Observation of collaborative activities in a Game-Based Learning Platform

<table>
<thead>
<tr>
<th>Journal:</th>
<th>Transactions on Learning Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID:</td>
<td>Draft</td>
</tr>
<tr>
<td>Manuscript Type:</td>
<td>Special issue on Intelligent support systems for CSCL</td>
</tr>
<tr>
<td>Keywords:</td>
<td>K.3.1.a Collaborative learning &lt; K.3.1 Computer Uses in Education &lt; K.3 Computers and Education &lt; K Computing Milieux, J.8.g Games &lt; J.8 Internet Applications &lt; J Computer Applications, Observation facilities, Traces, Indicators, Regulation of activity, Learning games</td>
</tr>
</tbody>
</table>
Observation of collaborative activities in a Game-Based Learning Platform

Jean-Charles MARTY, and Thibault CARRON, SysCom Lab, University of Savoie, France

Abstract— The work reported here takes place in the educational domain. Learning with Computer-Based Learning Environments changes habits, especially for teachers. In this paper, we wish to demonstrate through examples how learning sessions set up in a Game-Based Learning environment may be regulated thanks to observation facilities. Providing teachers with feedback (via observation) on the ongoing activity is thus central to being aware of what is happening in the classroom, in order to react in an appropriate way and to adapt a given pedagogical scenario. The first part deals with the description of our view of learning games illustrated through the "pedagogical dungeon": a game-based environment that we have developed. The second part of the paper focuses on the description of different ways and means to obtain information about the learning activities. It is based on traces left by users in their collaborative activities. The information existing in these traces is rich but the quantity of traces is huge and very often incomplete. Furthermore, the information is not always at the right level of abstraction. That is why we explain the observation process, the advantages of a multi-source approach and the need for visualisation linked to the traces. In the third part, we illustrate these concepts in the pedagogical dungeon equipped for observation and with the capacity for collaboration in certain activities. We pay particular attention to the regulation process of the collaborative activity by providing case studies explaining how the observation process enables the regulation actions. Finally, the feedback about the experiments presented is discussed at the end of the paper.

Index Terms—Learning Games, Trace, Indicator, Observation facilities, Regulation and Collaborative Learning Tools.

1 INTRODUCTION

RECENT years have seen a rise in learning with Computer-Based Learning Environments. These environments offer functionalities that are recognized as being valuable, but students tend to consider them as unexciting. Agreeing with Vygotski’s school of thought and activity theory [42], we consider that the social dimension is crucial for the cognitive processes involved in the learning activity. Consequently, the question is 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 [36] and even in education [35], [37], it was decided to develop one 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.

This new way of learning changes habits, especially for teachers. Most of the time, a teacher prepares his/her learning session by organizing the different activities in order to reach a particular educational goal. This organization can be rather simple or complex according to the nature of the goal. For instance, the teacher may decide to split the class into groups, to ask the students to search for an exercise in parallel, to put different solutions on the blackboard, to have a negotiation debate about the proposed solutions, and to ask the students to write the chosen solution in their exercise-books. The organization of the different sub-activities in an educational session is called the “learning scenario” (see [20] for a definition of a pedagogical scenario).

In traditional teaching, namely in an environment with no computers, a teacher tries to be as aware as possible of his/her students’ performance, searches for indicators that allow him/her to know a student’s comprehension status and which activity of the learning scenario this student is performing. The teacher then adapts his/her scenario, e.g. by adding further introductory explanations or by keeping an exercise for another session. Once the training session is finished, the teacher often reconsider his/her learning scenario and annotates it with remarks in order to remember some particular points for the next time. For instance, s/he may remark that the order of the sub-activities has to be changed or that splitting into groups was not a good idea. In this case, the teacher is...

J.C. Marty is with the SysCom Laboratory, Campus Scientifique, 73376 Le Bourget du Lac Cedex, France. E-mail: Jean-Charles.Marty@univ-savoie.fr
T. Carron is with the SysCom Laboratory, Campus Scientifique, 73376 Le Bourget du Lac Cedex, France. E-mail: Thibault.Carron@univ-savoie.fr

http://mc.manuscriptcentral.com/tlt-cs
continuously improving his/her learning scenario, thus following a quality approach [32]. In educational platforms, and in particular in game-based learning environments, the teacher would like to have the same possibility as in traditional teaching, to be aware of what is going on in the classroom, in order to react in an appropriate way. Of course, s/he cannot have the same feedback from the students, since s/he lacks human contacts. However, in such environments, participants leave traces that can be used to collect clues, providing the teacher with awareness of the ongoing activity. These traces reflect in-depth details of the activity and can reveal very accurate hints for the teacher.

This observation feature in learning environments enables the provision of tools to the teacher allowing him/her to react to a particular situation, for instance: a student is in trouble; there are too many interactions among a group of people; there is not enough communication in a collaborative task. Being aware of these particular situations helps the teacher to adapt his/her following actions, that is to say the learning session. For instance, s/he can communicate with a student and help him/her or s/he can deactivate the communication tools within the group of participants. This adaptation of actions in a collaborative activity is also called “regulation of the activity”.

In this paper, we wish to point out the different aspects which enable the regulation of collaborative activities in a Game-Based Learning environment. We first describe our view of learning games illustrated through the “pedagogical dungeon”. We thus present the problems linked to observation through traces. Although this approach is very powerful, we will see that observation is a tricky task, with a lot of problems to be solved in order to obtain relevant observations allowing decisions to be made for an improvement of the collaborative process. Then, we classify the different kinds of possible actions to regulate the activity. We also introduce indicators, deduced from what has been observed, reflecting particular contexts. Finally, we give examples of regulation of the pedagogical activity in the “pedagogical dungeon”.

2 Description of a Game-Based Learning Environment: The Pedagogical Dungeon

In this part, we describe our game-based learning environment. We first address the links between an educational activity and this game. Then, we demonstrate how the different participants interact with this game. Although the concepts presented in this part are not key research issues, we prefer to explain them to the readers in order to facilitate the comprehension of our approach presented thereafter and linked to the observation, awareness, and regulation in the collaborative activity.

2.1 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 [10] for a definition of Activity Theory) resulting from Vygotski’s proposals[42]. 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 by Dunne in [10]).

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 & Henderson [16], we propose in this section to link the objects used in our game-based learning environment 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>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>Actors</td>
<td>Avatars (teachers, students)</td>
</tr>
</tbody>
</table>

Table 1: Correspondence between AT Concepts and Game-Based LMS Representation

2.2 Breakdown of a learning session: rooms and topology

The learning session (or learning activity) is very often divided into different activities. This is the case when the teacher proposes to her/his students a set of interlinked exercises in order to reach a pedagogical goal. Each activity has its own local goal, generally a concept to acquire. For a student, successfully carrying out 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 the 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.
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 by Koper in [22] or more flexible approaches [13], [18]. 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 the type of activities; for instance, an activity can be collaborative.
- The description of the available resources for each activity. Resources could be either local files (content included within the scenario’s definition) or links to online material. These files usually explain the topic of the activity. The teacher chooses the most appropriate form for these resources: a simple text, videos or even simulation applications, as is the case in [29].
- The definition of the validation procedure for each 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 Multiple Choice or open. In the latter case, the teacher will be in charge of validating the answers to that question. We would also point out that questions may be collaborative, in which case the whole team gives the answers.
- The definition of the constraints on the activities (organisational and logical temporal links). According to these constraints, the map of a dungeon is automatically generated and saved. (see [2] for details). Figure 2 is an example of the result of such a generation.

### 2.3 Enactment of a learning session with students

Similarly to in a Role-playing Game, 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: students can access knowledge through resources (documents found inside the game), via help from teachers, or from work with other students. For pedagogical purposes, the dungeon can be flexible. For instance, “teleportation portals” can lead to new rooms created dynamically during the session in order to resolve a problem.

Each room is dedicated to an activity. One 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 completes a test, an exercise or a quiz successfully. The assessments are thus also located in the room.

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 one, it only means that these students are performing the same activity. They can of course access the resources at the same time.

Each room can be accessed through doors. These doors are the guardians 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.

Most of the time, activities are well ordered in the dungeon: 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 from previous activities before entering a new room.

Nevertheless, activities need not necessarily be ordered in the dungeon: students may thus choose between several exercises if the teacher wishes to offer this flexibility.

Application to the dungeon: Use case 2 - Embarking on the quest: moving, answering questions, obtaining keys.

