

EDoS: An authoring environment for serious games design based on three models

Chi Dung TRAN, Sébastien GEORGE, Iza MARFISI-SCHOTTMAN
Université de Lyon
INSA-Lyon, LIESP, F-69621, Villeurbanne, France

chi-dung.tran@insa-lyon.fr
iza.marfisi@insa-lyon.fr
sebastien.george@insa-lyon.fr

Abstract: Serious games (SGs), the confluence of e-learning and videogames, have been developing very fast these past years. Indeed, SGs combine aspects of tutoring, teaching, training, communication and information, with entertainment elements derived from videogames, in order to capture people's attention for purposes that go beyond pure entertainment. However, the creation of a SG for educational purposes and professional training is a very time-consuming and expensive process. The challenge is to combine learning objectives and fun characteristics with an acceptable budget and time. For these reasons, we propose an interactive authoring environment, called EDoS (Environment for the Design of Serious Games), designed to assist SG authoring team.

The EDoS is based on three models: (1) a formal domain-specific model of pedagogical objectives (competencies, knowledge, and behaviours) at which a SG aims; (2) IMS-LD-SG, an extension of the IMS-LD specification made specifically for SGs and (3) a task model formalized with CTT (Concurrent Task Trees) used to formalize Human-Computer Interactions sequences for screens of the game.

The first model is the base of any SG design. Once pedagogical objectives are specified, the SG scenario must be elaborated very precisely to achieve them. The second model IMS-LD-SG is used for this purpose. A SG scenario is structured into logical chains and organizations, at different levels (play, modules, acts, activities) with which the users (learners or staff such as teachers, tutors, etc.) must interact in order to achieve their learning or assistance objectives. During the whole scenario elaborating process, these elements must be clearly defined and linked with the pedagogical objectives of the first model and also with gaming activities. These gaming elements and fun characteristics (amusing interactions, actions of characters, attractive competition, adventure, etc.) are specified by the CTT task model.

The process of designing a SG is long and complicated and EDoS is developed to help the authoring team accomplish it. Indeed, it allows designing a SG through several steps, from formalizing the pedagogical objectives to elaborating a multi-level scenario and modelling Human Computer Interactions (HCI) sequences, in a structural and formal way. Moreover, EDoS also allows reusing resources (web applications, mini-games or components) of different suppliers. The outcome of EDoS is a structured scenario which will be automatically executed by an engine in the next phase of our production chain model. The EDoS provides SG designers with a method to create SGs faster and more efficiently, in a well organized process.

Keywords: Serious games, scenario design, authoring environment, e-learning, pedagogical objective

1. Introduction to Serious games

Today, Serious Games (SGs), the confluence of e-learning and videogames, are widely used in a broad range of application domains such as army training, education, advertisement, healthcare, etc. Indeed, SGs combine aspects of tutoring, teaching, training, communications and information, with entertainment elements derived from videogames, in order to capture people's attention for purposes other than pure entertainment (Alvarez, 2008). A SG can be used in order to advertise a product (advergame, ex. Pepsiman developed to advertise for Pepsi Cola); to teach effective skills and behaviours in a simulated context (simulation games, ex. FlightGear that simulates a flight environment) or to capture people's interest as well as recruit them for a certain organisation (persuasive games, ex. American's Army that helps recruit soldiers for the American army) etc. In this

paper, we are particularly interested in the creation of SGs to teach different engineering skills to engineering students and company employees. The learning objectives of these SGs are therefore a set of professional competencies in the specific engineering domain. Such SGs have been designed in our laboratory and used in our engineering school for more than 15 years (<http://liesp.insa-lyon.fr/nosSeriousGames>).

Currently, the use of SGs is still restricted because their creation is a very time-consuming and expensive process. Indeed, based on our design and production experience, one hour of SG costs about 15000 dollars. For these reason, we propose a multi-phase process to efficiently create SGs. Each phase of this process offers a set of interactive tools to help users accomplish their tasks in a visual and easy way. In this paper, we present EDoS (Environment for the Design of Serious Games), an interactive authoring environment that supports the designing phase and a part of the production phase. These are the central phases of our global production process going from the collection of the client's requirements to the actual execution and use of the game (Marfisi-Schottman, 2009). EDoS is indeed used after collecting the client's requirement and its output is a structural and formal scenario that is ready to be executed by our game engine in the final phase. EDoS provides SG authors with a method to create a SG faster and more efficiently, with a well organized process.

In the next section, we will introduce some existing authoring tools that can be used for designing SGs. The design principles of EDoS will be discussed in the third section. Then, the framework and design process of EDoS will be presented. Some screenshots of EDoS will also be used for the illustration. The conclusion and future works are depicted in the last section.

2. Related Work

Here is a selection of authoring tools that can be used for designing SGs. Their different characteristics and similarities with EDoS will be discussed in the next section:

E-Learning tools. Such tools help authors design e-learning courses. For example, Resource Center (Hoermann *et al.* 2005) features a web-based authoring toolset which allows the creation of SCORM-compliant courses; Editors like Reload (<http://www.reload.ac.uk/>), MOT+ (<http://www.licei.ca/LinkClick.aspx?link=1397&tabid=1393&language=en-US>), LAMS (<http://www.lamsinternational.com>), etc. allow creating IMS-LD compliant or IMS-LD close units of learning. In general, these tools aim at designing, managing and delivering learning activities; they are not enhanced with fun elements and characteristics specific to SGs. Moreover, tools, like Reload, are designed for pedagogical engineers, and not for teachers who cannot easily understand the structure of IMS-LD.

