Teaching with Game Based Learning Management Systems: Exploring and observing a pedagogical dungeon.

Thibault Carron, Jean-Charles Marty, Jean-Mathias Heraud

Laboratoire SYStèmes COMmunicants de l'Université de Savoie
Thibault.Carron@univ-savoie.fr
Jean-Charles.Marty@univ-savoie.fr
JM.Heraud@esc-chambery.fr
Abstract: The work reported here takes place in the educational domain. We propose an approach based on a learning environment based on a graphical representation of a course: a pedagogical dungeon equipped with the capacity for collaboration in certain activities. The emergence of online multiplayer games led us to apply the following metaphor to these digital work environments: the way of acquiring knowledge during a learning session (a lesson) is similar to the exploration of a dungeon where each student collects knowledge related to a learning activity. In the first part, the paper focuses on the description of how concepts of learning activities can be represented in the dungeon view. In the second part, we deal with the support of the observation task for the teacher during a learning session and more generally with providing the users with awareness. We thus propose a multi-agent system using data collected from traces resulting from the collaborative learning activity. We then describe our game based application: the way the teacher creates a dungeon according to the general pedagogical goal of the session and the use of the dungeon from the student’s and the teacher’s point of view are described. Finally, this environment allowed us to set up experiments with students in our University. The feedback about these experiments is discussed at the end of the paper.

Keywords: collaborative activity; learning environment; online multiplayer game; trace observation.
1 Introduction

Nowadays, learning management systems (LMS) offer functionalities that are recognized as being valuable from different points of view. For instance, students can learn at their own speed. These environments also allow the teacher to evaluate specific activities in a uniform way. However, although these environments enable powerful features, they also incur two major kinds of criticism. The first one deals with the non-attractiveness of such environments for the students, as very often students tend to consider them as unexciting. The second one copes with the lack of awareness (see Greenberg (1996) for a definition of awareness) from the teacher’s point of view as shown by Kian-Sam (2002): s/he no longer has the usual and helpful student’s feedback anymore (eyes, general attitude). As reported in Hijon and Carlos (2006), where the authors compare the built-in student tracking functionality of various CMS tools, this functionality is far from satisfactory. The regulation of the activity is thus much more difficult.

Concerning the first point, agreeing with Vygotski’s school of thought and activity theory, we consider that the social dimension is crucial for the cognitive processes implied in the learning activity. Consequently, the question was how to enhance the social dimension in such environments.

Observing the emergence and success of online multiplayer games with our students –the so-called “digital natives”-[Summit on educational Games, October 2006 (http://www.fas.org/gamesummit/)], more generally in the world (Rosenbloom 2004) and even in education (Purdy (2007)), (Scott (2007)), it was decided to use it as a support for our course. This led us to apply the metaphor of exploring a virtual world, a dungeon, where each student collects knowledge related to a learning activity. It is our view that the way to acquire knowledge during a learning session is similar to the exploration of a dungeon. This approach reveals advantages such as a recreation-type process, a large usability of the tool or its adaptation to the student’s speed. Such game based learning environments can thus be proposed as a way of implementing learning sessions, in which teachers can prepare and follow a pedagogical scenario (see Kinshuk (2006) for a definition of a pedagogical scenario).

Concerning the second point, for usability purposes, it is essential that Computer Based Education offer the possibility of monitoring the activity performed by the students and of obtaining information or feedback about it. For example, being aware of the learning progression of each student is an important goal for the teacher. Here, we explain how we can avoid the loss of perception for the teacher in these environments.

In this article, we propose a Game Based LMS called a pedagogical dungeon equipped with cooperation abilities for particular activities (see Dillenbourg (2006) for a list of cooperation abilities). The second section demonstrates the links between pedagogical concepts and their representation in the dungeon. The following part concerns observation of what is occurring in the dungeon: some awareness indicators are provided thanks to specific observation probes. Moreover, we present some means of providing feedback and retrieving the perception thanks to activity trails or traces left by the users (see Marty (2007) for details on these traces). In the fourth part, the system itself is described: we focus on activity preparation and the generation of the dungeon, explaining our metaphor more precisely, describing each artefact. We then explain the enactment of the game. The last section deals with description of experiment lessons.
2 Links between a learning session and the objects of the dungeon.

We have chosen to derive a set of principles from a formal theory of Human Work Activities called Activity Theory (see Dunne (1995) for a definition of Activity Theory) issued from Vygotski (1934) proposals. In this theory, the social dimension is crucial for the cognitive processes involved in the learning activity. A learning activity consists of one or more (sub) activities linked and ordered to achieve a given pedagogical goal. Actors (students or teachers) can perform these (sub) activities when their associated conditions (or prerequisites) are satisfied. They carry out these activities in collaborative spaces called arenas, through social interactions or through personal actions. An activity is mediated by tools (such as communication tools or evaluation tools) and uses artefacts (defined in Dunne (1995)).