In our virtual environment, we represent a student by an avatar (see avatar choice in figure 3) whose characteristics can evolve dynamically over time.

---

**Figure 1:** An example of a scenario seen as a dungeon topology

**Figure 2:** A dungeon map
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 4, two students and the teacher (with the helmet) are present in a 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 4).

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, 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 5). 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 5 on the right) containing all the stated information.

An overall view from above (mini-map) is always supplied for the teachers during the game (fig 2). 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 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).

2.4 Collaboration in the dungeon

The teacher may want several activities to be collaborative. In that case, the rooms associated to these activities are collaborative places. These places require the students to answer in groups as indicated on the access door to the room. The advantage is to facilitate the exchange of data between students, making them ask questions and (or to) co-resolve a problem. As in traditional teaching, these collaborative places offer the opportunity to create small groups in order to take advantage of the abilities of the various members of the group, who can explain their resolution strategy to each other.

Currently, a chat facility is provided in the dungeon rooms (see the translucent window in figure 4), but we can of course imagine other collaborative tools available in the collaborative places (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. The teacher may thus create or modify groups and dispatch students into them.

In collaborative rooms, the quiz is also collaborative: The crystal hiding a group activity has a specific colour and notifies the first student to arrive (see figure 6). Students in the same team must all be present in the room. They may exchange via the chat tool before answering the question. In the event of a correct answer being given for a collaborative quiz, a collaborative key is provided to all the members of the team.

As in “traditional classrooms”, a student may also collaborate with a teacher, for instance if s/he needs help from her/him. The teacher is also present and contactable in the game, represented with a specific (and impressive) avatar.
This Game-Based Platform has been used in real situations and from this experience we can easily conclude that it is attractive for the users. But, 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. Loss of perception for the teacher in these environments can make the tool unusable for him/her, because s/he can no longer regulate the collaborative activity.

### 3 OBSERVING A LEARNING ENVIRONMENT

The tracing activity is an appropriate way of reflecting in-depth details of the activity and of revealing very accurate hints for the teacher.

Unfortunately, traces are objects which are very difficult to manage and understand. We propose first to demonstrate the kind of problems linked to observation and to explain them through a pragmatic approach (experiments).

#### 3.1 Pragmatic Approach to Observation Problems

**Criterion 1: Rich information available in Log Files but hard to extract and to exploit.**

A first aspect to consider, central to the observation area, is the form of the traces. Many e-learning Platforms or Learning Management Systems are based on Web Servers [43], [1]. These servers easily supply logs (information concerning the connections on this server) stored in specialised files. We first used this information in an experiment carried out at our university. As we needed to analyse the new usages induced by the use of our local e-learning platform (“the electronic schoolbag”), we decided to work from the traces left by thousands of users. The source of these traces was a web server providing data in the SQUID format, as for instance, 193.48.120.76 22/04/2003 04:25:31 POST TCP_MISS/200 http://www.univ-savoie.fr:443/Portail/logged_in


It is obvious that these traces are not directly interpretable. They should be transformed and rewritten, in order to make it possible to understand them. For instance, 193.48.120.?? => “Connection to the e-learning platform from the university”.

Here, we want to identify connections matching the 193.48.120.?? address, meaning an access from the university site, where ?? can be replaced by any number from 1 to 255.

The traces were analysed a posteriori by a researcher in the “information and communication” field. From this experiment, new practices were revealed such as the use of the platform at home, but without using collaborative tools [5]. The experiment also pointed out the need to address the problem of treating the huge amount of data available in the log files.

**Criterion 2: Necessity for trace transformation for information extraction.**

In order to better manage this huge and fine-grained information, we specified a transformation chain allowing the manipulation of traces (figure 7). The main purpose is to reach a good level of granularity, thus enabling an enhanced comprehension of the user behaviour [24].

This chain proposes several functionalities to manipulate the traces: filtering in order to reduce the huge quantity of logs, aggregation in order to change the level of granularity (abstraction) of the traces, transformation into a uniform format in order to take into account several log formats (SQUID, APACHE, I2S), or storage in a database and use of a Data Base Management System through SQL requests.

An experiment enacting this transformation chain allowed us to make an “observatory” tool, dedicated to non-computer scientist users. This tool enables the gathering of statistics on the usage of the “electronic schoolbag”, such as the number of connections, the types of users connected, the kind of preferred tools. Visualization functionalities were added to this tool for obvious reasons of classical representations of statistical data (graph representations, figure 3), thus adding a visu-
alisation step to the transformation chain.

**Criterion 3: A multi-source approach greatly enhances the understanding of the activity.**

A certain number of research works linked to pedagogical platforms concern the formalisation of educational scenarios [22]. The teacher frequently foresees a sequence of activities to be performed during the learning session. This sequence, also called a scenario, guides the session, and comparing the learners’ activities and the predefined scenario becomes crucial [14]. This comparison allows the teacher to be provided with awareness of the ongoing activities, and enables the improvement of the scenario itself.

This is not an easy task, since the users can use simultaneously tools that are not integrated in the educational platform (forums, web sites, chat). We do not want to restrict our understanding to the tasks included in the predicted scenario. We want to widen the scope of observation, so that other activities performed by a student are effectively traced. Even if these activities are outside the scope of the predicted scenario, they may have helped him/her to complete the exercise or lesson. We thus need to collect traces from different sources. It is therefore interesting, from a general point of view, to be able to take into account more than one source of data. Such an approach allows the deduction, from the multi-source traces, of non-foreseen behaviour. For instance, during a session in the pedagogical dungeon, a student may leave the game and start to communicate with other students to have the correct answer to a quiz.

Through an experiment described in [27], we have observed non-foreseen student behaviour. It is then possible to choose from the collected logs from different sources to specify, annotate or better explain what has happened during the session (see Fig 8), through a “trace composer”.

To help the user to better understand the trace generated, a graphical representation is a good support to make links between the learning scenario and the traces. We also take the different sources into account, in order to refine the understanding of the effective activity. We propose a metric to see how much of the activity performed by students is understood by the teacher, which is graphically represented on a "shadow bar".

The comprehension of a general activity implies situating non-foreseen behaviour with foreseen activity sequences, as shown in figure 4 with exercise 1, document 1 read. It is thus useful to be able to reposition the users’ actions in the pedagogical scenario. In this experiment, we suggested a scenario improvement, since we pointed out that all the students who finished the learning session communicated with the teacher at the end of the first exercise in order to validate it. This validation making them more confident can thus be proposed in the scenario itself.

The comprehension of a general activity implies situating non-foreseen behaviour with foreseen activity sequences, as shown in figure 4 with exercise 1, document 1 read. It is thus useful to be able to reposition the users’ actions in the pedagogical scenario. In this experiment, we suggested a scenario improvement, since we pointed out that all the students who finished the learning session communicated with the teacher at the end of the first exercise in order to validate it. This validation making them more confident can thus be proposed in the scenario itself.

**Figure 8: Tool visualising traces from different sources**

This study thus allows us to address problems that are linked to the scenario and the necessity of following the activity. Analysing the traces after the session provides the analyst with interesting results but does not solve the problem of giving the teacher the necessary awareness to react immediately to particular situations.

**Criterion 4: Immediate analysis enables reaction. Visualisation improves the teacher’s awareness.**

Detecting potential problems as soon as possible is a crucial issue. In order to alert the teacher to the fact that the collaborative activity is not progressing as expected, we need to compare the traces representing the actual activities with the ones mentioned in the predefined scenario and try to establish links between them. It is essential for the teacher to have a view of what is going on, in order to be able to react to given situations. The a posteriori analysis remains valid but can be expanded by analysis during the activity. New observation goals can also appear during the session. For instance, it can be useful to observe the status of the students during the first part of the session and to synchronise them before starting the second part of the session, being sure that everyone has acquired the requisite concepts.

This adaptive observation, needing high flexibility from the system, can be implemented through agents. A set of "pedagogical observation agents", set up on the students’ computers, inspects certain users’ actions (the ones that are on focus for the observer) and notifies an awareness agent before invoking a visualisation agent to provide the teacher with the appropriate information. This distributed system is thus able to collect the significant logs directly on the machines through specialised agents [2].

The visualisation agent interprets the traces sent by the observer agents in order to display them on a dashboard for the teacher. An example of such an agent, called “classroomviz” has been developed (figure 9). Indicators are computed from activity traces and from a predictive scenario, offering the average performance time for each activity [15]. The teacher can thus easily follow the students that are behind with certain activities (red faces).
3.2 Architecture of an Observation Software

From the criteria pointed out in the previous section, we propose an architecture suited for taking these points into account.

