Observation of Collaborative Activities in a Game-Based Learning Platform

Jean-Charles Marty and Thibault Carron

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 by the teacher 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 observation of a learning environment, 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 assets of a multisource approach and the need for visualization linked to the traces. The second part of the paper focuses on our view of learning games illustrated through the “pedagogical dungeon,” a game-based environment that we have developed. In the third part, we illustrate these concepts in the pedagogical dungeon equipped for observation and with the capacity for collaboration in certain activities. 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

R
cent years have seen a rise in learning with Computer-Based Learning Environments. These environments provide functionalities that are recognized as being valuable, but students tend to consider them as unexciting. 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 [38] and even in education [37], [39], 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, offers new opportunities to let the students organize their learning activities, applying their own strategies according to what they know from their academic strengths and weaknesses. This way of self-regulating their activity [47] can be supported in such environments. When these game-based learning environments are multiplayers, another central issue concerns the collaborative learning aspects. The use of collaborative tools is increasing, allowing the students to coconstruct knowledge efficiently [28]. However, it is often difficult for users to know how to use these tools effectively, especially because the interactions take place in a social context. Self-regulation, coregulation, and socially shared regulation are precisely described in [20].

In this paper, we focus on another kind of regulation, the teacher’s regulation of the activity, where the teacher applies his/her own strategies in order to monitor the collaborative activity. 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 subactivities in an educational session is called the “learning scenario” (see [25] for a definition of a pedagogical scenario).

In a teacher-directed environment, 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.

In educational platforms, and specially in game-based learning environments, the teacher would like to have the same possibility as in traditional teaching, to actually 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 “teacher regulation of the activity.”

In this paper, we wish to point out the different aspects which enable the teacher regulation of collaborative activities in a Game-Based Learning environment. We first present how to observe a learning environment and 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 describe our view of learning games illustrated through the “pedagogical dungeon” and we explain how the observation can be set up in such an environment. We also explain how these observation features are useful for teacher regulation.

2 OBSERVING A LEARNING ENVIRONMENT

The tracing activity is an appropriate way of reflecting in-depth details of the activity and of revealing accurate hints for the teacher, for the student for metacognition purpose, for the pedagogical engineer [34] and even for the researcher looking for emerging usages [6].

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 (experimentations).

When s/he monitors a pedagogical session, a teacher is looking for information linked to the activity status. This information should be provided by the platform on the basis of the user traces. Let’s take an example of what the teacher should observe.

The teacher first checks whether the students are well connected to the platform. Then, s/he focuses on the activity, looking at which documents are accessed and in which order. The collaborative aspects may also interest him/her and s/he may want to know which students are helpful with the others or the numbers of exchanges in the different groups.

This observation scenario is rather basic, but we immediately see tricky items: we need to obtain the traces generated by another tool (chat or forum). All the information must be presented to the teacher in an understandable and immediately usable form (we can note that this statement is also mandatory if the students access their traces in a reflexive way in order to understand better how they have constructed their knowledge).

From this small scenario showing the interest to use trace data in a collaborative system, we propose several criterions to take into account when designing a trace-based system.

2.1 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 [46], [2]. These servers easily supply logs (information concerning the connections on this server) stored in specialized files. We first used this information in an experiment carried out at our university. As we needed to analyze 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, for instance,

193.48.120.76 22/04/2003 04:25:31 POST TCP_MISS/200

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 analyzed 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 [6]. 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 (Fig. 1). The main purpose is to reach a good level of granularity, thus enabling an enhanced comprehension of the user behavior.

This chain proposes several features 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, and 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 noncomputer 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, and the kind of preferred tools.

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

**Criterion 3: A multisource approach greatly enhances the understanding of the activity.**

A certain number of research works linked to pedagogical platforms concern the formalization of educational scenarios [26]. 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 [16]. 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, and 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 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 multisource traces, of unforeseen behavior. For instance, during a session in the pedagogical dungeon, a student may leave the game nonforeseen behavior. For instance, during a session in the pedagogical dungeon, a student may leave the game or stay for a long time, being interested in other parts of the game. This nonforeseen behavior with foreseen activity sequences, as shown in Fig. 2 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 therefore 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. Analyzing 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. Visualization 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. Having such indicators at his/her disposal can however lead the teacher to overreliance on them. It is thus important that the teacher uses these facts as decision tools but keep the entire control of the activity. 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 synchronize 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 visualization 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 specialized agents [27].