To enhance this social dimension, we have chosen to put the students together in a common virtual environment during the entire learning process. In order to link the game world to the learning one and according to Hainley (2006), we propose in this section to link the objects used in our game based framework with the concepts that we usually find in a learning system. Table 1 summarizes these links.

<table>
<thead>
<tr>
<th>Classical concept in the activity theory</th>
<th>Corresponding representation in our Game Based LMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arena / Collaborative space</td>
<td>Dungeon for the learning activity</td>
</tr>
<tr>
<td></td>
<td>Room for a (sub) activity</td>
</tr>
<tr>
<td>Link between activities</td>
<td>Corridor</td>
</tr>
<tr>
<td>(sub) Activity</td>
<td>Crystals (Exercises)</td>
</tr>
<tr>
<td>Condition / Requisite</td>
<td>Room Door</td>
</tr>
<tr>
<td>Resources</td>
<td>Knowledge Spheres</td>
</tr>
<tr>
<td>Assessment, Validation</td>
<td>Door Key</td>
</tr>
<tr>
<td>Communication tool</td>
<td>Chat window</td>
</tr>
<tr>
<td>Persons</td>
<td>Avatars (teachers, students)</td>
</tr>
</tbody>
</table>

Table 1: Correspondence between AT Concepts and Game based LMS Representation
Decomposition of a learning session: Rooms and topology

The learning session (or learning activity) is very often split into different activities. It is the case when the teacher proposes to her/his students a set of exercises linked together in order to reach a pedagogical goal. Each activity has its own local goal, generally a concept to acquire. For a student, performing all the activities ensures that s/he has reached the general goal of the session, i.e. s/he has gained the knowledge associated with the session.

The dungeon represents the place where the learning session takes place. A particular dungeon is dedicated to a particular learning activity, for a particular subject. Each room of the dungeon represents the place where a given (sub) activity can be performed. The dungeon topology represents the overall scenario of the learning session, i.e. the sequencing between activities. There are as many rooms as actual activities, and rooms are linked together through corridors, showing the attainability of an activity from other ones. An example of a scenario seen as a dungeon topology is presented in figure 1.

![Figure 1: An example of a scenario seen as a dungeon topology](image)

Actors (Students or teachers) can move through the dungeon, performing a sequence of sub activities in order to acquire knowledge. Activities can be carried out in a personal or collaborative way: you can access knowledge through documents, via help from teachers, or from work with other students. As we will see later, the dungeon can be flexible. For instance, “teleportation portals” can lead to new rooms created dynamically for instance.

Achievement of activities

Each room is dedicated to an activity. You can find explanatory resources such as texts, links, and videos. These provide the student with useful information. The student reaches the local goal of the activity if s/he answers a quiz successfully. This quiz is thus also located in the room. In Figure 13, we can see an example of a room in the dungeon.
As users move through the dungeon, they can meet other students or teachers involved in the same session. When a student is in the same room as another student, it only means that these students are performing the same activity. They can of course access the resources at the same time.

The teacher may want several activities to be collaborative. In that case, the rooms associated to these activities are collaborative places. Currently, a chat facility is provided in the dungeon rooms, but we can of course imagine other collaborative tools available in these rooms (shared space, forums, etc.). If the teacher uses collaborative work in a session, s/he must set up teams of students: students belonging to the same team are supposed to carry out collaborative activities together. In collaborative rooms, the quiz is also collaborative. Students in the same team must all be present in the room. They may exchange via the chat before answering the question.

As in “traditional classrooms”, a student may also collaborate with a teacher, for instance if s/he needs help from her/him.

### Sequencing of activities

Each room can be accessed through doors. These doors are the guards of the activity. They ensure that the student has the necessary prerequisites to perform the activity correctly. When users answer a quiz correctly, the associated key is obtained. In the event of a correct answer given for a collaborative quiz, a collaborative key is provided to all the members of the team.

Activities must not necessarily be ordered in the dungeon. However, most of the time, they are well-ordered: it is quite rare for a teacher to provide the students with a set of exercises without any order. By ordering the activities, teachers may want either to define an order representing a progressive approach to the general goal of the session (logical order), or simply to force the group to carry out the activities in the same order with the purpose of following the students more easily (temporal order). When users play out a session in the dungeon, this ordering is ensured by the fact that they have to have obtained the key of previous activities before entering a new room.

