lifecycle, since users are looking for a system able to adapt itself to its context and environment (McKee, 2010). The resulting system has to be accessible for everyone and should not requirea particular training or equipment;these aretwo of the main problems associated with the implementation of ODL systems (Cardoso and Bidarra, 2009).

Thepaper presents our ODL scenario elaboration methodology, gives a quick overview of our situation-based model and details its application to an ODL case study.

2. ODL METHODOLOGY

The purpose of our work is to show how our experience and results in the interactive storytelling domain can be applied to ODL(Pham et al 2011) sincewe are convincedthat seamless adaptive scenario management suitswellto this kind of environment, though it is still rarely used or studied. We have defined a methodology to establish a library of "situations" within an authoring system that allows: i) trainers to create prepare and modify their courses scenarios, and ii) the system to automatically pilot the learning session.

2.1 Environment

We must first determine the actors and their environment. We define the following types of human actors:

- ***Learner:*** attending the lessons. The learners are assimilated to students.
- ***Trainer:***courses designer (pedagogue) and/or courses presenter (lecturer). We distinguish the both cause a lesson presenter/lecturer may not be its designer. A trainer can be assimilated to a teacher.
- ***Training organization:*** decision-makers for the content and the modalities of a lesson. This actor definesthe rules to be followed by the designer and may be involved in the course's review process, as evaluator.

We also define the "roles" that assign a set of behaviors to an actor.Each actor can have several roles. In our work, actors' roles are deeply linkedto the situation-based architecture. The "situation" structure restricts actors' interactions, allowing actors to execute only a limited set of actions bounded by theconsidered role:

- ***Auditor:***learner's default role. An *auditor*has a view on the trainer's virtual board and access to the classroom resources: documents, audio/video material, messaging… An *auditor* can write on his own virtual notebook and may have access to other *auditors*' notebooks under certain circumstances (for instance teamwork). He can create private discussions with classmates and ask questions to the trainer.
- ***Presenter:***a*presenter* can write on the trainer's board and speak to the virtual classroom. His audio and video streams are automatically broadcasted to every other participant.
- ***Moderator:*** manages groups and rights. He can create polls and designate auditors to become presenters.
- ***Designer:*** manages the lecture's content. He can add or remove resources, alter the course's scenario…
- ***Evaluator:*** evaluates the pertinence of the events recorded during the class. These events are commented (contextualized), and dispatched to the involved actors to improve the next execution.

2.2 Global Methodology

We have adopted a three-phase cycle, namely *Before*(scenario construction), *Throughout*(scenario execution) and *After*(scenario evaluation).Using this cycle, we identifywhen thedifferentactors are involved in theoverall scenario's lifecycle togive them theappropriate tools and clues to improve it.

Each ODL actor is involved in one or several phases of the scenario's lifecycle. Figure 1 illustrates, through a three-segment arrow, how each actor, by its roles fits into this cycle. The three phases are subdivided into seven steps:

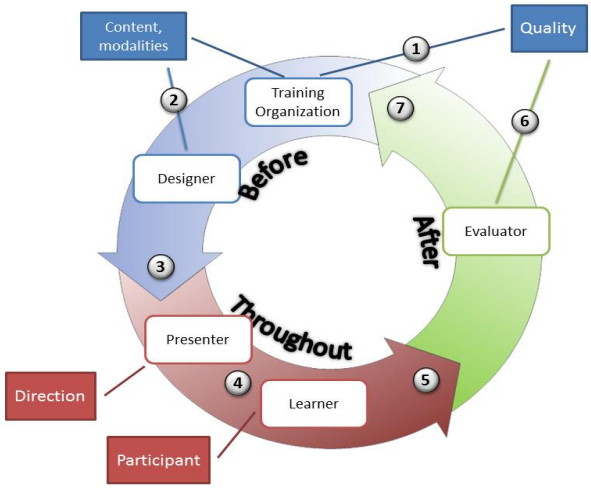

Figure 1 Scenario's lifecycle in ODL

1.The training organization sets:training quality standards,training rules related to the expected results, events the system should keep track of...
2.The designer, and possibly the training organization, defines the content and the modalitiesof the training. They elaborate the training scenario, choose the associated resources,define the possible exercises…
3.The training material is given to the presenter, who may collaborate with the designer to alter the scenario or just to get complementary information.
4.The training session is executed. The presenter directs it, and may alter the scenario again. The learner participates to the session. Events occurring during the session are recorded.
5.The recorded events are given to the evaluator.
6.The events are evaluated, and the most relevant ones, according to the global rules and to the training's quality requirements, are highlighted.
7.These events' records are finally dispatched to the concerned actors, who can draw conclusions that serve to improve the next training session.