Suggested analysis viewpoints reinforce the established phases linked to the observation lifecycle. These phases can be described as follows:

1. A collecting phase, where relevant traces are identified and collected before being treated by a dedicated agent (structuring or visualisation);
2. A transformation phase (structuring, abstraction) of collected data in order to make the rough traces more explicit and to make these traces understandable for the observer (researcher, teacher, or student);
3. And, a visualisation phase, where visualisation techniques will be used in order to reveal the semantics from the traces, make it easier to understand and help an analysis from a particular viewpoint. The phase is aiming at facilitating the interpretation of the on-going activity by a non-specialist.

3.2.1 Model of a Trace-Based System

We base our work on a model elaborated in collaboration with the SILEX Team of the LIRIS laboratory. This model, called Trace-Based System (TBS) defines the different modules associated with the different phases mentioned previously.

Figure 10 illustrates the process allowing the observer to interact with a traced e-learning platform in order to visualise and regulate the activity using the traces. The observer plays the role of a “trace composer”. S/he furnishes both the pedagogical scenario possibly expressed with IMS-LD [22], and the description of the experiment pointing out the analysis needs [Carron et al., 2006]. S/he thus sets up the e-learning platform by adjusting collection and transformation tools. Then, the experiment can be enacted, providing the analysts with usage feedback.

Collection phase

As demonstrated in figure 10, the collection phase is prepared before using the TBS and consists in gathering the traces generated in the e-learning platform. Trace collection is a complex computer science problem, due to the large volume of rough traces that one can possibly collect. This collection can be made through instrumented software according to the trace composer’s intentions [40] or through files generated by the operating system, or through dedicated spy software, such as key loggers.

Another problem related to trace collection is the heterogeneity of rough traces (particularly in a multi-source approach), which necessitates studying a way to model them [19].

Transformation phase

The transformation phase is performed inside the TBS. The trace being an object in itself, the notion of Trace-Based System has emerged over the last few years, in order to allow and facilitate the exploitation and the interpretation of traces [38]. The functionalities of such systems therefore concern the manipulations of the traces. From the rough traces, a TBS offers a set of operations among these objects: filtering, joining or abstracting them. When the results of these operations are still traces, they remain inside the TBS and they can possibly be used for other manipulations. A TBS also offers services allowing trace organisation, such as storage or historical mechanisms. Research questions related to this phase overlap with those of trace cleaning [7], trace aggregation according to temporal [25], semantic or syntactic constraints [41], trace rewriting or modelling [38], [6].

Trace visualisation

The visualisation phase consists in making requests among traces and in visualising traces. These visualisation tools are part of the interface between the TBS and a user observing the system. We decide to situate the visualisation and the request system outside of the TBS, since these tools do not fit the definition of trace manipulation as given by Settouti in [38]. Indeed, visualisation techniques produce results that are not traces. Visualisation consists in elaborating a graphical representation, ad-
apted to the analyst’s objective, from traces contained in the TBS. This representation can take many forms, such as a temporal 2D visualisation of a trace [15], of several traces [28], or a spatial 3D visualisation [8]. The visualisation system relies strongly on the analyst’s objective. For instance, the visualisation system must be able to provide the analyst with a real-time visualisation of the enactment of the users’ activities, and particularly to detect and show the users in difficulty. The system must also provide him/her with information about activities causing problems for these users. Finally, a visualisation of individualised paths showing the path of activities for each user must allow the analyst to make an intermediate assessment of the users’ progression (see label (G) in figure 9).

This model led us to set up an architecture dedicated to the observation problem. We also took into consideration that a centralised approach could not offer adequate functionalities for diverse observations.

### 3.2.3 Distributed approach: an agent-oriented architecture.

#### Reasons for multi-agent architecture

The observation of collaborative activities has several salient characteristics that provide good reasons for an agent approach.

1. **First**, the problem is geographically and functionally distributed. Indeed, each student works on his/her own workstation and some information must be collected locally before being sent to other stations (for instance, the teacher’s) in order to be treated or displayed.

2. **Furthermore**, it is not possible to foresee which machine will receive or send the information. This depends mainly on the observation goal and on the students’ actions. This is thus highly context-dependent. There is no a priori solution to this problem because one cannot know the students’ behaviour in advance.

3. **Each machine** must remain autonomous in order to keep the progress of the pedagogical activity unchanged. It must also be able to communicate with each of the other machines, either to ask for information or to furnish some itself if necessary.

4. **Finally**, the set of collected traces possibly comes from different software and can be quite heterogeneous. It is thus difficult from a practical point of view to transfer the whole set of data coming from all the workstations to a unique station dedicated to the treatment of this data.

All these points justify the multi-agent approach. It would however be possible to add other advantages of such an approach, such as for instance the necessity for an observation system to be open or fault tolerant. The enactment of this kind of system must take into account the deployment context and the constraints imposed by the experiment in particular classrooms.

Multi-Agent Systems offer solutions for distributed systems in which autonomous software entities, the agents, can cooperate by means of interactions between them or with the environment. The choice of a multi-agent approach is thus particularly well adapted for such observation software. The general idea is to enact observer agents, autonomous software installed on each station, and that are in charge of collecting the relevant (according to a particular goal) actions performed on the station. This provides the teacher with a powerful means for being aware of the status of each student and thus being able to react in an appropriate way.

### Multi-agent system (MAS) enactment

In order to set up the experiments described above, we have developed and used such a system. As we have already highlighted, the observation goal of the pedagogical experiment is central for the technical choices. The architecture enacted is represented in figure 11.

![Fig 11. MAS Architecture for observation](image)

It contains 3 types of agents that may be installed on the machines: the collector agents (C), the structuring ones (S), and the visualization ones (V). From a technical point of view, some agents (not represented in the figure) are only dedicated to the system functionalities. This is the case for instance for the facilitator agent (directory service: white and yellow pages), or the deployment agent (launching and killing agents).

Generally, the MAS are based on multi-agent platforms [33]. Our objective is however to keep our solution as simple as possible, and to be able to deploy it with a minimum of constraints. That is why we have chosen JAVA agents, that are platform independent and that can be launched easily on each station by a simple click. From a conceptual point of view, this solution is open and allows us to develop new agents when needed, without changing what is already working. Agents with specific functionalities (useful in particular situations) can thus be enabled or disabled when needed.

From a technical point of view, these observation features must work on any pedagogical platform. The software
environment becomes a trace generator. The agents are
developed in such a way that they can be considered as a
probe on any trace source. The main constraint is of
course that the educational platform should provide
traces and their interpretation model. Naturally, this
“equipment” phase involves having access to the soft-
ware of this platform, in order to have precise and rich
traces.

4 APPLYING THE OBSERVATION FEATURES TO THE
PEDAGOGICAL DUNGEON

In this part, we describe how all these observation fea-
tures are included in a learning environment based on
game concepts: the pedagogical dungeon. The three iden-
tified phases of observation will be taken into account.

4.1 Collecting phase

We have chosen an approach which is as generic as possi-
ble 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 observa-
tions. For instance, in the dungeon, actions such as “enter-
ing 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 contain some parameters and may be flagged at any
moment by any client of our application.

1. WorkshopArriving <UserName>, <WorkshopName>
2. WorkshopLeaving <UserName>, <WorkshopName>
3. WorkshopAnswering <UserName>, <WorkshopName>,
   <AnswerContent>
4. WorkshopAnsweringWithContent <UserName>, <WorkshopName>,
   <AnswerContent>
5. WorkshopTeacherValidating <UserName>, <WorkshopName>,
   <TeacherName>
6. WorkshopTeacherValidatingWithContent <UserName>, <WorkshopName>,
   <TeacherName>, <Comment>
7. WorkshopCorrectlyAnswering <UserName>, <WorkshopName>,
   <Boolean>
8. StudentConnecting <UserName> Informal awareness
9. HelpConsulting <UserName>, <HelpName>
10. Chatting <UserName>, <ChannelName>
11. ChatContentListening <UserName>, <ChannelName>, <SentMes-
    sage>
12. GroupCreating <GroupName>, <GroupType>, <User1Name>,
    <User2Name>, …
13. GroupSplitting <GroupName>, <GroupSplitter>, <User1Name>,
    <User2Name>, …
14. StudentDeconnecting <UserName>
15. TeacherConnecting <TeacherName>
16. TeacherDeconnecting <TeacherName>
17. TeacherHelpCalling <UserName>, <TeacherName>

The list of events occurring during the experiment is
visualized through a Probe Visualisation tool (figure 12).