Interactive Digital Storytelling (IDS) tools. Such IDS can be used to create story-based scenarios of SGs. They provide an easy and efficient way to design interactive stories for various application domains, including entertainment and education. Here are a few examples of the existing tools. (1) Tools for conversation design like Façade (Mateas and Stern 2005), CrossTalk (Gebhard *et al.* 2003) and Scenejo (<http://scenejo.origo.ethz.ch>). Such tools allow specifying the structured dialogue between game story characters and can even support natural language interaction, based on keyboard input, with virtual characters (like Façade). (2) Tools for rapidly designing game scenes like U-CREATE (<http://www.u-create.org>), StoryTech (Göbel *et al.* 2008), etc. In such tools, the overall game story is partitioned into individual scenes or complex scenes that may contain other scenes. The user can drag and drop content elements (virtual characters, pictures, text, sound, animation, AV-clips, etc.) on each scene to create the layout representation. After, the authors define the logic that governs the flow of events in each scene by selecting actions from a list of predefined actions (speech action, set as target, etc.) and applying them to elements of the scene. Transitions between scenes are also "drawn" in terms of "arrows". Game screens can be "drawn" very fast using such tools.

Some other tools, ex. e-adventure (<http://e-adventure.e-ucm.es>) is an authoring environment for the creation of adventure games. The author can design virtual characters, conversations, player's avatar (as well as its movement trajectory) and scenes; but these tools are specific to adventure games that are usually built around a fantastic world in which the player must navigate, interact with diverse objects and virtual characters in order to solve multiple challenging puzzles. The navigational in-game world is made up by scenes, therefore, the creation of scenes and player's animations are two first tasks of authors. Another authoring tool, ScenGame (Pernin and Aquire 2009) allows specifying and associating, in a visual way, learning activities to adaptable mini-SG or SG components (Image Puzzle, MCQ, etc.) so that the game engine can execute them.

Although some authoring tools for SGs have been proposed, most of them do not take the specificities of engineer training into account. Moreover, there exist few tools which aim at reusing available and adaptable SG components. The IDS Tools such as StoryTech, only allow reusing small granularity resources, such as GUI elements (compass, text display, etc.); 3DCharacter, 2DAsset, etc. to create the layout of the game scenes. They do not allow adapting and orchestrating available SG components of different granularity levels to create large SG. The ScenGame tool allows it but it has some other inconvenient: (1) The scenario of ScenGame is mono-user (at least for the time being); (2) This tool works only at one granularity level: SCO. Indeed, in ScenGame, the author has to choose a SG component of SCO granularity level for each learning activity. This SCO level corresponds to the SCORM vocabulary: asset, SCO (sharable content object) and content aggregation (Dodds and Thropp 2004). In the next section, a set of design principles of our environment EDoS will be discussed to remedy these problems.

3. Principles of our authoring environment EDoS

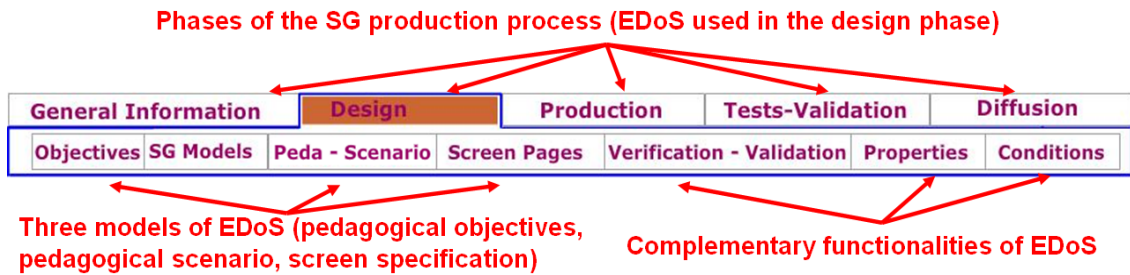
In order to meet the needs for designing engineer training SGs (at first for our school), we propose an interactive authoring environment called EDoS. This environment provides the SG authoring team with a method to create SGs faster and more efficiently, in a visual, structured and formal way. It is used after collecting the client's needs for the SG and it helps the collaboration between the pedagogues, who lack the IT abilities for the actual game development, and the IT developers, who do not specialize in pedagogy. In order to remedy the problems that have been discussed above, EDoS must integrate the following design principles:

- EDoS must take the specificities of engineer training SGs into account.
- EDoS must help authors achieve three purposes: (1) organise and formalize pedagogical objectives of an engineer training SG; (2) design the pedagogical scenario to achieve these specific objectives and (3) specify the Human Computer Interactions (HCI) scenarios for each screen of the SG to facilitate the IT developers' work. Some authoring tools (ex. e-learning tools) work only at a pedagogical scenario level. Others (ex. storytelling tool) work only at a screen level with their gaming content and multimedia elements. According to us, an authoring environment for an SG must include these two levels. Moreover, for the needs of academic engineer training, we feel the environment must also include a higher level concerning the professional competencies at which the SG aims.
- EDoS must support multi-roles pedagogical scenario in order to design multi-users and collaborative SGs. In an engineer training SG, groups of learners often play the SG while the tutor is present to help, examine or simply supervise them.
- EDoS must favour the reuse of SG components of different granularity levels to create large SG. This principle is important to reduce the cost and time of SG development.
- In EDoS, the independence between the pedagogical scenario of a SG and the various computing resources (SG components, web applications, etc.) used to execute it, must be assured. This principle increases the reusability and adaptability of the pedagogical scenarios.
- EDoS should provide familiar functionalities and manipulations of graphical editors such as: "drag and drop", "bring to front", "send to back" graphical objects, zoom in/out, save, etc.