It is worth noting that the above three phase scenario lifecycle can be applied to any interactive system with scenario-based execution.

2.3 Structuring Lectures into Activities

Our scenario specification is based on the decomposition of pedagogic activities in a face-to-face classroom. A classical course is regularly unfolded into several steps, such as lesson lecture, exercise working, going to the blackboard, examinations… Each step consists of a set of activities between the trainer and the learners in relation to pedagogic content and resources. Figure 2 gives an example of our partitioning of main activities in classical face-to-face course. The granularity depends on application specification. The main activities can be subdivided into more detailed ones, but the primitives (last level) should be elementary actions. InFigure 2, we detail two levels of activities: level 1 refers to the general learning steps and level 2 refers to sequences of activities composing the general steps. The course progression is a sequence combining activities from the level 2 according to course's goals and trainer's intention.

This lecture structuring into activities justifies the "situation"block in our work. Indeed, our "situation" concept is well suited to represent learning activities in ODL because these scenario structuring blocks allow actions and activities contextualization. The situation library is defined according to the basic ODL action sequences. Moreover, the transition from face-to-face class to ODLvirtual classroom requires a larger quantity of interactions and collaborations between trainers and learners. Thus, we thus also propose some others additionalactivities, such as the trainer's moderation, or course supporting tools.

In the next section, we presentthe general concepts of oursituation-based model: static structure of elementary situations, graph of situation and then dynamic execution mechanism.

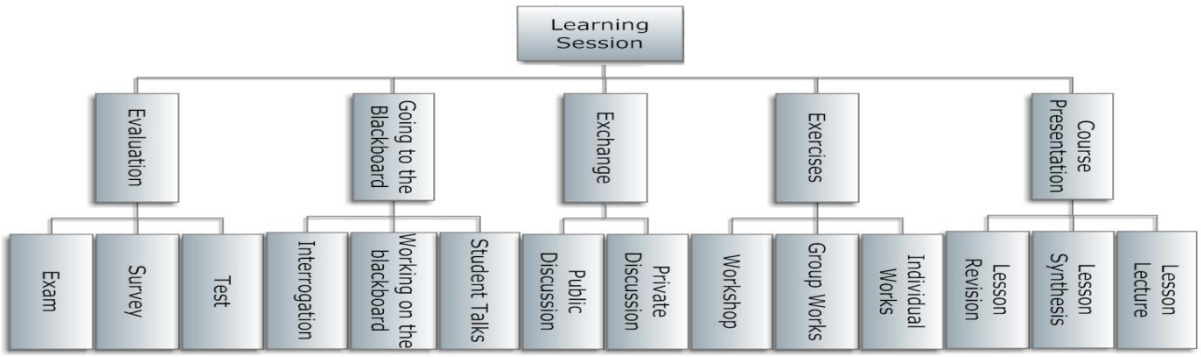

Figure 2 Activities in ODL

3. APPLICATION STRUCTURING USING SITUATIONS

In the interactive domain, story or plot structuring has been studied in many works, such as in games, interactive drama and storytelling. The story in Haunt 2 (Magerko, B. 2005), a 3D interactive storytelling game is represented by a set of ordered scenes, called plot points. Each plot point is defined by a set of preconditions, a set of actions to fulfill and a timing constraint. The Façade project (Mateas& Stern 2005) developed by Mateas and Stern is another interactive drama, in which the story is structured in 27 beats. Each beat is composed of a collection of joint dialog behaviorsthat are the atomic unit of dramatic actions representing several lines of dialog between two characters. In PAPOUS (Silva et al. 2003), the story to be told is organized in levels and each level consists of a set of StoryBits,characterized by different properties, characters and events. All of these works focus on representing a plot as a sequence of actions/events linked together by cause and effect (Paul et al. 2011). In MIST, the approach for story creation and management comes from Hierarchical Task Network planning, where each task is re cursively decomposed into subtasks until the primitive actions.