This basic tool provides the teacher with coloured text (a
colour can be associated with each kind of event).

These elementary probes raise two problems:

- Quantity: there is a large amount of (sometimes use-
  less) information coming from all the clients
and some selection or filtering methods are thus
needed. As a matter of fact, in such an environ-
ment, 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 configura-
tion 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 informa-
tion.

- Quality: these probes can only handle informa-
tion that is too basic and not very easy to under-
stand 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 develop-
ment more complex probes based on the combination or
transformation of elementary ones.

4.2 Transformation phase

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 ones (e.g. who regularly cooperates before an-
swering 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 ob-
servation occurs just after s/he has given a wrong answer
and then followed that with success in the same activity,
the teacher may be reassured as to the usefulness of the

Figure 12: Probe Visualisation (coloured text mode).
help file related to the activity. The combination of these
three indicators (simple probes) allows one to create a
complex probe and thus to provide a higher-level expla-
ation about the on-going activity.
As stated previously, 17 basic indicators (simple probes)
have been extracted from the pedagogical dungeon and 3
operators have been proposed: AND, OR, THEN to com-
bine one probe with another. All these indicators may be
combined with each other to define new complex probes.
Figure 13 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.

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 personal-
isation of the teachers’ expected perception.

4.3 Visualisation phase
We set up experiments with this probe visualization and
reached the conclusion that it was very difficult for the
teacher to remain immersed in the dungeon (answering
students’ questions) while looking at the probe visualisa-
tion. We were obliged to set up a new organization during
the experiment session with two teachers, one playing the
game and the other one looking at the probes and making
regulation decisions.
We thus decided to design indicators calculated from the
basic probes. These indicators help the actors to be in-
formed of the on-going activity while remaining im-
mersed in the game. Basically, we had two categories of
indicators: 1) knowledge indicators that show what
knowledge a particular student has already acquired.
This is visualized through pieces of armour gained when
answering the quizzes correctly. 2) behavioural indicators
represented by auras directly on the avatar. For instance,
in figure 14, the red spiral symbol was chosen by the
teacher to display a loneliness indicator and white snow
for empathetic behaviour.

5  EXAMPLES OF REGULATION IN THE
“PEDAGOGICAL DUNGEON”

5.1 Reacting in particular contexts
The regulation consists in linking observations about a
particular situation with a regulation action or set of regu-
lation actions, thus defining regulation rules, that consti-
tute our dedicated artefacts for regulating a learning ses-
sion [42].

As stated previously, it is possible from different means to
obtain a great number of traces that are most of the time
heterogeneous. Although it is necessary to equip the sta-
tions with as many observation functionalities as possible
in order to increase the observation possibilities, only
certain specific information is really interesting for the
different actors of the learning process at any one time.
Therefore it is crucial to raise the abstraction level accord-
ing to the user concerned in order to provide synthetic adapted information.
For that purpose, we have defined indicators based on
observed traces to present a specific view on the on-going
activity. Indicators can be considered as signals enabled
when a particular, interesting situation happens. For ex-
ample, the teacher can use an indicator to know which
students are behind with an activity. The indicator is here
the result of a calculus from the trace ‘entering an activity’
and the expected duration of this activity. In this particu-
lar situation, the teacher may want to accomplish a spe-
cific action and thus to regulate the activity, such as for
example, making a new help file available for these stu-
dents with an associated notification.

A difficult part of the regulation specification or more
precisely of the definition of regulation artefacts is related
to the description of the situation when a regulation ac-
tion is to be considered. These situations are quite hard to
describe since they are often complex ones, where a sin-
gle indicator is not enough. That is why several indicators
are activated and a set of several complementary ones
allows us to define a context representing a more accurate
view of the situation (see fig.15). In the previous example,
we are able to extend and complement the information,
should it be the case that almost the whole set of students
is behind with this activity. In that event, the regulation
action must be changed: the difficulty of this activity, the
intelligibility of the statement or even the quality of the
resources provided must be reconsidered. The final regulation process will not be the same: there are now several other possible regulation actions adapted for such a situation.

Case study 2:
The teacher wants to know which exercises his/her students are failing. Most of them have difficulty in solving certain problems. In that case, the teacher can regulate the activity by adding new resources to help these students, by modifying existing ones or by opening a dialogue session with these students to provide them with hints to solve the problem.

This case study deals with the modification of the content of a specific activity during the session. The observation context depends on the time spent doing the same activity. It is made up of an indicator which allows us to observe the duration of all the activities. It especially shows during the session that more than 50% of the students are behind with one activity. The teacher is thus warned of this situation and decides to modify the help file.

The regulation action is composed of two actions: modifying the help file and warning the students that a new help file is available. Furthermore, it is possible for the teacher to interact directly with a user via the interface by clicking on his/her avatar. A private chat session is thus initiated.

Case study 3:
The students are chatting a lot through the collaborative tools, but the results are poor, according to the teacher. S/he needs to regulate the activity by disabling the chat tool and letting the students continue the other activities individually.

This case study concerns the possibility of acting on a tool via its interface. The observation context deals with the quantity of messages exchanged in the same activity. An indicator “chatting too much” is helpful here to describe the context mentioned. The regulation action is to disable the chat facility access for the students concerned by the indicator and thus to act on tools available in the learning environment.

These 3 case studies illustrate different levels of regulation that can be exhibited in a learning environment. The different experiments that we have carried out in our university in real situations have shown that such an environment is well perceived by the students. Involved in an immersive pedagogical session, they are not exactly aware of the regulation process. Most of the time, they are amused by the appearance of new pedagogical resources. The disabling actions on tools are very efficient but more disturbing if no satisfying explanation is given. We have not focused on this fact here, but each regulation action should be used with a notification message. Regarding the teacher, the cognitive overload prevents him/her from being able to develop from scratch and add a new activity during the session. Currently, the content of such high-level regulation actions must be prepared before the start of the learning session.
6 Conclusion and perspectives

In this article, we have demonstrated through examples how learning sessions set up in a Game-Based Learning environment may be regulated thanks to observation facilities. We have explained how to obtain relevant information about a specific expected situation and how to react through different levels of regulation actions directly during a pedagogical session.

We have also introduced indicators, deduced from what has been observed, reflecting particular contexts of observation. For future work, this implies extended research on the definition and classification of collaborative indicators and on a classification of the possible collaborative actions.

More precisely, we have described many actions on modelled elements of a pedagogical session that are relevant for regulation of the activity. We think that these actions or sets of actions should be seen as a means to resolve particular disturbing situations and could also be classified along with this point of view.

These concepts have been illustrated through a Game-Based Learning Management System called a pedagogical dungeon. Although the advantages of a game approach are obvious regarding the experiments that we conducted at our university, we have identified some research areas in which we need to go further.

Indeed, we are currently working on a user model adapted for learning games and updated thanks to observation traces exhibiting links between such a user model and the learner profile [2]. Moreover, we are aware that immersion is very important for motivation purposes [3]. For this reason we are developing a new 3D version of the learning game providing new collaboration tools.

Figure 16: Future version of our game-based learning environment

We are also working on innovative tools (such as, for example, “brain storming”, “sketch storming” or “model storming” tools for supporting and developing creativity) in order to enhance collaboration inside the game.

Acknowledgement

We greatly appreciate the commitment of the many people who have contributed to the development of the software presented here: M. Bodin, G. Dalla Costa, J. Demol, Y. Henri, L. Kepka, M. Michard, J. Pavlou, J. Roche, V. Saisset, J.C. Serra, D. Viala. This work was supported in part by the European Fund for Regional Development (Learning Games Factory Project) and by the Region Rhône Alpes (Cluster ISLE, subproject “Personnalisation des EIAH”).

References


Jean-Charles Marty first worked in a research centre of a private company, Cap Gemini, for 12 years, as a project manager and later as a scientific manager. He is now an associate professor at the University of Savoie (France). He leads the “Traces and Observation” group at the SysCom laboratory. His research interests are in the observation of collaborative activities, through the traces of these activities. The results of his research are applied to Technology Enhanced Learning, and particularly to learning game environments. He is a member of the scientific committee of the European Conference on Games Based Learning (2008 and 2009) and participates in several scientific committees in French workshops in the same domain. He is involved in three projects in this field funded either by the EU or by the French Government.