The visualization 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 (Fig. 3). Indicators are
computed from activity traces and from a predictive scenario, offering the average performance time for each activity [17]. The teacher can thus easily follow the students that are behind with certain activities (light gray faces (slightly behind) or dark gray (really late)). S/he can also see what activities a given student has followed (e.g., for Melanie in the picture).

2.2 Architecture of an Observation Software

We need to make this trace data approach operational, by finding a process allowing to manage the traces from their capture to their visualization. From the criterions pointed out in the previous section, we can list several steps of this process. A first phase consists in collecting data (criterion 3: select relevant sources of observation and criterion 1: get a lot of traces). Then, all information has to be structured (criterion 2: transformation). Finally, this information is displayed in an adapted manner, as soon as possible for the teacher or the student (criterion 4: visualization).

We propose here an architecture suited for taking the following phases into account:

1. A collecting phase, where relevant traces are identified and collected before being processed by a dedicated agent (structuring or visualization);
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 visualization phase, where visualization 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 ongoing activity by a nonspecialist.

2.2.1 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.

Fig. 4 illustrates the process allowing the observer to interact with a traced e-learning platform in order to visualize 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 [26], and the description of the experiment pointing out the analysis needs [5]. 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. The different steps of the process of trace data interpretation are displayed on the left part of the figure, while the right part refers to the trace data themselves.

Collection phase. As demonstrated in Fig. 4, 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 [42] 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 multisource approach), which necessitates studying a way to model them [24].

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 [40]. The features of such systems therefore are related to 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 organization, such as storage or historical mechanisms. Research questions related to this phase overlap with those of trace cleaning [7], trace aggregation according to temporal [30], semantic or syntactic constraints [43], trace rewriting, or modeling [40], [13].

Trace visualization. The visualization phase consists in making requests among traces and in visualizing traces. These visualization tools are part of the interface between the TBS and a user observing the system. We decide to situate the visualization and the request system outside of the TBS, since these tools do not fit the definition of trace manipulation as
given by Settouti et al. in [40]. Indeed, visualization techniques produce results that are not traces. Visualization consists in elaborating a graphical representation, adapted to the analyst’s objective, from traces contained in the TBS. This representation can take many forms, such as a temporal 2D visualization of a trace [17], of several traces [33], or a spatial 3D visualization [8]. The visualization system relies strongly on the users’ objective. For instance, the visualization system must be able to provide the analyst with a real-time visualization of the enactment of the users’ activities, and particularly to detect and show the students in difficulty. The system must also provide him/her with information about activities causing problems for these students. Finally, a visualization of individualized paths showing the path of activities for each user must allow the analyst to make an intermediate assessment of the users’ progression (as for label (G) in Fig. 3).

2.2.2 Distributed Approach: An Agent-Oriented Architecture

The model proposed in the previous section led us to set up an architecture dedicated to the observation problem. We also took into consideration that a centralized approach could not offer adequate functionalities for diverse observations.

Reasons for multiagent 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’ behavior 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 these data.

All these points justify the multiagent 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.

Multiagent 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 multiagent 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.

Multiagent 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 Fig. 5.

It contains three 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 multiagent platforms [36]. 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.
In the next part, we describe how this approach can be set up in a game-based learning (GBL) environment that we have developed.

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

In the description of our GBL environment, we first address the links between an educational activity and this game. Then, we demonstrate how the different participants interact with the environment. We also mention how the “observation” approach can improve the use of a GBL environment. 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 teacher regulation in the collaborative activity.

3.1 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 [11] for a definition of Activity Theory) resulting from Vygotsky’s proposals [45]. 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 artifacts (defined by Dunne in [11]).