Story or scenario planning can be found easily in the game or drama applications, but also in simulation or education applications. The interactive applications can be organized as a scenario defining the interactive environment and tasks that actors have to do. And as we mentioned above, several works have differently considered scenario decomposing and structuring. In order to provide a scenario design model that suits to ODL, we propose the notion of "situation" corresponding to a scene that contextualizes actions and interactions in a scenario. The situations are our narrative basic blocks; therefore, they facilitate the story planning and management by characterizing and confining interactions.

3.1 Elementary Situation Structure: Basic Scenario Block

Definition: An elementary situation is a sequence of interactions between two or more actors in a givencontext to achieve a predictive objective. It is characterized by the following set of elements (illustrated inFigure 3):

- ***Pre-conditions***: set of conditions, concerning the global system statethatmust be fulfilledto enter the situation. The situations'**local context**will be established from these pre-conditions.
- ***Post-conditions***: set of conditions or results that must be completed in order to quit the situation. These areresults re-injected back into the global state.
- ***Progression:*** presents a frame where actors behave and interact usinglocal resources, launching different tasksaccording to their objectives. Due to the fact that actors' behaviors are not always modeled, the progression of a situation can be executed in a non-predictable way. Asituation progressioninvolves: *Actors* (physical or virtual actors), and *Application Designer Logic* (what the system is supposed to do).

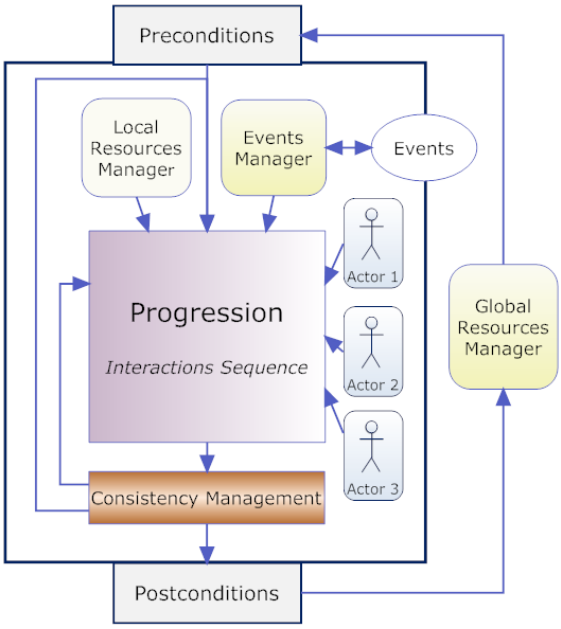

Figure 3 Model of a situation

- ***Event management:*** deals with the incoming/outgoing events from/tothe external world or others simultaneous situations.
- ***Resources managers:*** are responsible forlocal or global resources' accesswith respect to actors' demands.
- ***Consistency management***: mechanisms devoted to the prevention, detection and treatment solutions, in order to redress or adjust the situation progression despite data inconsistency and actors misunderstandings (Pham et al 2011).

Besides the basic elementary situation, we distinguish 2 additional kinds of situations: *composite situations*are abstract situations that recursively include other situations and *usable situations* or parameterized situations are ready-to-use situations, predesigned and instantiated with common parameters.

3.2 Application Scenario: Situation graph

An *application scenario* is represented as an ordered sequence or directed graph of all possible situations, elementary and composite, that can happen between actors. This graph shows the causal relationships between scenario situations, without taking into account resources uses and events management. Figure 4 gives a general example of *situations graph* that includes alternative(choice) or parallel(afterwards synchronized) situations chaining.

It is worth noting that situation composition is recursive: a composite situation can include other composite situations. Thus, a *scenario* can be seen as the highest-level composite situation that is not covered by any other situation.