Thibault Carron is an associate professor of computer science at the University of Savoie. He is a member of the SysCom...
laboratory. He obtained his PhD in computer science at the "Ecole Nationale Supérieure des Mines de Saint-Etienne" in 2001. His current research interests deal with collaborative activity observation and with learning games. He is a member of the scientific committee of the European Conference on Games Based Learning (2008 and 2009) and of the French Conference "Rencontres interdisciplinaires sur les systèmes naturels et artificiels de Rochebrune" (since 2007). He is also involved in several projects in this field.
Observation of collaborative activities in a Game-Based Learning Platform

Jean-Charles MARTY, and Thibault CARRON, SysCom Lab, University of Savoie, France

Abstract—The work reported here takes place in the educational domain. Learning with Computer-Based Learning Environments changes habits, especially for teachers. In this paper, we wish to demonstrate through examples how learning sessions set up in a Game-Based Learning environment may be regulated thanks to observation facilities. Providing teachers with feedback (via observation) on the ongoing activity is thus central to being aware of what is happening in the classroom, in order to react in an appropriate way and to adapt a given pedagogical scenario. The first part deals with the description of our view of learning games illustrated through the "pedagogical dungeon": a game-based environment that we have developed. The second part of the paper focuses on the description of different ways and means to obtain information about the learning activities. It is based on traces left by users in their collaborative activities. The information existing in these traces is rich but the quantity of traces is huge and very often incomplete. Furthermore, the information is not always at the right level of abstraction. That is why we explain the observation process, the advantages of a multi-source approach and the need for visualisation linked to the traces. In the third part, we illustrate these concepts in the pedagogical dungeon equipped for observation and with the capacity for collaboration in certain activities. We pay particular attention to the regulation process of the collaborative activity by providing case studies explaining how the observation process enables the regulation actions. Finally, the feedback about the experiments presented is discussed at the end of the paper.

Index Terms—Learning Games, Trace, Indicator, Observation facilities, Regulation and Collaborative Learning Tools.

1 INTRODUCTION—Learning Games, Trace, Indicator, Observation facilities, Regulation and Collaborative Learning Tools.

Recent years have seen a rise in learning with Computer-Based Learning Environments. These environments offer functionalities that are recognized as being valuable, but students tend to consider them as unexciting. Agreeing with Vygotski’s school of thought and activity theory [42], we consider that the social dimension is crucial for the cognitive processes involved in the learning activity. Consequently, the question is 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 [36] and even in education [35, 37], it was decided to develop one 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.

This new way of learning changes habits, especially for teachers. Most of the time, a teacher prepares his/her learning session by organizing the different activities in order to reach a particular educational goal. This organization can be rather simple or complex according to the nature of the goal. For instance, the teacher may decide to split the class into groups, to ask the students to search for an exercise in parallel, to put different solutions on the blackboard, to have a negotiation debate about the proposed solutions, and to ask the students to write the chosen solution in their exercise-books. The organization of the different sub-activities in an educational session is called the "learning scenario" (see [20] for a definition of a pedagogical scenario).

In traditional teaching, namely in an environment with no computers, a teacher tries to be as aware as possible of his/her students’ performance, searches for indicators that allow him/her to know a student’s comprehension status and which activity of the learning scenario this student is performing. The teacher then adapts his/her scenario, e.g. by adding further introductory explanations or by keeping an exercise for another session. Once the training session is finished, the teacher often reconsiders his/her learning scenario and annotates it with remarks in order to remember some particular points for the next time. For instance, s/he may remark that the order of the sub-activities has to be changed or that splitting into
groups was not a good idea. In this case, the teacher is continuously improving his/her learning scenario, thus following a quality approach [32].

In educational platforms, and in particular in game-based learning environments, the teacher would like to have the same possibility as in traditional teaching, to be aware of what is going on in the classroom, in order to react in an appropriate way. Of course, s/he cannot have the same feedback from the students, since s/he lacks human contacts. However, in such environments, participants leave traces that can be used to collect clues, providing the teacher with awareness of the ongoing activity. These traces reflect in-depth details of the activity and can reveal very accurate hints for the teacher.

This observation feature in learning environments enables the provision of tools to the teacher allowing him/her to react to a particular situation, for instance: a student is in trouble; there are too many interactions among a group of people; there is not enough communication in a collaborative task. Being aware of these particular situations helps the teacher to adapt his/her following actions, that is to say the learning session. For instance, s/he can communicate with a student and help him/her or s/he can deactivate the communication tools within the group of participants. This adaptation of actions in a collaborative activity is also called “regulation of the activity”.

In this paper, we wish to point out the different aspects which enable the regulation of collaborative activities in a Game-Based Learning environment. We first describe our view of learning games illustrated through the “pedagogical dungeon”. We thus present the problems linked to observation through traces. Although this approach is very powerful, we will see that observation is a tricky task, with a lot of problems to be solved in order to obtain relevant observations allowing decisions to be made for an improvement of the collaborative process. Then, we classify the different kinds of possible actions to regulate the activity. We also introduce indicators, deduced from what has been observed, reflecting particular contexts. Finally, we give examples of regulation of the pedagogical activity in the “pedagogical dungeon”.

2 DESCRIPTION OF A GAME-BASED LEARNING ENVIRONMENT: THE PEDAGOGICAL DUNGEON

In this part, we describe our game-based learning environment. We first address the links between an educational activity and this game. Then, we demonstrate how the different participants interact with this game. Although the concepts presented in this part are not key research issues, we prefer to explain them to the readers in order to facilitate the comprehension of our approach presented thereafter and linked to the observation, awareness, and regulation in the collaborative activity.

2.1 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 [10] for a definition of Activity Theory) resulting from Vygotsky’s proposals [42]. 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 by Dunne in [10]).

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 & Henderson [16], we propose in this section to link the objects used in our game-based learning environment 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>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>Actors</td>
<td>Avatars (teachers, students)</td>
</tr>
</tbody>
</table>

TABLE 1: CORRESPONDENCE BETWEEN AT CONCEPTS AND GAME-BASED LMS REPRESENTATION

2.2 Breakdown of a learning session: rooms and topology

The learning session (or learning activity) is very often divided into different activities. This is the case when the teacher proposes to her/his students a set of interlinked exercises in order to reach a pedagogical goal. Each activity has its own local goal, generally a concept to acquire. For a student, successfully carrying out 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 the 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.
Application to the dungeon: Use case 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 by Koper in [22] or more flexible approaches [13], [18]. 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 and type of activities; for instance, an activity can be collaborative.
- The description of the available resources for each activity. Resources could be either local files (content included within the scenario’s definition) or links to online material. These files usually explain the topic of the activity. The teacher chooses the most appropriate form for these resources: a simple text, videos or even simulation applications, as is the case in [29].
- The definition of the validation procedure for each 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 Multiple Choice or open. In the latter case, the teacher will be in charge of validating the answers to that question. We would also point out that questions may be collaborative, in which case the whole team gives the answers.
- The definition of the constraints on the activities (organisational and logical temporal links). According to these constraints, the map of a dungeon is automatically generated and saved. (see [2] for details). Figure 2 is an example of the result of such a generation.

2.3 Enactment of a learning session with students

Similarly to in a Role-playing Game, 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: students can access knowledge through resources (documents found inside the game), via help from teachers, or from work with other students. For pedagogical purposes, the dungeon can be flexible. For instance, “teleportation portals” can lead to new rooms created dynamically during the session in order to resolve a problem.

Each room is dedicated to an activity. One 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 completes a test, an exercise or a quiz successfully. The assessments are thus also located in the room.

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 one, it only means that these students are performing the same activity. They can of course access the resources at the same time.

Application to the dungeon: Use case 2 - Embarking on the quest: moving, answering questions, obtaining keys.

In our virtual environment, we represent a student by an avatar (see avatar choice in figure 3) whose characteristics can evolve dynamically over time.
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 4, two students and the teacher (with the helmet) are present in a 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 4).

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, 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 5). 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 5 on the right) containing all the stated information.

An overall view from above (mini-map) is always supplied for the teachers during the game (fig 2). 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 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).