Before EDoS, our authoring team had to use natural language to describe the pedagogical objectives, pedagogical scenario and HCI scenario of each screen of the SG. Developers had to read a thick document of this description to code the SG. EDoS allows authors to avoid such a tedious and inefficiency authoring method.

4. Framework and design process of EDoS

The design process of EDoS is based on three models: (1) a formal domain-specific model of pedagogical objectives (competencies, knowledge, and behaviours) at which a SG aims; (2) IMS-LD-SG, an extension of the IMS-LD specification made specifically for the SG and (3) a CTT task model (Concurrent Task Trees) used to formalize HCI scenario on each screen of the SG. The figure 1 below illustrates the functional tabs present at the top of the screen of the EDoS environment.



4.1 Domain-specific model of pedagogical objectives

After collecting the client's requirements, the SG design begins by defining the target pedagogical objectives of the SG. The objectives of an engineer training SG are a set of professional competencies for engineer's job of a certain domain. As the result, we have defined a formal domain-specific model of pedagogical objectives to represent them. This model is currently used to design SGs at our engineering school. The figure 2 below illustrates a screenshot of EDoS that allows authors to define the pedagogical objectives of the SG. On this figure, the objectives of the SG named Samouleau (designed to teach the "Lean" production management methodology) are presented.

Games >> Samouleau >>

General Information | **Design** | **Production** | **Tests-Validation** | **Diffusion**

Objectives | **SG Models** | **Peda - Scenario** | **Screen Pages** | **Verification - Validation** | **Properties** | **Conditions**

General Objectives: Teaching production management methodology Lean

Formation Time: 300h

Target Public: 3rd or 4nd year engineers

Competences to acquire:

Add competence | Add knowledge | Add behaviour | Save objectives

- C1 - Identify and use the global indicators of a company
 - K1 - Diagnose a company (Threshold=100)
 - K2 - Correspondance I.O. / Domain (Threshold=90)
 - K3 - Finding the operation indicators (Threshold=90)
 - K4 - Lean indicators (Threshold=80)
 - B2 - Quality process (2 times)
 - B4 - Collect useful information (2 times)
 - B5 - Use the pertinent tool (1 times)
 - B6 - Respect the confidentiality of information (1 times)
 - C2 - Know how to do a general Lean analysis (Threshold 1)
 - K5 - Hierarchically organization of the company (Threshold=72)
 - K6 - BCG matrix (Threshold=100)
 - B1 - Follow the instructions (1 times)
 - B2 - Quality process (1 times)
 - B3 - Organize one's actions (2 times)
 - B4 - Collect useful information (2 times)
 - B5 - Use the pertinent tool (1 times)
 - B6 - Only bother people when useful (1 times)
 - B7 - Take initiative (1 times)
 - B8 - Respect the confidentiality of information (1 times)
 - C3 - Know how to do a general Lean analysis (Threshold 2)
 - C4 - Manage a deep investigation
 - C5 - Establish a sector-based diagnose to determine the operational objectives

Knowledges:

Knowledge

- K1 - Diagnose a company
- K2 - Correspondance I.O. / Domain
- K3 - Finding the operation indicators
- K4 - Lean indicators
- K5 - Hierarchically organization of the company
- K6 - BCG matrix

Add knowledge | Modify

Behaviours:

Behaviour

- B1 - Follow the instructions
- B2 - Quality process
- B3 - Organize one's actions
- B4 - Collect useful information
- B5 - Use the pertinent tool
- B6 - Only bother people when useful
- B7 - Take initiative
- B8 - Respect the confidentiality of information

Add behaviour | Search behaviour | Modify

Figure 2: Defining the pedagogical objectives with EDoS

The objective model is composed of domain-specific target competencis. Learners play the SG to acquire these competencies. Each competency is composed of a coherent set of knowledge and behaviours.

As illustrated on the figure 2, each piece of knowledge, identified by a Ki, is composed of a title and a point scale (Min, Max) to evaluate the learning degrees of this knowledge in the SG. Each type

of behaviour, identified by a Bi, is composed of a title and a scale of test number (MinNb, MaxNb) to express the minimal and maximal test number of this behaviour that learners can do in the SG. For each competency, a local threshold point is given to indicate the minimal value in order to validate each piece of knowledge in this competency. Similarly, if the behaviour appears in several competencies, then for each competency, a local test number is used to specify the minimal test number in order to validate this behaviour in this competency. For example, on the figure 2, for the C1 competency of the Samoulean SG, the learner must obtain at least 90/100 to validate the K2 knowledge and prove at least 13 times the right B1 behaviour to validate it. A competency is validated if all its pieces of knowledge and behaviour are validated.