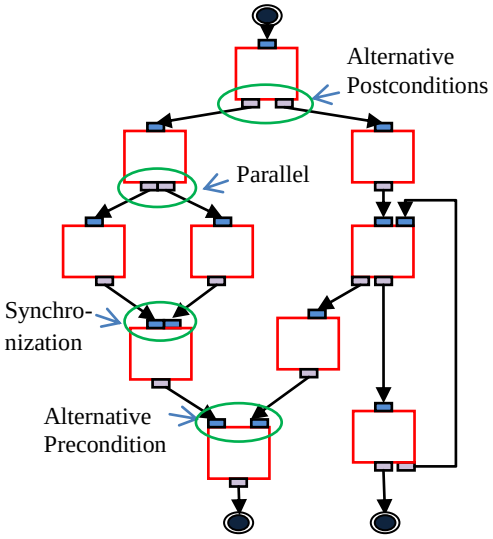

Figure 4 Example of situation graph

3.3Situation Execution Mechanism

The execution of a situation-based scenario uses a main loop similar to what can be found on many systems or applications: first, choose the next action to be performed, and then execute it. The execution of a situation uses the before/throughout/after model, similar to the one presented in section 2, and integrates the consistency management mechanism. The core execution of a situation is surrounded by preand post conditions processing. Hence, the execution is divided into three phases: prologue, execution loop and epilogue. This three-phase organization allows us integrate algorithms and techniques to achieve theinteraction consistency. Figure illustrates the execution's process using BPMN[2]. A situation is executed according to the following steps:

- ***Prologue:***processing the situation's preconditions, local context initialization, synchronizing actors' local visions (in order to avoid forthcoming errors)…
- ***Execution loop:*** situation's core, where interactions take place. The loop goes on until the situation's post conditions are reached, or until an event causes interactions stop. The "execution" task starts and ends with actors' local visions "synchronization". The situation interactions consist of one or more loop executions. They can be interrupted by the detection of inconsistencies that lead to dynamic consistency management, or by some internal or external event that may lead to adaptation management. The execution mechanisms are further discussed in other papers (as Pham et al 2011).
- ***Epilogue:*** processing the situation's post conditions and ensuring the system's consistency. If the required conditions are not met at the end of the situation, some supplementary interactions may be carried out to correct it, or, if necessary, bring the whole situation back to the initial state. If there is no possible correction, then the situation is forced to endand a message is dispatched to the system.

3.4 Situationschaining

The other question related to situations execution is how to choose the right following situation according to the context. This step refers to the plot direction. There are two main approaches of plot direction in interactive storytelling domain: "*Oriented Scenario*" (Magerko, B. 2005;Mateas & Stern 2005;Silva et al. 2003), and "*Emergent*"(Prigent et al. 2005). According to these two approaches, we propose two ways of choosing the next situation to execute: if a scenario is associated with the course, we try to follow it. If there is no scenario, or if it doesnot fulfill the pre or post conditions of the situations, the system draws from the situation library, the most suited one, according to conditions and past executions.

[2] Business Process Model and Notation, an Object Management Group standard - http://www.bpmn.org/

4. APPLICATION TO ODL CASE STUDY

This section describeshow we have applied our proposalto our OnlineDistance Learning framework[3]. This framework is devoted toactivities structuring in virtual distributed classroom. We use our situation-based structuring to organize learning activities. We list the corresponding situations and build the situations graph.

4.1 Situations Library

An ODL session is an organization of pedagogic activities that are carried out between the trainer and the learners in a virtual class. As mentioned in section 2, an ODL session activities can be structured assituation-based scenario. ODL sessions are created from a predefined situations library by the course designer or the trainer himself. The scenario structuring into situations is based on the main pedagogic activities as in a real classroom. ODL situation library containsthe twocategories of situations: elementary and composite.

Elementary situations correspond to generic activities, such as presentation, individual exercise working, discussion, examination… We have identified the following elementary situations (SE): Presentation (SE-Pre), Moderation (SE-Mo), Discussion (SE-Disc), Individual Working (SE-IW), Collaborative Working (SE-CW), Browser Sharing (SE-BroSha), Survey (SE-Surv) andGrade (SE-Grade).