### 2.4 Collaboration in the dungeon

The teacher may want several activities to be collaborative. In that case, the rooms associated to these activities are collaborative places. These places require the students to answer in groups as indicated on the access door to the room. The advantage is to facilitate the exchange of data between students, making them ask questions and (ou to) co-resolve a problem. As in traditional teaching, these collaborative places offer the opportunity to create small groups in order to take advantage of the abilities of the various members of the group, who can explain their resolution strategy to each other.

Currently, a chat facility is provided in the dungeon rooms (see the translucent window in figure 4), but we can of course imagine other collaborative tools available in the collaborative places (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. The teacher may thus create or modify groups and dispatch students into them.

In collaborative rooms, the quiz is also collaborative: The crystal hiding a group activity has a specific colour and notifies the first student to arrive (see figure 6). Students in the same team must all be present in the room. They may exchange via the chat tool before answering the question. In the event of a correct answer being given for a collaborative quiz, a collaborative key is provided to all the members of the team.

As in “traditional classrooms”, a student may also collaborate with a teacher, for instance if s/he needs help from her/him. The teacher is also present and contactable in the game, represented with a specific (and impressive) avatar.
This Game-Based Platform has been used in real situations and from this experience we can easily conclude that it is attractive for the users. But, 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. Loss of perception for the teacher in these environments can make the tool unusable for him/her, because s/he can no longer regulate the collaborative activity.

3 OBSERVING A LEARNING ENVIRONMENT

The tracing activity is an appropriate way of reflecting in-depth details of the activity and of revealing very accurate hints for the teacher.

Unfortunately, traces are objects which are very difficult to manage and understand. We propose first to demonstrate the kind of problems linked to observation and to explain them through a pragmatic approach (experiments).

3.1 Pragmatic Approach to Observation Problems

Criterion1: Rich information available in Log Files but hard to extract and to exploit.

A first aspect to consider, central to the observation area, is the form of the traces. Many e-learning Platforms or Learning Management Systems are based on Web Servers [43], [1]. These servers easily supply logs (information concerning the connections on this server) stored in specialised files. We first used this information in an experiment carried out at our university. As we needed to analyse the new usages induced by the use of our local e-learning platform (“the electronic schoolbag”), we decided to work from the traces left by thousands of users.

The source of these traces was a web server providing data in the SQUID format, as for instance, 193.48.120.76 22/04/2003 04:25:31 POST TCP_MISS/200 http://www.univ-savoie.fr:443/Portail/logged_in FIRST_PARENT_MISS/www3-ssl2.univ-savoie.fr text/html.

It is obvious that these traces are not directly interpretable. They should be transformed and rewritten, in order to make it possible to understand them. For instance, 193.48.120.?? => “Connection to the e-learning platform from the university”.

Here, we want to identify connections matching the 193.48.120.?? address, meaning an access from the university site, where ?? can be replaced by any number from 1 to 255.

The traces were analysed a posteriori by a researcher in the “information and communication” field. From this experiment, new practices were revealed such as the use of the platform at home, but without using collaborative tools [5]. The experiment also pointed out the need to address the problem of treating the huge amount of data available in the log files.

Criterion2: Necessity for trace transformation for information extraction.

In order to better manage this huge and fine-grained information, we specified a transformation chain allowing the manipulation of traces (figure 7). The main purpose is to reach a good level of granularity, thus enabling an enhanced comprehension of the user behaviour [24].

This chain proposes several functionalities to manipulate the traces: filtering in order to reduce the huge quantity of logs, aggregation in order to change the level of granularity (abstraction) of the traces, transformation into a uniform format in order to take into account several log formats (SQUID, APACHE, I2S), or storage in a database and use of a Data Base Management System through SQL requests.

An experiment enacting this transformation chain allowed us to make an “observatory” tool, dedicated to non-computer scientist users. This tool enables the gathering of statistics on the usage of the “electronic schoolbag”, such as the number of connections, the types of users connected, the kind of preferred tools.

Visualization functionalities were added to this tool for obvious reasons of classical representations of statistical information.
data (graph representations, figure 3), thus adding a visualisation step to the transformation chain.

**Criterion 3:** A multi-source approach greatly enhances the understanding of the activity.

A certain number of research works linked to pedagogical platforms concern the formalisation of educational scenarios [22]. The teacher frequently foresees a sequence of activities to be performed during the learning session. This sequence, also called a scenario, guides the session, and comparing the learners’ activities and the predefined scenario becomes crucial [14]. This comparison allows the teacher to be provided with awareness of the ongoing activities, and enables the improvement of the scenario itself.

This is not an easy task, since the users can use simultaneously tools that are not integrated in the educational platform (forums, web sites, chat). We do not want to restrict our understanding to the tasks included in the predicted scenario. We want to widen the sphere of observation, so that other activities performed by a student are effectively traced. Even if these activities are outside the scope of the predicted scenario, they may have helped him/her to complete the exercise or lesson. We thus need to collect traces from different sources. It is therefore interesting, from a general point of view, to be able to take into account more than one source of data. Such an approach allows the deduction, from the multi-source traces, of non-foreseen behaviour. For instance, during a session in the pedagogical dungeon, a student may leave the game and start to communicate with other students to have the correct answer to a quiz.

Through an experiment described in [27], we have observed non-foreseen student behaviour. It is then possible to choose from the collected logs from different sources to specify, annotate or better explain what has happened during the session (see Fig 8), through a “trace composer”.

To help the user to better understand the trace generated, a graphical representation is a good support to make links between the learning scenario and the traces. We also take the different sources into account, in order to refine the understanding of the effective activity. We propose a metric to see how much of the activity performed by students is understood by the teacher, which is graphically represented on a “shadow bar”.

The comprehension of a general activity implies situating non-foreseen behaviour with foreseen activity sequences, as shown in figure 4 with exercise 1, document 1 read. It is thus useful to be able to reposition the users’ actions in the pedagogical scenario. In this experiment, we suggested a scenario improvement, since we pointed out that all the students who finished the learning session communicated with the teacher at the end of the first exercise in order to validate it. This validation making them more confident can thus be proposed in the scenario itself.

This study thus allows us to address problems that are linked to the scenario and the necessity of following the activity. Analysing the traces after the session provides the analyst with interesting results but does not solve the problem of giving the teacher the necessary awareness to react immediately to particular situations.

**Criterion 4:** Immediate analysis enables reaction. Visualisation improves the teacher’s awareness.

Detecting potential problems as soon as possible is a crucial issue. In order to alert the teacher to the fact that the collaborative activity is not progressing as expected, we need to compare the traces representing the actual activities with the ones mentioned in the predefined scenario and try to establish links between them. It is essential for the teacher to have a view of what is going on, in order to be able to react to given situations. The a posteriori analysis remains valid but can be expanded by analysis during the activity. New observation goals can also appear during the session. For instance, it can be useful to observe the status of the students during the first part of the session and to synchronise them before starting the second part of the session, being sure that everyone has acquired the requisite concepts.

This adaptive observation, needing high flexibility from the system, can be implemented through agents. A set of “pedagogical observation agents”, set up on the students’ computers, inspects certain users’ actions (the ones that are on focus for the observer) and notifies an awareness agent before invoking a visualisation agent to provide the teacher with the appropriate information. This distributed system is thus able to collect the significant logs directly on the machines through specialised agents [2].

The visualisation agent interprets the traces sent by the observer agents in order to display them on a dashboard for the teacher. An example of such an agent, called “classroomviz” has been developed (figure 9). Indicators are computed from activity traces and from a predictive scenario, offering the average performance time for each activity [15]. The teacher can thus easily follow the students that are behind with certain activities (red faces).
3.2 Architecture of an Observation Software

From the criteria pointed out in the previous section, we propose an architecture suited for taking these points into account.

Suggested analysis viewpoints reinforce the established phases linked to the observation lifecycle. These phases can be described as follows:

1. A collecting phase, where relevant traces are identified and collected before being treated by a dedicated agent (structuring or visualisation);
2. A transformation phase (structuring, abstraction) of collected data in order to make the rough traces more explicit and to make these traces understandable for the observer (researcher, teacher, or student);
3. And, a visualisation phase, where visualisation techniques will be used in order to reveal the semantics from the traces, make it easier to understand and help an analysis from a particular viewpoint. The phase is aiming at facilitating the interpretation of the on-going activity by a non-specialist.

3.2.1 Model of a Trace-Based System

We base our work on a model elaborated in collaboration with the SILEX Team of the LIRIS laboratory. This model, called Trace-Based System (TBS) defines the different modules associated with the different phases mentioned previously.