This model constitutes the cognitive base of a SG. Once the pedagogical objectives are specified, the pedagogical scenario must be elaborated very precisely to achieve them. Our second model is used for this purpose.

4.2. Pedagogical scenario of SG

The model of our pedagogical scenario is inspired and derived from the IMS-LD model (<http://www.imsglobal.org/learningdesign>). Indeed, we have adapted IMS-LD to the pedagogical scenario design of SG.

4.2.1. Why IMS-LD?

IMS-LD was initially developed at the Open University of the Netherlands, with the IMS-LD specification that was released by the IMS Technical Board in February 2003. IMS-LD is a formal pedagogical standard to design units. Here are the reasons why we consider IMS-LD as a good base to build a specific model for SGs.

- IMS-LD is pedagogically neutral. It supports all theories of learning and it does not mandate a specific pedagogical approach (according to the claim of authors).
- As opposed to IMS-SS/SCORM (<http://www.imsglobal.org/simplesequencing>) that focuses only on the organization of learning activities, IMS-LD takes the support activities of teachers into account. Moreover, IMS-SS is based on a single learner model, whereas IMS-LD supports multi-learner situations such as group and collaborative learning processes to be modelled (Tattersall *et al.* 2003). This is also one of design principles of EDoS presented above.
- According to (IMS-LD 2003), the *Properties* and *Conditions* of IMS-LD level B enable personalization and more elaborate sequencing and interactions based on learner portfolios. They can be used to direct the learning activities as well as record outcomes. The separation of *Properties* and *Conditions* into a separate schema also enables them to be used independently of the rest of the IMS-LD specification. This is typically as an enhancement to IMS-SS/SCORM.
- The independence between the pedagogical scenario of a SG and the computing resources (SG components, web applications, etc.) used to execute it, is assured in IMS-LD. Indeed, pedagogical activities and computing resources are separately defined. This is one of the design principles presented above of EDoS to increase the reusability and adaptability of pedagogical scenario.
- IMS-LD allows designing SGs as reusable activities. This is a great advantage for our European project LGF (Learning Games Factory) for which the objective is to create a large database containing various reusable SG components.
- The definition of activities and their realizations (moment, place) are independent. Indeed, in IMS-LD, activities and role-parts (concerning the moment and place in the scenario where these activities are realized by roles) are distinct. This feature allows authors changing the arrangement of the activities' realization in organisational units of the scenario (*play*, *acts*, etc.) without affecting the activities' definitions. In IMS-LD, the navigation between activities can also be changed in a static or dynamic way.

4.2.2. Our model of pedagogical scenario IMS-LD-SG

We modified IMS-LD to design SG scenarios. Similarly to IMS-LD, our SG scenario is composed of pedagogical *activities* - central element of our model. There are two basic types of activities

(*learning and support activity*) that are realized respectively by two basic types of roles (*learner and staff*). We retain other elements of IMS-LD: *Play, act, role-part* (that define the static structure of the scenario); *Condition, Properties* (that allow specifying dynamic and conditional evolution of the scenario); *notification* (to describe event flows of the scenario).

In the SG pedagogical scenario, the relations between scenario objects and pedagogical objectives (defined in the objective model) must be specified. In IMS-LD, the *activities* are concurrently realized by *roles* (*role-part*, ex. learner does Image Puzzle, MCQ, etc.) in sequential *acts*. These *acts* and *activities* are only enough to teach and validate particular knowledge and/or behaviours. Therefore, we propose a new higher level called *module*. This module level is intermediate between play and act level. A *module* is composed of a set of *acts* to validate involving a set of competencies. This level is useful, especially for our engineer training SG for which the story is usually organised around the competencies. Three types of sequences can be used to design the module flow: sequential, Or (choice) and Exclusive Or (exclusive choice).

Nevertheless, in IMS-LD, there are only two types of resources: *webcontent* and *imsldcontent*. This is not sufficient for our SGs. In the LGF project, a collection of SG components, coming from different suppliers, can be assembled to create large SGs. These components can be harvested and adapted via a web-based protocol (called *LGF protocol*) in order to be associated to units of the SG scenario (*play, module, act, activity*), depending on their granularity level. These components will be orchestrated by the execution engine to run the whole scenario. Therefore, a new type of resource must be added: *LGFComponent*.

In order to design SG scenarios, authors must start by specifying the pedagogical objectives (specified above) and the type of SG they want to use. For example, Samouleane is a Role-Playing type SG in which the learner must assume a Japanese Samurai character to study and diagnose a company using Lean production methodology. The learner must face 5 challenges to accomplish his/her mission. Each challenge aims at a core competency. For this game the authors specified five sequential *modules* to validate the five competencies of Samouleane. The figure 3 below illustrates those five modules.