But these conditions are not sufficient to ensure the correct progress of the pedagogical activity. As stated previously, the teacher is used to perceiving some informal information in a real classroom. S/he often wants to observe specific information or behaviour and providing her/him with the expected feedback on the overall activity is still a research problem. Our approach consists in taking advantage of the traces left by the actors participating in the mediated learning activity in order to calculate awareness indicators for the pedagogical dungeon. We describe this idea more precisely in the next section.

### 3 Observation and traces of collaborative activities

In order to obtain an acute awareness for the actors, we first situate the kind of expected needs according to a categorization of awareness. We then describe how such a system can be observed by agreeing on a certain number of basic observation features. Finally, we set up a way for combining easily observed basic events to notify more complex events to the users.

#### 3.1 Awareness needs in such an environment

Teachers using such educational environments need a means of following the progress of the learning activity accurately and of avoiding the terrible lack of feedback already mentioned in the introduction of this paper. Generally, the teacher in a virtual classroom perceives less information than in a real classroom since s/he does not see her/his students. We must first point out that it is possible to perceive partially thanks to relatively classic (in the video game domain) and basic awareness indicators such as the ones proposed in Nova (2007). As we want to deal in more depth with how the
problem of awareness is tackled in the pedagogical dungeon, we propose to examine the different facets which make up the categorization of awareness proposed by Greenberg (1996) (see figure 2).

Figure 2: Awareness Categories from Greenberg

• **Informal awareness**: It concerns the general consciousness of others’ presence, capability or availability, all things that are implicitly felt when the teacher is in a real classroom. Part of this awareness is brought about by the dynamic representation of users through avatars, but little information can be inferred from the behaviour of an avatar in the game, for instance the fact that an avatar is frozen in the same activity for a long time (total inactivity).

• **Social awareness**: It deals with the feeling that is given by another person in a dialogue context. Is s/he listening? Is s/he interested by the discussion? What is her/his emotional state? These pieces of information are generally obtained thanks to non-verbal indicators such as facial expression, look or body behaviour in a real classroom. Social awareness is often supported by webcams. No feedback about this is integrated in the Graphic User Interface (GUI) of the pedagogical dungeon.

• **Group-structural awareness**: It concerns the organisational consciousness role, responsibility or status of each user. The task realisation process may also be taken into account here, too. In the pedagogical dungeon, the user who has the teacher’s role is able to create groups and to place students in them. Some indicators are given through the interface about group-structural awareness: the notification message of group joining, the specific representation of the teacher: teachers’ avatars are magicians.

• **Workspace awareness**: The most intuitive aspect of awareness. In fact, it represents the group activity consciousness (of each user) relative to their workspace. Major points concern their personal identity information, their position, their activity and sometimes-immediate changes that they are making in the workspace. In the dungeon, a user is represented with an avatar and her/his name; the discussion on the chat window is displayed: all the participants are identified by their name; the user position is displayed both in the room and on a spatial representation of the dungeon called the mini-map (see figure 18); the colour of each room on the mini-map reminds the user of her/his own progression: succeeded, failed or not visited; the behaviour of the crystal in the main game window also provides this kind of awareness; and help calls as well as answer propositions for an activity validation are instantly displayed with a notification window on the teacher’s screen.

As shown in the literature, most of the work on awareness focuses on workspace awareness; particularly Greenberg (1996) who defined the concept, mainly worked on solutions in this domain. For the moment, it is also the case in our software: we mainly dealt with workspace awareness through our graphic user interface, as explained in section 4.

However, we can provide teachers with an improved awareness since computers provide the possibility of obtaining much more interesting information: the visualisation of traces left by users during their activity within the system. We have already proposed some trace visualisations as a solution to specific observation problems in Carron (2006), France (2005), France (2006), Heraud (2004) and Marty (2007). The purpose here is to be more general and offer observation features on any pedagogical tool. The following paragraph explains how this problem is addressed in our approach.
An observation API

We have chosen an approach which is as generic as possible and thus possibly independent from our application. The idea is to equip any application (here, the whole of the pedagogical dungeon) with a tracing possibility. This implies the definition of an API of required basic observations. For instance, in the dungeon, actions such as “entering a room” or “correctly answering a quiz” may be traced and thus collected by specific elementary probes.

Basically, in our context, we defined 17 elementary probes that may be flagged at any moment by any client of our application (see table 2 and figure 3).