Figure 10 illustrates the process allowing the observer to interact with a traced e-learning platform in order to visualise and regulate the activity using the traces. The observer plays the role of a “trace composer”. S/he furnishes both the pedagogical scenario possibly expressed with IMS-LD [22], and the description of the experiment pointing out the analysis needs [Carron et al., 2006]. S/he thus sets up the e-learning platform by adjusting collection and transformation tools. Then, the experiment can be enacted, providing the analysts with usage feedback.

Collection phase

As demonstrated in figure 10, the collection phase is prepared before using the TBS and consists in gathering the traces generated in the e-learning platform. Trace collection is a complex computer science problem, due to the large volume of rough traces that one can possibly collect. This collection can be made through instrumented software according to the trace composer’s intentions [40] or through files generated by the operating system, or through dedicated spy software, such as key loggers. Another problem related to trace collection is the heterogeneity of rough traces (particularly in a multi-source approach), which necessitates studying a way to model them [19].

Transformation phase

The transformation phase is performed inside the TBS. The trace being an object in itself, the notion of Trace-Based System has emerged over the last few years, in order to allow and facilitate the exploitation and the interpretation of traces [38]. The functionalities of such systems therefore concern the manipulations of the traces. From the rough traces, a TBS offers a set of operations among these objects: filtering, joining or abstracting them.

When the results of these operations are still traces, they remain inside the TBS and they can possibly be used for other manipulations. A TBS also offers services allowing trace organisation, such as storage or historical mechanisms. Research questions related to this phase overlap with those of trace cleaning [7], trace aggregation according to temporal [25], semantic or syntactic constraints [41], trace rewriting or modelling [38], [6].

Trace visualisation

The visualisation phase consists in making requests among traces and in visualising traces. These visualisation tools are part of the interface between the TBS and a user observing the system. We decide to situate the visualisation and the request system outside of the TBS, since these tools do not fit the definition of trace manipulation as given by Settouti in [38]. Indeed, visualisation techniques produce results that are not traces. Visualisation consists in elaborating a graphical representation, ad-
apted to the analyst’s objective, from traces contained in the TBS. This representation can take many forms, such as a temporal 2D visualisation of a trace [15], of several traces [28], or a spatial 3D visualisation [8]. The visualisation system relies strongly on the analyst’s objective. For instance, the visualisation system must be able to provide the analyst with a real-time visualisation of the enactment of the users’ activities, and particularly to detect and show the users in difficulty. The system must also provide him/her with information about activities causing problems for these users. Finally, a visualisation of individualised paths showing the path of activities for each user must allow the analyst to make an intermediate assessment of the users’ progression (see label (C) in figure 9). This model led us to set up an architecture dedicated to the observation problem. We also took into consideration that a centralised approach could not offer adequate functionalities for diverse observations.

3.2.3 Distributed approach: an agent-oriented architecture.

Reasons for multi-agent architecture

The observation of collaborative activities has several salient characteristics that provide good reasons for an agent approach.

1. First, the problem is geographically and functionally distributed. Indeed, each student works on his/her own workstation and some information must be collected locally before being sent to other stations (for instance, the teacher’s) in order to be treated or displayed.

2. Furthermore, it is not possible to foresee which machine will receive or send the information. This depends mainly on the observation goal and on the students’ actions. This is thus highly context-dependent. There is no a priori solution to this problem because one cannot know the students’ behaviour in advance.

3. Each machine must remain autonomous in order to keep the progress of the pedagogical activity unchanged. It must also be able to communicate with each of the other machines, either to ask for information or to furnish some itself if necessary.

4. Finally, the set of collected traces possibly comes from different software and can be quite heterogeneous. It is thus difficult from a practical point of view to transfer the whole set of data coming from all the workstations to a unique station dedicated to the treatment of this data.

All these points justify the multi-agent approach. It would however be possible to add other advantages of such an approach, such as for instance the necessity for an observation system to be open or fault tolerant. The enactment of this kind of system must take into account the deployment context and the constraints imposed by the experiment in particular classrooms.

Multi-Agent Systems offer solutions for distributed systems in which autonomous software entities, the agents, can cooperate by means of interactions between them or with the environment. The choice of a multi-agent approach is thus particularly well adapted for such observation software. The general idea is to enact observer agents, autonomous software installed on each station, and that are in charge of collecting the relevant (according to a particular goal) actions performed on the station. This provides the teacher with a powerful means for being aware of the status of each student and thus being able to react in an appropriate way.

Multi-agent system (MAS) enactment

In order to set up the experiments described above, we have developed and used such a system. As we have already highlighted, the observation goal of the pedagogical experiment is central for the technical choices. The architecture enacted is represented in figure 11.

Fig 11: MAS Architecture for observation

It contains 3 types of agents that may be installed on the machines: the collector agents (C), the structuring ones (S), and the visualization ones (V). From a technical point of view, some agents (not represented in the figure) are only dedicated to the system functionalities. This is the case for instance for the facilitator agent (directory service: white and yellow pages), or the deployment agent (launching and killing agents).

Generally, the MAS are based on multi-agent platforms [33]. Our objective is however to keep our solution as simple as possible, and to be able to deploy it with a minimum of constraints. That is why we have chosen JAVA agents, that are platform independent and that can be launched easily on each station by a simple click. From a conceptual point of view, this solution is open and allows us to develop new agents when needed, without changing what is already working. Agents with specific functionalities (useful in particular situations) can thus be enabled or disabled when needed.

From a technical point of view, these observation features must work on any pedagogical platform. The software
environment becomes a trace generator. The agents are developed in such a way that they can be considered as a probe on any trace source. The main constraint is of course that the educational platform should provide traces and their interpretation model. Naturally, this “equipment” phase involves having access to the software of this platform, in order to have precise and rich traces.

4 APPLYING THE OBSERVATION FEATURES TO THE PEDAGOGICAL DUNGEON

In this part, we describe how all these observation features are included in a learning environment based on game concepts: the pedagogical dungeon. The three identified phases of observation will be taken into account.

4.1 Collecting phase

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 contain some parameters and may be flagged at any moment by any client of our application.

1. WorkshopArriving <UserName>, <WorkshopName>
2. WorkshopLeaving <UserName>, <WorkshopName>
3. WorkshopAnswering <UserName>, <WorkshopName>
4. WorkshopAnsweringWithContent <UserName>, <WorkshopName>, <AnswerContent>
5. WorkshopTeacherValidating <UserName>, <WorkshopName>, <TeacherName>
6. WorkshopTeacherValidatingWithContent <UserName>, <WorkshopName>, <TeacherName>, <AnswerContent>
7. WorkshopCorrectlyAnswering <UserName>, <WorkshopName>, <AnswerContent>
8. StudentConnecting <UserName> Informal awareness
9. HelpConsulting <UserName>, <HelpName>
10. Chatting <UserName>, <ChannelName>
11. ChatContentListening <UserName>, <ChannelName>, <SentMessage>
12. GroupCreating <GroupName>, <GroupType>, <UserName1>, <UserName2>, ...
13. GroupSplitting <GroupName>, <GroupSplitter>, <UserName1>, <UserName2>, ...
14. StudentDeconnecting <UserName>
15. TeacherConnecting <TeacherName>
16. TeacherDeconnecting <TeacherName>
17. TeacherHelpCalling <UserName>, <TeacherName>

The list of events occurring during the experiment is visualized through a Probe Visualisation tool (figure 12). This basic tool provides the teacher with coloured text (a colour can be associated with each kind of event).

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 only 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.

4.2 Transformation phase

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 ones (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 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 one 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 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 13 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.

![Figure 13: 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 teachers’ expected perception.

4.3 Visualisation phase

We set up experiments with this probe visualization and reached the conclusion that it was very difficult for the teacher to remain immersed in the dungeon (answering students’ questions) while looking at the probe visualisation. We were obliged to set up a new organization during the experiment session with two teachers, one playing the game and the other one looking at the probes and making regulation decisions.

We thus decided to design indicators calculated from the basic probes. These indicators help the actors to be informed of the on-going activity while remaining immersed in the game. Basically, we had two categories of indicators: 1) knowledge indicators that show what knowledge a particular student has already acquired. This is visualised through pieces of armour gained when answering the quizzes correctly. 2) behavioural indicators represented by auras directly on the avatar. For instance, in figure 14, the red spiral symbol was chosen by the teacher to display a loneliness indicator and white snow for empathetic behaviour.