Each composite situation corresponds to a major step in a session and includesseveral generic activities, such as the lesson presentation, class working… Here is the set of ODL composite situations (SC): Lesson Lecture (SC-LesLec), Class Working (SC-ClaWork), Going to the Blackboard (SC-GotoB) and Evaluation (SC-Eval).Forexample, the "Lesson Lecture"(SC-LesLec) situation is a composition of the 5 elementary situations: SE-Pre, SE-Mo, SE-Disc, SE-BroSha and SE-Surv. The links between these 5 situations are not static or predefined but dynamic and based on the fulfillmentof pre/post-conditions.

The situations library is determined by the application designer according to the development point of view. The situations have to satisfy the reusability and generalization. Otherwise, they will not be fully exploited by trainers to create learning scenarios in application's authoring system. In order to facilitate their creation, we have also defined a set of "usable situations" (see 3.1). These situations can be seen as elementary or composite situations instantiations. Indeed, each situation in the library is a kind of class, so each usable situation is an elementary or composite situation parameterized and/or contextualized by advance with appropriate actors' roles, system's resources…This increases the application ergonomics, avoiding the repetitive and complex situations configuration. Table 1gives samples of each kind of mentioned situations.

4.2 Nominal Execution

To illustrateour proposition, we have defined a standard typical scenario to be structured with situations. The lesson scenario is as follows: Start of the lesson, Class work, Sending a student to the blackboard and Discussion.This four-step lesson must now be "converted" into situations.Figure 6shows a possible

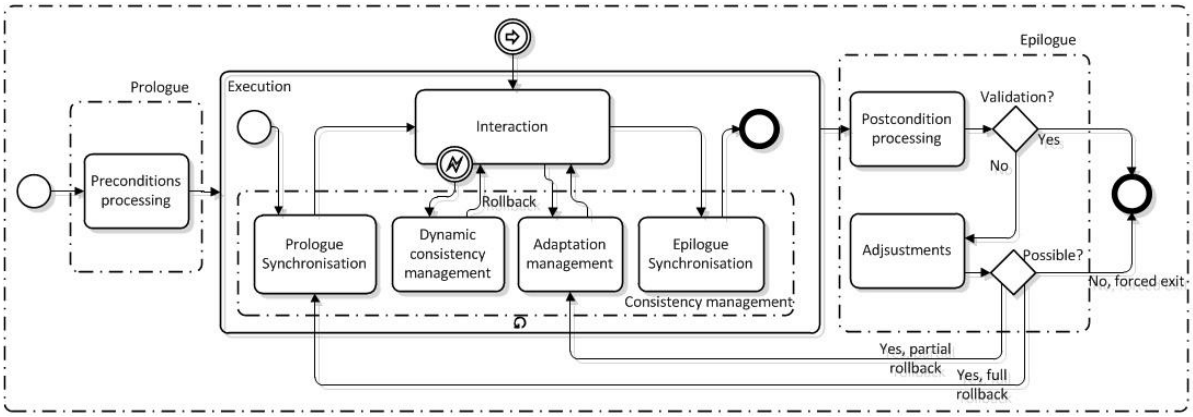

Figure 5Situation's Execution Mechanism using BPMN Representation

[3] see http://foad-l3i.univ-lr.fr for more information about our ODL project

corresponding scenario using the situations introduced above. The situation chaining is triggered by various events during the execution (although it can also be manually triggered). Here are some possible examples:

- Publishing the exercise ends Presentation (SE-Pre) and starts Individual Work (SE-IW).
- Designating a learner starts Going to the Blackboard (SC-GotoB).
- Sending the learner back to his place starts Discussion (SE-Disc).
- Publishing a survey starts Survey (SE-Surv).

4.3 Adaptation Examples

The scenario inFigure 6is constructed before the course begins. This predetermined scenario shows some weak points because of its rigid and static structure. The trainer may plan out his course scenario in the *before* step (see 2.2), but it is not certain that he will follow it exactly as he intended initially. Therefore, although the scenario is predetermined, it must be adaptive in order to be easily modified or customized during the course. To illustrate this adaptation, we take a linear version of the above scenario composed of the sequence SE-Pre, SE-IW, SC-GotoB, SE-Disc and SE-Surv.