<table>
<thead>
<tr>
<th>Probe Name</th>
<th>Parameters</th>
<th>Awareness category</th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkshopArriving</td>
<td>&lt;UserName&gt;, &lt;WorkshopName&gt;</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>WorkshopLeaving</td>
<td>&lt;UserName&gt;, &lt;WorkshopName&gt;</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>WorkshopAnswering</td>
<td>&lt;UserName&gt;, &lt;WorkshopName&gt;</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>WorkshopAnsweringWithContent</td>
<td>&lt;UserName&gt;, &lt;WorkshopName&gt;, &lt;AnswerContent&gt;</td>
<td>Social awareness</td>
</tr>
<tr>
<td>WorkshopTeacherValidating</td>
<td>&lt;UserName&gt;, &lt;WorkshopName&gt;, &lt;TeacherName&gt;</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>WorkshopTeacherValidatingWithContent</td>
<td>&lt;UserName&gt;, &lt;WorkshopName&gt;, &lt;TeacherName&gt;, &lt;Comment&gt;</td>
<td>Social awareness</td>
</tr>
<tr>
<td>WorkshopCorrectlyAnswering</td>
<td>&lt;UserName&gt;, &lt;WorkshopName&gt;, &lt;Boolean&gt;</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>StudentConnecting</td>
<td>&lt;UserName&gt;</td>
<td>Informal awareness</td>
</tr>
<tr>
<td>HelpConsulting</td>
<td>&lt;UserName&gt;, &lt;HelpName&gt;</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>Chatting</td>
<td>&lt;UserName&gt;, &lt;ChannelName&gt;</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>ChatContentListening</td>
<td>&lt;UserName&gt;, &lt;ChannelName&gt;, &lt;SentMessage&gt;</td>
<td>Social awareness</td>
</tr>
<tr>
<td>GroupCreating</td>
<td>&lt;GroupName&gt;, &lt;GroupType&gt;, &lt;UserName1&gt;, &lt;UserName2&gt;, …</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>GroupSplitting</td>
<td>&lt;GroupName&gt;, &lt;GroupSplitter&gt;, &lt;UserName1&gt;, &lt;UserName2&gt;, …</td>
<td>Group-structural awareness</td>
</tr>
<tr>
<td>StudentDeconnecting</td>
<td>&lt;UserName&gt;</td>
<td>Informal awareness</td>
</tr>
<tr>
<td>TeacherConnecting</td>
<td>&lt;TeacherName&gt;</td>
<td>Informal awareness</td>
</tr>
<tr>
<td>TeacherDeconnecting</td>
<td>&lt;TeacherName&gt;</td>
<td>Informal awareness</td>
</tr>
<tr>
<td>TeacherHelpCalling</td>
<td>&lt;UserName&gt;, &lt;TeacherName&gt;</td>
<td>Social awareness</td>
</tr>
</tbody>
</table>

Table 2: List of Awareness Indicators

As seen in table 2, each probe contains some parameters and has a particular aim in order to complete a specific category of awareness which is not often addressed.
These elementary probes raise two problems:

- **Quantity**: there is a large amount of (sometimes useless) information coming from all the clients and some selection or filtering methods are thus needed. As a matter of fact, in such an environment, many users play simultaneously. All these indicators create a great number of signals from which we have to select the interesting ones. The parameters of a probe may help this configuration to describe the indicator. For example, the teacher may want to observe a specific activity or a particular student: the appropriate probe will thus be configured to collect the right information.

- **Quality**: these probes can handle information that is too basic and not very easy to understand or not very meaningful. That is why the user or the teacher must provide some specific treatments to transform the information into something understandable.

These statements led us to offer the possibility of developing more complex probes based on the combination or transformation of elementary ones.

### 3.2 Complex probes, personalization of the expected perception

For a teacher, the expectations concerning the perception through the system are somewhat difficult to express. The level of what needs to be perceived may vary, as is also the case in traditional teaching: a teacher may want to observe basic facts (e.g. who starts a new activity) or more abstract facts (e.g. who regularly cooperates before answering a quiz properly). The API presented provides the users with elementary probes. They are thus useful for observing basic facts. However, they may not be helpful enough when the level of abstraction needed is higher. For instance, being aware only of a student consulting a help file can be not very meaningful. But if the same observation occurs just after s/he has given a wrong answer and then followed that with a success in the same activity, the teacher may be reassured as to the usefulness of the help file related to the activity. The combination of these three indicators (simple probes) allows to create a complex probe and thus to provide a higher level explanation about the on-going activity.
As stated previously, 17 basic indicators (simple probes) have been extracted from the pedagogical dungeon (see table 2) and 3 operators have been proposed: AND, OR, THEN to combine one probe with another. All these indicators may be combined with each other to define new complex probes. Figure 4 shows an example of such a definition. The new probe is available in the educational platform, with the same properties as the basic probes. In particular, the new ones can be reused to create a more complex probe. Thanks to this mechanism, the set of probes naturally increases with the help of the users, guided by the needs of observation in the platform.