![Figure 14: Two auras representing collaborative indicators.](Image)

5 EXAMPLES OF REGULATION IN THE “PEDAGOGICAL DUNGEON”

5.1 Reacting in particular contexts

The regulation consists in linking observations about a particular situation with a regulation action or set of regulation actions, thus defining regulation rules, that constitute our dedicated artefacts for regulating a learning session [42].

As stated previously, it is possible from different means to obtain a great number of traces that are most of the time heterogeneous. Although it is necessary to equip the stations with as many observation functionalities as possible in order to increase the observation possibilities, only certain specific information is really interesting for the different actors of the learning process at any one time. Therefore it is crucial to raise the abstraction level according to the user concerned in order to provide synthetic adapted information.

For that purpose, we have defined indicators based on observed traces to present a specific view on the on-going activity. Indicators can be considered as signals enabled when a particular, interesting situation happens. For example, the teacher can use an indicator to know which students are behind with an activity. The indicator is here the result of a calculus from the trace ‘entering an activity’ and the expected duration of this activity. In this particular situation, the teacher may want to accomplish a specific action and thus to regulate the activity, such as for example, making a new help file available for these students with an associated notification.

A difficult part of the regulation specification or more precisely of the definition of regulation artefacts is related to the description of the situation when a regulation action is to be considered. These situations are quite hard to describe since they are often complex ones, where a single indicator is not enough. That is why several indicators are activated and a set of several complementary ones allows us to define a context representing a more accurate view of the situation (see fig.15). In the previous example, we are able to extend and complement the information, should it be the case that almost the whole set of students is behind with this activity. In that event, the regulation action must be changed: the difficulty of this activity, the intelligibility of the statement or even the quality of the
resources provided must be reconsidered. The final regulation process will not be the same: there are now several other possible regulation actions adapted for such a situation.

![Diagram of a regulation rule](image)

Figure 15: Description of a regulation rule

All roles (see fig.10) are concerned by observation, even the students for reflexity purposes [12], [31]. Each role or person may define or select their own observation contexts and associate actions to them. It is a general way by which to create one’s own regulation context. A tutor can create a particular set of rules to which the students will work. But, we may easily imagine that a student can also create regulation rules (if the rights are enabled) in order to perform a subtask in a specified way. This is the case for instance when a student is designated as being responsible for a particular collective task (the tutor role for this subtask). Naturally, each rule is adapted to a specific goal: increase collaboration, develop metacognition, enhance memorisation, verify prerequisite(s), and magnify the transfer between knowledge and learned abilities.

### 5.2 Examples of regulation in the pedagogical dungeon

In order to understand well how our approach allows better regulation, we shall present 3 case studies.

**Case study 1:**

A certain concept is particularly important. The teacher wants all the students in the associated room to succeed. As a regulation, s/he needs to introduce dynamically new sub-activities for those who have failed. These sub-activities can be similar to the one that caused a problem and will be proposed only to the students who were unsuccessful.

This case study is high level. It is thus based on the structure of the pedagogical session. The regulation rule is rather easy to define: when a student gives a wrong answer then access to a new room is available.

The observation context is here a simple indicator based on one elementary probe: WorkshopCorrectlyAnswering=(user,"activity_X","false"). It is based on the success rate of an activity.

The regulation action is add(activity_Y,activity_X). As a result, a teleport (spiral icon) appears in this case and lets the user access the new activity _Y_.

**Case study 2:**

The teacher wants to know which exercises his/her students are failing. Most of them have difficulty in solving certain problems. In that case, the teacher can regulate the activity by adding new resources to help these students, by modifying existing ones or by opening a dialogue session with these students to provide them with hints to solve the problem.

This case study deals with the modification of the content of a specific activity during the session.

The observation context depends on the time spent doing the same activity. It is made up of an indicator which allows us to observe the duration of all the activities. It especially shows during the session that more than 50% of the students are behind with one activity. The teacher is thus warned of this situation and decides to modify the help file.

The regulation action is composed of two actions: modifying the help file and warning the students that a new help file is available. Furthermore, it is possible for the teacher to interact directly with a user via the interface by clicking on his/her avatar. A private chat session is thus initiated.

**Case study 3:**

The students are chatting a lot through the collaborative tools, but the results are poor, according to the teacher. S/he needs to regulate the activity by disabling the chat tool and letting the students continue the other activities individually.

This case study concerns the possibility of acting on a tool via its interface. The observation context deals with the quantity of messages exchanged in the same activity. An indicator “chatting too much” is helpful here to describe the context mentioned. The regulation action is to disable the chat facility access for the students concerned by the indicator and thus to act on tools available in the learning environment.

These 3 case studies illustrate different levels of regulation that can be exhibited in a learning environment. The different experiments that we have carried out in our university in real situations have shown that such an environment is well perceived by the students. Involved in an immersive pedagogical session, they are not exactly aware of the regulation process. Most of the time, they are amused by the appearance of new pedagogical resources. The disabling actions on tools are very efficient but more disturbing if no satisfying explanation is given. We have not focused on this fact here, but each regulation action should be used with a notification message. Regarding the teacher, the cognitive overload prevents him/her from being able to develop from scratch and add a new activity during the session. Currently, the content of such high-level regulation actions must be prepared before the start of the learning session.

http://mc.manuscriptcentral.com/tlt-cs
6 CONCLUSION AND PERSPECTIVES

In this article, we have demonstrated through examples how learning sessions set up in a Game-Based Learning environment may be regulated thanks to observation facilities. We have explained how to obtain relevant information about a specific expected situation and how to react through different levels of regulation actions directly during a pedagogical session.

We have also introduced indicators, deduced from what has been observed, reflecting particular contexts of observation. For future work, this implies extended research on the definition and classification of collaborative indicators and on a classification of the possible collaborative actions.

More precisely, we have described many actions on modelled elements of a pedagogical session that are relevant for regulation of the activity. We think that these actions or sets of actions should be seen as a means to resolve particular disturbing situations and could also be classified along with this point of view.

These concepts have been illustrated through a Game-Based Learning Management System called a pedagogical dungeon. Although the advantages of a game approach are obvious regarding the experiments that we conducted at our university, we have identified some research areas in which we need to go further.

Indeed, we are currently working on a user model adapted for learning games and updated thanks to observation traces exhibiting links between such a user model and the learner profile [2]. Moreover, we are aware that immersion is very important for motivation purposes [3]. For this reason we are developing a new 3D version of the learning game providing new collaboration tools.

![Future version of our game-based learning environment](image)

We are also working on innovative tools (such as, for example, “brain storming”, “sketch storming” or “model storming” tools) for supporting and developing creativity) in order to enhance collaboration inside the game.

ACKNOWLEDGEMENT

We greatly appreciate the commitment of the many people who have contributed to the development of the software presented here: M. Bodin, G Dalla Costa, J. Demolis, Y. Henri, L. Kepka, M. Michard, J. Pavlou, J. Roche, Y. Saisset, J.C. Serra, D. Viala. This work was supported in part by the European Fund for Regional Development (Learning Games Factory Project) and by the Region Rhone Alpes (Cluster ISLE, subproject “Personnalisation des EIAH”).

REFERENCES


issued Learning Technologies Published by the IEEE Computer Society, 2005, (pp. 476-481), Kaoshiung, Taiwan.


Jean-Charles Marty first worked in a research centre of a private company, Cap Gemini, for 12 years, as a project manager and later as a scientific manager. He is now an associate professor at the University of Savoie (France). He leads the ‘‘Traces and Observation’’ group at the SysCom laboratory. His research interests are in the observation of collaborative activities, through the traces of these activities. The results of his research are applied to Technology Enhanced Learning, and particularly to learning game environments. He is a member of the scientific committee of the European Conference on Games Based Learning (2008 and 2009) and participates in several scientific committees in French workshops in the same domain. He is involved in three projects in this field funded either by the EU or by the French Government.

Thibault Carron is an associate professor of computer science at the University of Savoie. He is a member of the SysCom laboratory. He obtained his PhD in computer science at the "Ecole Nationale Supérieure des Mines de Saint-Etienne" in 2001. His current research interests deal with collaborative activity observation and with learning games. He is a member
of the scientific committee of the European Conference on Games Based Learning (2008 and 2009) and of the French Conference “Rencontres interdisciplinaires sur les systèmes naturels et artificiels de Rochebrune” (since 2007). He is also involved in several projects in this field.