If the trainer decides to change the classwork part: he prefers workgroup and doesn't want to send a learner to the blackboard,Figure shows the required changes in the scenario to fulfill the trainer's wish.These changes can be triggered in two different ways:

- ***Manually:*** the trainer uses the scenario tool to remove the unwanted situations and insert the desired one.The system has just to check the pre/postconditions consistency to validate or not the changes.
- ***Automatically:***before or after publishing the exercise, the trainers use the moderation tools to create groups. The preconditions for *Individual Work* are no longer fulfilled, since groups have been created, so the system looks for the closest situation available, here*GroupWork*. The *Going to the blackboard*situation can easily be skipped by not designating a learner.

We do not mention here the adaptation resulting from consistency management, external events or other types of adaptation. These are further discussed in other papers. More details about the case study architecture and the adaptive scenario management can be found in (Trillaud et al. 2011).

5. CONCLUSION

We have presented the methodology overview of our situation-based adaptive scenario management approach. We have defined "situations" as elementary narrative units. Each situation confines a sequence of interactions between actors in a common context and with a given set of resources. It is also characterized by sets of preconditions andpostconditions, and may integrate consistency management in order to deal with actors' misunderstandings and local vision inconsistencies. A scenario is built as a partially ordered situation sequence. The situation model is not only used to build scenarios, but also to improve them through the scenario lifecycle. The situations support the scenario creation Before system execution beginning; the automatic navigation, dynamic adaptation and consistency management in the Throughout step; and scenario improvement After its execution.

Table 1 Example of 3 types of situations

Pre SE - CW Post	**Elementary Situation « Collaborative Working »** Pre = {*InWorking & SizePlus2*} A.role = {*auditor*} B.role = {*auditor*} C.role = {*auditor*} F.role = {*moderator, designer*} Post = {*FiniExe* or *WriteUp* or *OutDeadline* or *TeacherStop*}
Pre SC - ClaWork Post	**Composite Situation « Class Working »** Pre = {$Exe_{loading}$} A.role = {*auditor*} B.role = {*auditor*} C.role = {*auditor*} F.role = {*moderator, designer*} Post = {*AllFiniExe* or *AllWriteUp* or *OutDeadline* or *TeacherStop*}
Pre SU - GroupWork Post	**Parameter Situation « Group Working»** Pre = {$Exe_{loading}$} Parameters = { *s ≥ 2, Range= groupe, Consul = Yes*} A.role = {*auditor*} B.role = {*auditor*} C.role = {*auditor*} F.role = {*moderator, designer*} Post = {*AllFiniExe* or *AllWriteUp* or *OutDeadline* or *TeacherStop*}

The notion of "situation" is proposed to be a reusable model to all the types of interactive applications for which the execution can be organized as a scenario. Since the ODL is organized as sequence of learning activities, it suits well to situation-based structuring. We have categorized ODL actors with their different roles and identified lessons activities. Then we have defined the library of situations that is used to structure lessons in ODL. This may allow a better interactive online training, compared to most existing online training