![Complex Probes creation interface.](image)

The possibility of having basic or complex probes in the platform helps solve the problem of lack of feedback for the teacher. The definition of the API allows one to define properly what is observable on the platform whereas the possibility of defining complex probes enables a personalisation of the expected teachers’ perception.

### 3.3 Observation architecture

Technically, we propose a multi-agent architecture to achieve such an observation task. In fact, this is justified by the fact that we need to collect relevant information from each student’s workstation and to forward it to the teacher’s station. Furthermore, the system offers the possibility of reacting by adapting the session (e.g. sending messages to particular students, modifying some collect parameters, providing help files or even dynamically adding a part of the dungeon to provide new activities, etc.).

This decentralized system, exhibiting a strong need for collaboration/interaction, led us to base our observation architecture on a Multi-Agent-System (MAS) as described in Carron (2006). In this approach, specific software agents are distributed on every machine: collecting Agent on students’ workstations (“student Agents”) or servers (“server Agent”), structuring and visualisation Agent on the teacher’s station (“teacher Agent”). The whole system (pedagogical dungeon and multi-agent observing system) is described in Figure 5. The “teacher agent” enables one to configure the collect of any “student Agent” by specifying the traces of relevant activities that have to be sent to her/him. All these traces must of course be interpreted and presented to the teacher (as seen previously in figure 3).
As a matter of fact, when the system is installed, the observation may be finely tuned. Adapted complex probes may be defined by combining different simple probes through an interface (as seen in figure 4). Each probe may be generic or filtered on one or several specific parameters (see figure 6). For example, it is possible to be notified only when a student gives a wrong answer to the exercise X of the activity Y after having consulted the help file F.

All pieces of information related to the system may also be observed: the topology of the system, the probes deployed on each agent or the status of the different agents (see figure 7).

We have described the correspondence between a learning session and the objects of the dungeon, and the way of setting appropriate indicators in the system. The next section is related to a more descriptive view of our system: the creation of the dungeon (associated with a given learning session) by the teacher and the enactment of a pedagogical activity in this game based environment.
4 Description of the pedagogical dungeon

4.1 Creation of a new pedagogical session by the teacher

The creation of a pedagogical session is not an easy task for the teacher. This activity can be seen as the creation of a scenario, usually written with IMS-LD described in Koper (2003) or more flexible languages like LDL proposed by Ferraris (2007). If the teacher wants to construct a pedagogical session in the dungeon, s/he interacts directly with a session builder. This tool allows the four creation steps of a scenario: the definition of activities, the description of the available resources for each activity, the definition of the validation procedure for each activity, the definition of the constraints on the activities (organisational and logical temporal links). According to these constraints, the map of a dungeon will be automatically generated and saved.

Definition of a new activity

The user starts with a pedagogical session containing an initial activity called “prologue”. The students start their quest in the room associated to this activity. Usually, there is only one document available in this room, explaining the purpose of the quest and the main rules of the game.

The teacher defines her/his session by entering as many activities as s/he wants. Activities may be added in a session. The teacher then needs to define the information contained in the four tabs corresponding to the scenario creation steps (see figure 8).

Figure 8: creation of new activities

Providing general information

The teacher first supplies the name of the activity. Of course, this is useful in order to link this activity with the others, but also for the system to provide adequate feedback on a specific activity, as seen in section 3.1. The teacher also reports here whether the activity is collaborative or not.

Selecting the resources

The teacher chooses the different resources that will be available for a specific activity. This could be either local files (present on the teacher’s computer) or links to on-line material. These files usually explain the topic of the activity. The teacher chooses the most appropriate form for these resources: simple texts, videos or even simulation applications as it is the case in Michelet (2007).

Setting up links between activities

Ordering activities can be a difficult task. Let us consider independently the two kinds of ordering mentioned above. A teacher can construct her/his session as a sequence of activities: 'the students will start with exercise 1; they will continue with exercise 2 and finish with exercise 3'. In that case, the teacher only wants to declare an organisational order among the three activities of the session (see figure 9).
Another way of defining links between activities is to consider the pre-requisites for the different activities: ‘the students will start with activity 1 (presenting and testing concept C1); if they succeed, they can begin activity 2 (that introduces a new concept C2); but if they fail, they need to do activity 3 that provides additional and more detailed information about concept C1 since this item has not been acquired.’ In that case, we can consider that the teacher is defining the order by considering pre-requisites, i.e. s/he uses a logical order. In figure 10, a pre-requisite for activity 3 is a failure in activity 1.

There is also the possibility of stating parallel activities.