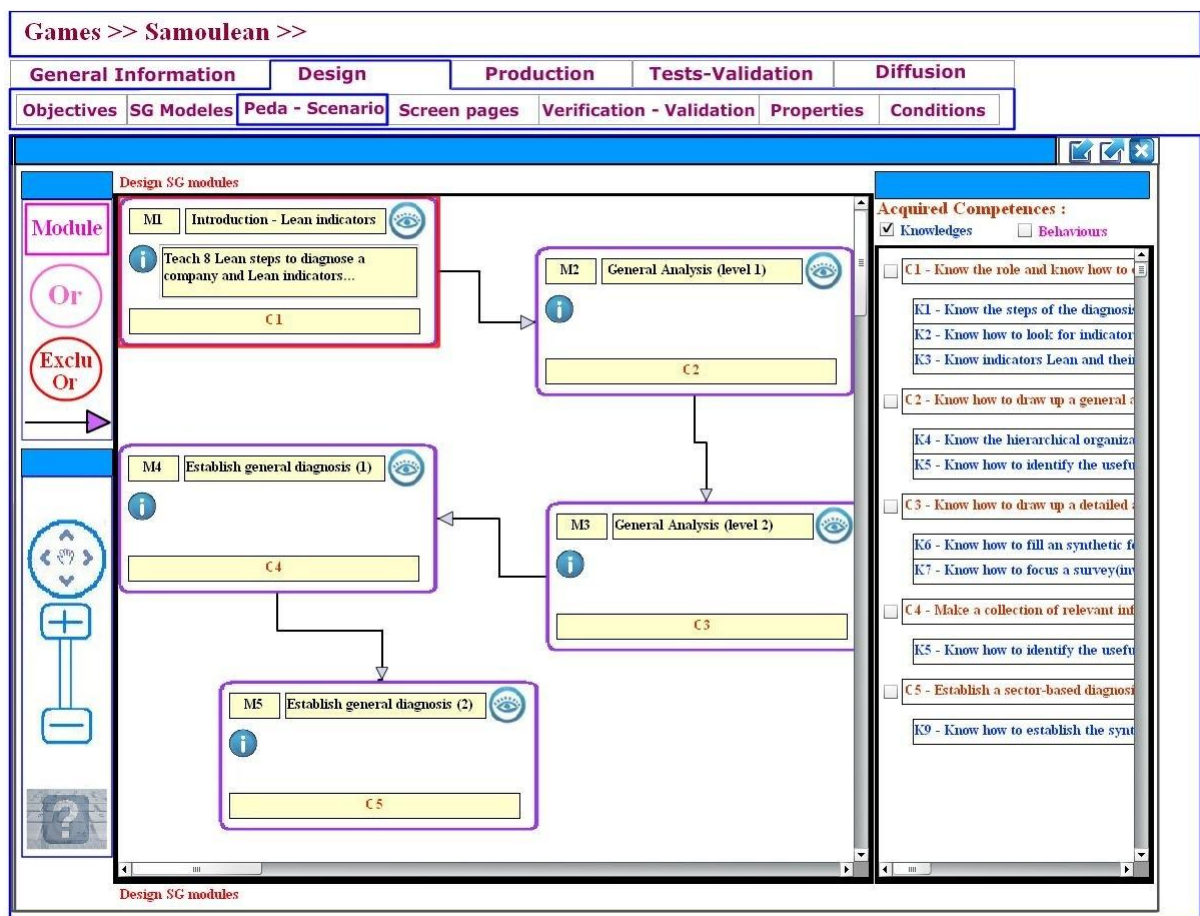


Figure 3: Description of SG modules with EDoS

The module 1 is currently selected and it contains two sequential acts. The first one is only an introduction of SG and module, the second one, identified as Act2.M1, get involved to four learning activities that the learner must achieve to overcome this challenge, as illustrated on the figure 4. After the authors are satisfied with their scenario, the units of the scenario (*play, module, act, activity*) can be associated to computing resources (web applications, SG component) of different suppliers to be executed. The figure 5 below illustrates an example of such an association. Activity LA4 involves the learner in a testing phase. To do such an evaluation, the authors can choose various existing computing resource items available in the resource database. In this case, the author has chosen an adaptable MCQ SG component (active status) that provides a MCQ (Multiple choices questions) test.