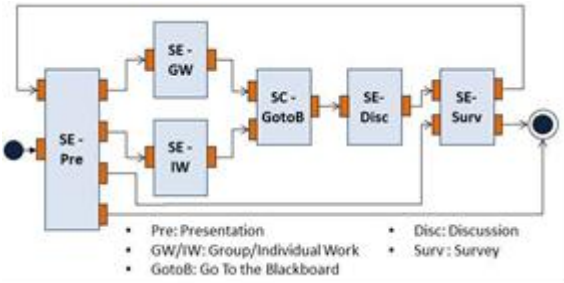

Figure 6Exemple of scenario

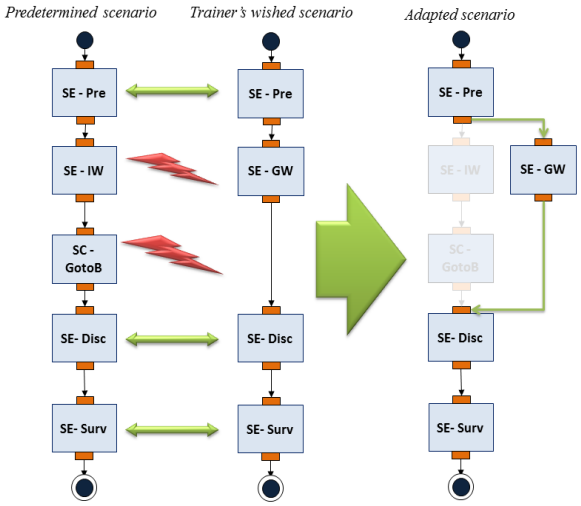

Figure 7 Example of Scenario's Adaptation

or meeting applications: the trainer gets a permanent feedback on the training execution; keeps control over the scenario and can adapt it dynamically.We gave a glimpse of what the scenario adaptation can be within situation-based scenario, but there is much more to say about it, through tracks analysis, consistency management or interactive storytelling. This will be covered in future publications.

We are now completing our work through several aspects: development of the authoring systemsupporting scenario creation and improvement, use of traces analysis to support the scenario's lifecycle, complete integration of scenario management within our ODL framework, and live tests with real online trainers and learners. Our final aim is to estimate and prove the performance and pertinence of our scenario management methodology for different scenarios in various interactive applications fields…

REFERENCES

Cardoso, V., &Bidarra, J. (2009). Open and Distance Learning: Does IT (Still) Matter? *E-Learning*, 1-6.

Hay, A., Hodgkinson, M., Peltier, J. W., &Drago, W. A. (2004).Interaction and virtual learning.*Strategic Change*, *13*(4), 193–204. Wiley Online Library.

Jung, M. L. L., Loria, K.,Mostaghel, R., &Saha, P. (2008). E-Learning: Investigating University Student's Acceptance of Technology. *European Journal of Open Distant E-Learning*

Kinser, K. (2003). Diversity Within the Virtual Classroom. *New Directions for Institutional Research*, *2003*(118), 69-77.

Lionarakis, A. (2009). The theory of distance education and its complexity.*E-Learning*, 1-7.

Magerko, B. (2005). Story Representation and Interactive Drama.(R. M. Young & J. E. Laird, Eds.)*Artificial Intelligence*, 87-92. AAAI Press.

Mateas, M., & Stern, A. (2005).Structuring Content in the Façade Interactive Drama Architecture.*Proceedings of Artificial Intelligence and Interactive Digital Entertainment* (Vol. 3, pp. 93-98).AAAI.

McKee, T. (2010). Thirty years of distance education: Personal reflections. *The International Review of Research in Open and Distance Learning*, *11*(2), 100-109.

Murphy, E., &Manzanares, M. A. R. (2008). Contradictions between the virtual and physical high school classroom: A third-generation Activity Theory perspective. *British Journal of Educational Technology*, *39*(6), 1061-1072.

Paul, R., Charles, D., McNeill, M., & D, M. (2011). Adaptive Storytelling and Story Repair in a Dynamic Environment.*The Fourth International Conference on Interactive Digital Storytelling ICIDS 2011*, Vancouver, Canada

Pham, P. T., Rabah, M., &Estraillier, P. (2011).Handling the Misunderstanding in Interactions : Definition and Solution.*The Annual International Conference on Software Engineering & Applications SEA 2011*, 47-52, Singapore

Pigliapoco, E., Torrisi, G., Messina, M., &Bogliolo, A. (2008).LoL Classroom: A Virtual University Classroom Based on Enhanced Chats. *European Journal ofOpen, Distance and E-Learning, Vol. II.*

Prigent, A.,Champagnat, R.,&Estraillier,P. (2005). Scenario building based on formal methods and adaptative execution.*ISAGA'2005 International Simulation and gaming association Georgia Institute of technology*, *72*(7), 1-19.

Silva, A., Raimundo, G., &Paiva, A. (2003). Tell me that bit again... Bringing interactivity to a virtual storyteller. *International Conference on Virtual Storytelling*, 1-10, Toulouse, France

Trillaud, F., Malki, J., &Estraillier, P. (2011). Going to the Blackboard: a case study. *Online Educa Berlin, Germany*