The teacher can thus easily use both of these ordering possibilities in order to describe the most suitable succession of activities for the session. Although the interface is quite simple, the tool allows expression of rather complex situations. For instance, we can say that Act7 must occur after Act1, that the pre-requisites for it are “success in Act3” and “success for all the group in Act4”, and that Act7 can occur at the same time as Act6. From a collaborative point of view, we can see that there is an easy way to synchronise a group: all the keys of the group should be needed. In that case, the student must wait for all the members of her/his team. In fact, most of the time, s/he goes back into the previous rooms to help them.

Evaluating the activity

Obtaining a key related to an activity depends on the evaluation of the activity. For each activity, the teacher can choose how to evaluate it.

The simplest way to obtain a key is just to read a text. But, most of the time, the student must answer a question or a set of questions. Each of these questions can be a Multiple Choice Question or an open
question. In this last case, the teacher will be in charge of validating the answers to that question. We would also point out that questions can be collaborative, in which case the answers are given by the whole team. We have developed here a simple way to obtain the keys.

Once the teacher has defined all the activities of the session, the dungeon can be generated. Figure 11 is an example of the result of such a generation.

![Dungeon Map](image1.jpg)

Figure 11: a dungeon map

**Graphical awareness of the activity status**

A similar overall view from above (mini-map) is always supplied for the teachers during the game. This view is dynamic, since one can see all the users involved in this pedagogical session moving through the different rooms. This input provides the teachers with some awareness about the on-going activity. The student’s view is restricted: s/he can see only the rooms that are accessible to her/him (i.e. the rooms whose keys s/he possesses).

### 4.2 Enactment of the pedagogical activity: exploring the dungeon

Two roles may be played in the dungeon: teacher or student. Both sides will be described in this section.

Each user is identified with a login. The progression of the student is saved when s/he quits the application and will be restored when s/he reconnects to continue the learning session or the exploration of the dungeon. The system may also be accessible from home but, in this case, other students or the teacher may not be present in the dungeon at the same time.

#### 4.2.1 Student side

In our virtual environment, a student is represented by an avatar (see avatar choice figure 12) whose characteristics evolve dynamically over time, according to the activities completed during the session (keys acquired). For instance, the avatar can wear different clothes that the student wins during his/her session.
Most of the time, a student is present in a virtual room representing an activity. S/he can access several resources related to the activity.

In figure 13, two persons are present in a room; the avatar on the bottom (with the helmet) is the teacher. The other one (a student) has his nickname written above his avatar (Antony) and the name of the activity (prologue) is written on the floor of the room. Touching a sphere/globe item (a resource) opens a text window with explanations or provides a web link, a file, etc. Touching a crystal item proposes an exercise, a test or a quiz. A correct answer to a crystal question generally gives the student a key to open the door and lets her/him continue the quest (see figure 14).

The translucent white area is a chat window for collaborative features. Each person present in this room can see what is said (see figure 14). Clicking on a specific avatar may open some private chat windows.
The visualisation is updated in real time and the student may move inside the room and see other avatars move and progress in the dungeon. When a question is related to the concept presented in the room, some clues may be found in the resources displayed in the room. The answer can be automatically identified as correct (closed question or key words present). In this case, the result is instantly notified to the student by the system and a door is possibly opened (see Figure 15). But very often, the involvement of the teacher is necessary. S/he corrects the exercise dynamically, can add remarks, and validates the answer or not. Whatever the case, the student is notified via a window (see figure 15 on the right) containing all the stated information.

The role of the teacher is presented in the following section.

4.2.2 Teacher side

The teacher view is specific and s/he may have an overall view of the dungeon (see figure 11). One of the teacher’s main aims is to follow the progression of the students. The overall map view shows the position of each student and provides the possibility of teleporting her/himself instantly to a specific room to see for instance the teamwork in a collaborative activity. Nevertheless a mini-map is always available in the game view (see figure 16).

Another part of the role of the teacher is to answer the questions or to validate the answers proposed by the student. When a question is awaiting a teacher validation, a specific window appears in the teacher interface (see figure 16).
As we will see in the next paragraph, collaborative activities are also possible. The teacher may thus create groups and dispatch students into them, as is shown in figure 17.

![Figure 16: Teacher’s view: Validating an answer and the mini-map on the bottom right.](image)

Thanks to these notification windows or even visualisation windows, the teacher is aware of the current activity of each student. S/he may choose to help slow students or to propose other activities or resources to particularly fast students. Some rooms also enable collaborative work and thus the synchronisation of students with each other, as presented in the next section.

### 4.2.3 Collaboration in the dungeon

In our system, some activities must be realised collaboratively. The associated rooms require the students to answer in groups as indicated on the accessing door of the room. The crystal hiding a group activity has a specific colour and notifies the first student arriving (see figure 18). The advantage is to make a rapid student help another one before continuing the progression. The groups are thus homogeneous. Collaborative keys are given to all the avatars that constitute the group.