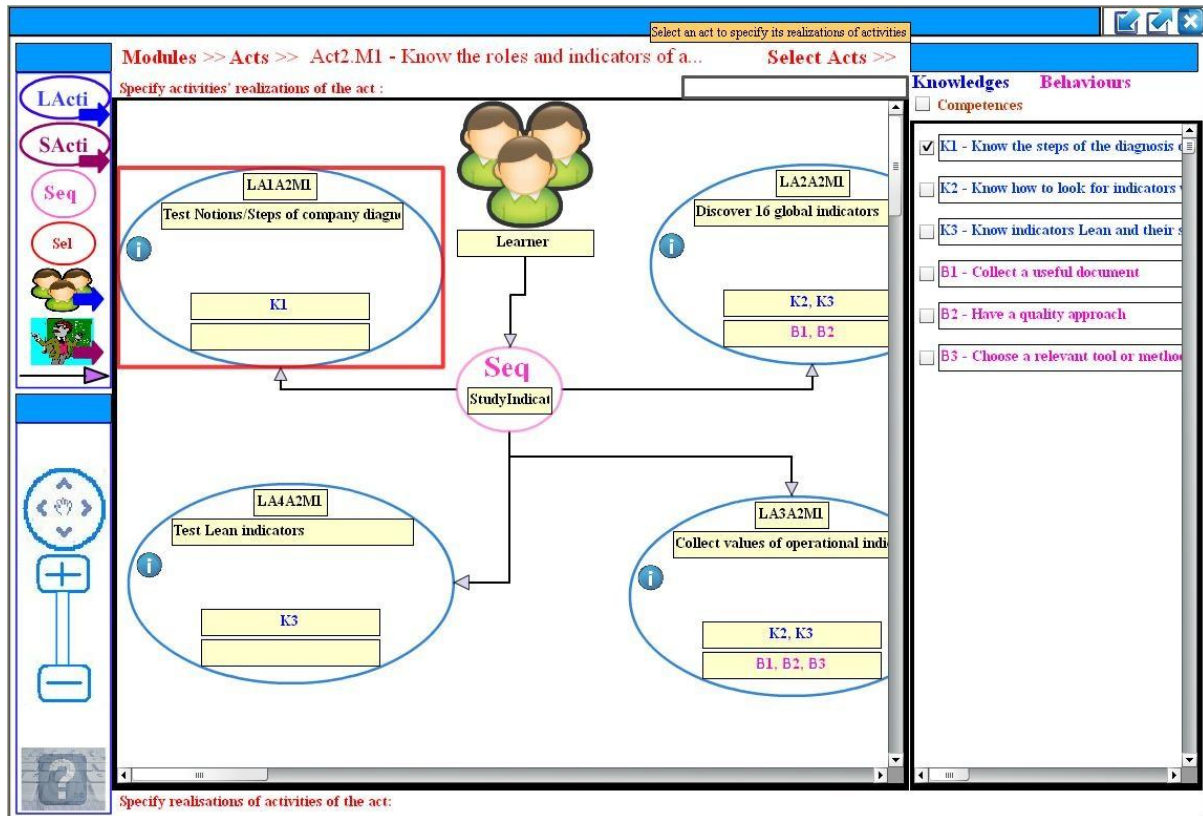


Figure 4: Description of activities with EDos

Configure the execution of the activity LA4

Information on this activity :

ID : LA4A2ML

Title : Test Lean indicators

Description : In this activity, learner must pass a test on Lean indicators. This is the last activity to validate the competence Q1: Know the role and know how to employ global and operational indicators of a company

Configure conditions concerning the accomplishment of this activity :

Conditions to complete this activity

Actions to perform after completion

Specifier a resource item to execute this activity :

Title	Type	Provider	Status	AL
MCQ LIESP	Generic LGF Component	LIESP INSA Lyon	Inactive	SCO
MCQ LIESP LeanIndicators	Parameterized LGF Component	LIESP INSA Lyon	Active	SCO
MCQ LIG	Generic LGF Component	LIG Grenoble	Inactive	SCO
Image Puzzle	Generic LGF Component	LIESP	Inactive	SCO
Introduction to Samoulean game	Web content	LIESP	Inactive	None

Figure 5: Description of the execution of an activity with EDoS

The scenario we have seen so far describes the pedagogical content of the SG but it must still be enhanced with fun elements and characteristics (amusing interactions, characters, attractive competition, etc.). These elements will be specified by the task model, presented in the next section.

4.3. Screen and Game-Play Description of SG

Unlike the two above models that represent the pedagogical aspects and type of SG, the model at this level concerns the game-play aspect of SGs. After specifying pedagogical activities in the scenario, the authors need to give a detailed description for each activity. For this reason, we propose a finer level to describe the screens with which the users will interact to accomplish each activity. These screens must involved gaming elements and amusing HCI scenarios to capture the learners' attention and inspire them. Before EDoS, authors had to use natural language to describe each screen and developers had to read and understand these textual descriptions to implement them. This non-visual method usually leads to quite a few misunderstandings and therefore time loss. EDoS helps authors design the screen layouts and specify their HCI scenario in a visual and formal way.

In this next section, we will present an example concerning the currently selected activity on the figure 4 above. This activity is a test concerning the necessary steps and notions to diagnose a company. It contains two screens. The figure 6 illustrates another screenshot of EDoS. Authors can "drag & drop" gaming elements (virtual characters and objects, background) to arrange the screen's layout for this activity. Naturally, the same gaming element can appear in several screens.

Design layout representation and HCI scenario of activities' screens :