![Figure 17: Teacher’s view: Dispatching students into groups](image)
This serious game visualisation is dynamic because all the elements are updated in real time and because the teacher can interact with this view. Nevertheless, we will see in the next section concerning real experiments that interactions with students may be very time-consuming.

5 Experiment

The pedagogical dungeon has been experimented during several practical works with a real classroom at the University of Savoie, France.

5.1 Description of the learning activity and of the system

The pedagogical objective

The pedagogical session modelled concerned a lesson about operating systems. Several independent concepts were exhibited to control an operating system through console commands. The objective of the work was to practice such commands and to verify that the links with theoretical concepts were acquired. The students were 18 years old and familiar with computer use. All the types of artefacts or exercises described in section 2 were used in this experiment. The students were working on Linux, and had to find or test some text commands in the console.

The methodology of the experiment

During the experiments, about fifteen students and their teacher were present in the classroom. Each student worked on a station equipped with the student client software and the teacher used the teacher client software. The students were allowed to consult part of their lesson or web sites (files or URLs found during the exploration of the dungeon). The students were warned that all their actions would be observed through trails. The students were free to refuse this observation but everyone agreed to follow the proposed protocol. They were explicitly allowed to communicate through
the chat tool provided with the system. An avatar of the teacher available for students’ questions was always present in the dungeon.

All the observation probes were configured at the beginning of the session and were active during the whole session. Moreover, forms filled in by the students just after the experimentation gave us interesting feedback about the software itself. Some necessary improvements have already been highlighted and resolved: for example, many students tried every possibility to validate a multiple-choice questionnaire.

5.2 Conditions and objectives of the experiment

The very first experiment aimed to validate the overall approach and to test the system technically: as expected, students used the proposed environment without any explanation. However, the main goal of the experiment remains to establish whether such an environment improves the learning process. From the students’ point of view, the feedback was very positive: a similar “classical” (without the support of such an application) practical work had been proposed before to the same students. They were much more enthusiastic about the system version: the multiplayer aspect was a great factor of motivation and commitment for the students. An informal competition appeared between users exploring the dungeon. The chat tool was intensively used to communicate about the exercises, boosting the competition and reinforcing the immersion feeling.

Subsequent experiments focused on improvements realised on the system. Three evolutions have already been implemented and validated. We can notice that “cosmetic improvements” have already been realised on the version presented here because it seemed that the first world proposed was not very attractive to girls. Special adaptation must be envisaged to adapt to both male and female students like in Hubbard and al. (2007).

5.3 Pedagogical results

Concerning the teacher, we noted that the use of such a tool may be somewhat disturbing: the way of following the learning progression has entirely changed and s/he may sometimes be overloaded by the number of questions from students waiting. From another point of view, some advantages were exhibited about the level of understanding of the lesson. Concerning the activity regulation, direct explanations were given to students proposing wrong answers, and additional files were included during the game. Collaborative activities have not been evaluated sufficiently, since they were located at the end of the scenario, and some students did not reach that point.

The teacher also complained about the cognitive overload generated by the system: too much information was displayed concerning the different events occurring in the dungeon. Although it was possible to reduce the volume of the traces displayed, this requires some additional time from the teacher, who was already overloaded. This remark raises the question of an adaptable control board containing suitable indicators, enabled or not according to the context.

The teacher thus prefers to use the system with Multiple Choice Questions and with a minimal awareness tool. He prefers to enrich the probes as needed during the activity’s
progression. As a matter of fact, other experiments have been set up concerning a lesson about Object Oriented Design, but we failed to set up an experiment where students had to write pieces of software (too difficult to evaluate quickly enough).

Some students pointed out that the style used in the exercises did not match the style of the game. This means that teachers should think about the form of their pedagogical activities to make them fit with the dungeon’s look and feel (here the medieval style). A good immersion of the session subject in the dungeon ensures a better motivation on the part of the students, as shown in Rosenbloom (2004) or Cheng (2005).

Thus, the most interesting point is that the pedagogical session has to be well prepared and to be specifically adapted to such environments. As stated previously, for the teacher, the experiments revealed a great difficulty to read and validate at the same time: a lot of student propositions are generated from open questions. The whole session has thus to be well designed in order to protect the teacher from such an overload: s/he must also allow time to observe the pedagogical activities and to react dynamically. Moreover, in the first experiment, the teacher was disappointed because he couldn’t change the pedagogical scenario dynamically: due to misunderstanding of a concept, most of the students had difficulties in a particular room. It would have been useful to add a special room for additional explanations.

Improvements based on this last remark were therefore implemented between the first and second experiments: a new part of the dungeon may thus be created and added to an existing one. The latter is linked with the extension via a teleportation portal (see spiral item in figure 13).

6 Discussion and perspectives

The prospects at the end of this experiment are numerous.

Several instances of the dungeon have currently been tested to date, but an interesting perspective is to propose the various scenarios to the students. Many teachers from other domains (electronics, mathematics) are very interested in testing it in their lessons. From our point of view, this perspective will show that the tool is not restricted to a specific domain and could be integrated as a tool into the “old” LMS we developed in Martel and al. (2004), which is currently used by 25 000 persons in our university.

Today, there are too few results to evaluate the impact of such a system on understanding, but the students have asked whether other lessons could be proposed in this way, and are motivated by the possibility of continuing at home. To them, the idea to finishing the exploration or of replaying a dungeon game seems to be a natural manner in which to continue and complete the learning process at home.

Many environments, such as Mazza and Dimitrova (2005), propose statistical tools and sometimes also specific functionalities for tracking students (see Moodle (2007) for a popular CMS proposing tracking) or for tutoring student groups tutoring, but in most LMS, this analysis can only be realised a posteriori and is static. The different ideas proposed in this paper give good reason for being particularly interested in real time visualisation: the knowledge of the current user activities is considered more helpful than only past activities. We need it in order to be as aware and as skilled as possible to take the best decision, and to regulate the collaborative activities.

Another possible use of the dungeon deals with the ‘a posteriori’ analysis of a session. Saving, storing or retrieving past activities may give sense to a learning session or a specific
serious game. Each signal or trace may be saved in a database in order to be consulted again at a later date.

Several advantages of this ‘a posteriori’ analysis may be shown. For example, a teacher does not always have the time during the session to analyse what is going on and thus, it could be interesting to replay some parts or even the complete session. In the educational domain, it is well known that the reflexive aspects of the trace may be interesting: a student may want to consult her/his own trace for meta cognition purposes or even to compare it with another student’s.

The teacher needs to be aware of the on-going activity and s/he needs to follow the progression of all the students. Indeed, the overall view of the dungeon presented in 4.2.2 is a good tool to perceive the on-going activity and an ‘a posteriori’ analysis of a session gives information about the progression of all the students. For instance, the traces can be used to draw the path of each student in the dungeon or the room can be highlighted with colours to indicate the state of a student with respect to an activity. It may be noted that this last type of information, regarding what has happened in the previous few minutes, can be revealed to the teacher in real time.

The results and students’ returns show that it is not only or just a transformation of a pedagogical scenario into a dungeon structure. Thanks to these digital environments, we are able to observe a learning session in a much more accurate manner. Therefore, we are able to adapt or regulate the pedagogical scenario in real time. This new dimension offers the possibility of providing additional help or exercises, or various paths to acquire knowledge, adapted to each sort of student. Today, it is difficult to give different exercises directly, according to the skill of each student, or to just present an alternative between several exercises.

7 Conclusion

In this paper, we have demonstrated through examples how educational games are relevant to providing students with a dynamic and pleasant learning platform. Although the user interface was quite important for usability purposes, we wanted to present a new way of designing learning activities. One of our next goals is to check the links between the creation of a pedagogical scenario with standard languages, such as IMS-LD, and the generation of the dungeon.

Collaboration between different roles (within a group of students, or between student and teacher) has been considered. We think that this aspect can be improved: in this version of the game, only two roles are implemented and are static. This notion of role is interesting and we would like to extend it to many dynamic roles, e.g. a student can become a leader for a strictly defined period of time in a collaborative task. We can also improve the collaborative space itself, by offering more than a chat in a room. Asynchronous tools must also be considered when we want to tackle distance learning (students are not always present at the same time in the dungeon).

Another part of this article deals with the awareness indicators providing feedback to the teacher on the ongoing activity. A multi-agent system is proposed to create, configure and deploy specific probes to observe the activity in the dungeon. We have shown that such an approach allows ones to address specific categories of awareness that are generally not tackled.

Finally, visualisation of the probes must be improved as well as the operators to elaborate new complex probes. For example, we imagine proposing specific visualisation with calculus functions. We have also planned to work on collaborative indicators in order to suggest adapted actions to enhance the overall group activities. These indicators must be defined through personalized tools for the teacher’s requirements.
Acknowledgements

We greatly appreciate the commitment of the many people who contributed to the development of the software presented here: M. Bodin, G Dalla Costa, J. Demolis, Y. Henri, M. Michard, J. Pavlou, J. Roche, V. Saisset, JC. Serra, D. Viala.

References