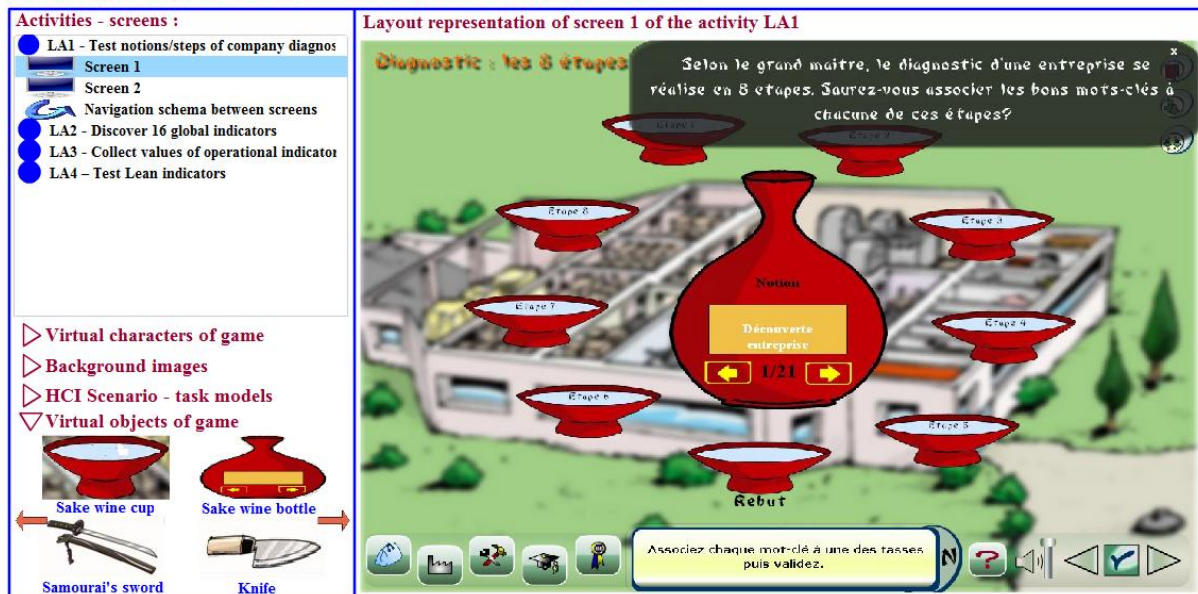


Figure 6: Elaborating each screen's layout representation of different activities

The learner must interact with each screen to accomplish the activities. We propose the CTT task model (Concurrent Task Trees) to visually and formally represent HCI scenarios. This representation facilitates the communication between authors and developers and it is also the first step towards an automatic generation of code prototype for each screen in order to partially support the developers' tasks. After all the screens of an activity are fully implemented by the developers, they will be assembled in a computing resource item (a LGF component or a web application) that will be launched by the engine in order to execute this activity. Authors can also associate an activity to an existing reusable computing resource without describing its screens if they can find a suitable one to execute this activity (cf. figure 5 above).

A task model is a description of the task sequences that a user can do by interacting with the system to reach a certain goal. Many task models have been proposed but we chosen the CTT task model (Paternò *et al.* 1997) in which tasks are hierarchically organised with temporal relations between them. We choose the CTT model in order to formally and visually describe HCI scenario of screens for our SGs because of the following advantages:

- The CTT notation is legible, visual and easy to understand.
- This CTT model can be used to develop tools that analyse and evaluate system's HCI and implementation using users' traces. This is content of evaluation phase in our SG production process. Some tools that are based on CTT to analyse users' traces have been proposed ex. WebRemUSINE (Paganelli and Paternò 2002), MultiModal Usine (Paternò *et al.*, 2006), etc.
- Most of other task models aim at mono-user. Even when they consider various user types, they suppose that these users do not have access at the same moment. The CTT model allows designing cooperative aspects and tasks that involve several actors, in order to accomplish a certain cooperative goal. On the other hand, IMS-LD (as well as our pedagogical scenario model) does not support any mechanisms that allow describing interactions between members of a group in a collaborative activity (Hernández Leo *et al.* 2004), although interactions between groups can still be described by IMS-LD. Therefore, the CTT model is a good complement for our pedagogical scenario.

The figure 7 below illustrates the CTT task model that describes the HCI scenario of the screen on the figure 6. According to it, the learner can put each notion, represented by a form in a bottle of sake wine, into one of the eight cups of sake wine (centre of screen), each corresponding to the eight steps of the company's diagnosis. The number of correct associations is used to evaluate the learner.

Design layout representation and HCI scenario of activities' screens :

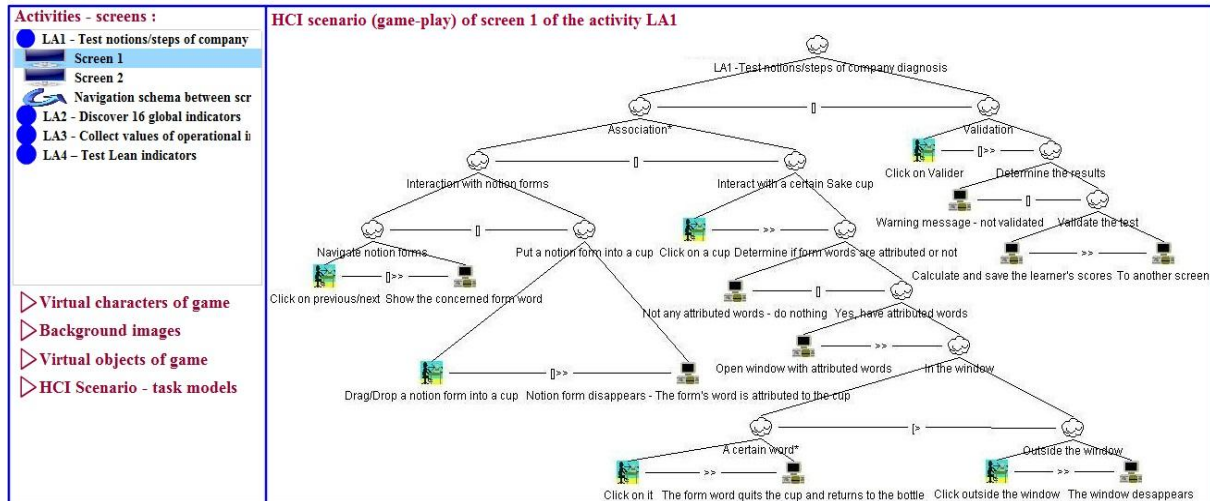


Figure 7: HCI scenario of the screen of Samoulean to test Lean steps/notions

This third model enables authors to embed fun and amusing aspects in the SG content; therefore, the attractiveness of the SG depends mainly on the choices made by the authors at this level.

5. Conclusion and future works

In this paper, we present an authoring environment, called EDoS, in order to design SGs. This environment is based on three models: (1) a formal domain-specific model of pedagogical objectives at which a SG aims; (2) a pedagogical scenario that is derived from the IMS-LD the SG in order to describe the pedagogical content of the SG; (3) the CTT task model used to formalize the HCI scenario of the screens of SG after arranging their layout representation. This model is a good complement for our pedagogical scenario model because it allows enhancing pedagogical activities with gaming elements and fun characteristics in order to captivate the learner's attention. Our approach aims at creating flexible and adaptable SG scenarios for engineer training SGs (used in our engineering school). We have validated this approach by applying it to describe Samoulean, a SG being developed at our laboratory to teach the "Lean" production management methodology to engineer students.

The first version of an execution engine has been developed in our laboratory but it is not yet made specifically to harvest, adapt and execute SG components shared by the different members of the "Learning Game Factory" project. This task is currently being done.

In order to design a SG, authors using EDoS will be able to start "from scratch" or use "templates". Indeed, we intend to define templates for each type of SG (role-playing game; board games, etc.) to reduce the time and cost of SG design. The functional tab "SG models" of the figure 1 corresponds to this work. Currently, our work is focused on the development of the authoring environment to answer the specific needs of role-playing games and in particular to teach engineering skills but the models we use are certainly applicable to other domains.

In this paper, the template of Role-Playing SG is partly shown thanks to the Samoulean SG. In such a type of SG, the players assume the role of characters in a fictional story to accomplish their mission. In order to achieve this goal, they must face and overcome several challenges through which the pedagogical objectives (competencies, knowledge and behaviors) are validated. In Samoulean, each core competency is modeled in one module and the story is composed of sequential modules and acts. This linear scenario is not an obligation and other SGs can use different models. An important future research objective of our team is to find appropriate scenario templates of each SG type.

Acknowledgements

Our research work is one of principal tasks of the "Learning Game Factory" project (www.learning-games-factory.com) supported by the European Community (FEDER). The authors gratefully acknowledge the support of this institution.

References

- Alvarez, J. (2008) *Serious games, Advergaming, edugaming, training and more*, IDATE 2008, BP 4167, 34092 Montpellier Cedex 5, France.
- Dodds, P. and Thropp, S. E. (2004) *SCORM content aggregation model, version 1.3.1*, Report of the Advanced Distributed Learning Initiative, July 2004
- Gebhard, P., Klipp, M., Klesen, M. and Rist, T. (2003) *Authoring scenes for adaptive, interactive performances*, Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia.
- Göbel, S., Salvatore, L., Konrad, R.A. and Mehm, F. (2008) *StoryTec: A Digital Storytelling Platform for the Authoring and Experiencing of Interactive and Non-linear Stories*, Proceedings of the 2008 International Conference on Automated solutions for Cross Media Content and Multi-channel Distribution, Florence.
- Hernández-Leo, D., Asensio-Pérez, J. and Dimitriadis, Y. (2004) *IMS Learning Design Support for the Formalization of Collaborative Learning Patterns*, Proceedings of the 4th International Conference on Advanced Learning Technologies, Finland.
- Hoermann, S., Hildebrandt, T., Rensing, C. and Steinmetz, R. (2005) *ResourceCenter - A Digital Learning Object Repository with an Integrated Authoring Tool Set*, In Kommers, P., Richards, G. (eds.) Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005, AACE, Chesapeake, VA, pp. 3453-3460.
- IMS-LD (2003), *IMS Learning Design Information Model, Version 1.0 Final Specification*, IMS Global Learning Consortium, Inc
- Marfisi-Schottman, I., Sghaier, A., George, S., Tarpin-Bernard, F. and Prévôt, P. (2009) *Towards Industrialized Conception and Production of Serious Games*, ICTE International Conference on Technology and Education, Paris, 25-27 June, France.
- Mateas, M. and Stern, A. (2005) *Structuring Content in the Façade Interactive Drama Architecture*, Artificial Intelligence and Interactive Digital Entertainment (AIIDE) Conference, Los Angeles.
- Paganelli, L. and Paternò, F. (2002) *Intelligent Analysis of User Interactions with Web Applications*, Proceedings of ACM IUI 2002, San Francisco, CA
- Paternò, F., Mancini, C. and Meniconi, S. (1997) *ConcurTaskTress : A Diagrammatic Notation for Specifying Task Models*, Proceedings of Interact'97, Chapman & Hall, London, pp. 362-369.
- Paternò, F., Piruzza, A. and Santoro, C. (2006) *Remote Web usability evaluation exploiting multimodal information on user behaviour*, Proceedings CADUI 2006, Bucharest, Romania
- Pernin, J.P and Aquire, J.L (2009) *Presentation of team MeTAH, LIG, Grenoble, France*, plenary reunion of the European project LGF, December 2009, France
- Tattersall, C., Manderveld, J., Hummel, H., Sloep, P., Koper, R. and de Vries, F (2003) *IMS Learning Design Frequently Asked Questions, version 1.0*, Educational Technology Expertise Centre, The Open University of The Netherlands.