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 and Henderson [21], 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></td>
<td>Room for a (sub) activity</td>
</tr>
<tr>
<td>Link between activities</td>
<td>Corridor</td>
</tr>
<tr>
<td>(sub) Activity</td>
<td>Crystals (Exercises)</td>
</tr>
<tr>
<td>Condition / Requisite</td>
<td>Room Door</td>
</tr>
<tr>
<td>Resources</td>
<td>Knowledge Spheres</td>
</tr>
<tr>
<td>Assessment, Validation</td>
<td>Door Key</td>
</tr>
<tr>
<td>Communication tool</td>
<td>Chat window</td>
</tr>
<tr>
<td>Actors</td>
<td>Avatars (teachers, students)</td>
</tr>
</tbody>
</table>

In the next part, we describe how this approach can be set up in a game-based learning (GBL) environment that we have developed.

3.2 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 his/her students a set of interlinked exercises in order to reach a pedagogical goal. Each activity has its own local goal. For instance, in our GBL environment, an activity can lead students to acquire a new concept. It is the case for instance when a student learns the difference between classes and instances while studying Object-Oriented Concepts. 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 Fig. 6.

Fig. 6. An example of a scenario seen as a dungeon topology.

The creation of a pedagogical session in the game can be seen as the creation of a scenario, usually written with IMS-LD described by Koper et al. in [26] or more flexible approaches [15], [23]. If the teacher wants to construct a pedagogical session in the dungeon, s/he interacts directly with a session builder. This tool allows:

- 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 or links to online material. These files usually explain the topic of the activity.
- 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, for instance, by answering a set of questions. The resolution of these questions may be collaborative, in which case the whole team collaborates through a dedicated tool (e.g., a chat or a post-it wall) before giving the answers.
- The definition of the constraints on the activities (organizational and logical temporal links). According to these constraints, the map of a dungeon is automatically generated and saved. (see [4] for details). Fig. 7 is an example of the result of such a generation.

This way of organizing activities is positive since it is flexible. The teacher can create several parallel activities with subgroups, for instance, according to their knowledge. However, there is a major drawback in such an approach. It is linked to the general awareness of the teacher and of the students. Without additional specific tools, it is impossible to have a general view of who is doing what, of who is progressing well in the session, and who has difficulties. We can thus expect from a trace approach that the teacher and the students are aware of the progress of the others thanks to the digital traces left on the platform.

### 3.3 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 subactivities in order to acquire knowledge. We represent an actor by an avatar whose characteristics can evolve dynamically over time. 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 solve a problem: when several students have failed in the resolution of a problem, the teacher can create an additional activity to explain this particular point and test the students’ knowledge before continuing the session. The problem here is to create dynamically a new room without changing the existing topology of the dungeon; otherwise the students can be disoriented. We thus decided to use the metaphor of “teleportation portals” which are enabled in order to let these students follow this reinforcement activity.

Each room is dedicated to an activity. One can find explanatory resources such as texts, links, and videos. These provide the student with information that the teacher has placed here and that is connected to the task. 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. In Fig. 8, two students and the teacher (with the helmet) are present in a room. 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 that 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 (activated through the crystal) 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 to be ordered in the dungeon: students may thus choose between several exercises if the teacher wishes to offer this flexibility.

The visualization 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.

### 3.4 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 doors to the room. The advantage is to ease the exchange of data between students, making them ask questions to coresolve 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, but we can of course imagine other collaborative tools available in the collaborative places (shared space, forums, etc.) in order to coconstruct a common knowledge. 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 color and notifies the first student at arrival. 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 every member of the team.

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 (see details in section “experiment”). 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.

3.5 Need for Trace Data for Monitoring and Assessment

New usages appear when using such environments. For instance, groups can be created or changed easily. We can easily imagine that trace-based tools can help the users here.

The teachers may lack information in order to monitor the session: Who is doing what? Who is belonging to which group? What is the behavior of a particular student in a group? Which activities has this student followed and were they successfully achieved?

The students can also take advantage of awareness tools based on the digital traces. They can, for instance, ask for their level of involvement in a collaborative activity or they can follow what the comembers of their group are doing. They can also have a look of other ways of doing a particular task, by replaying the traces generated by another student having achieved the same goal but in another way.

We have also mentioned earlier that the characteristics of a student are evolving over time. These characteristics are usually gathered into a user model [44] and the traces can of course be a good way to update the user model, according to the actions performed or the results obtained by the students [31].

In our experiment, we have chosen to focus on trace usage for facilitating the monitoring of the collaborative activity for the teacher. The architecture proposed in Section 2 is general enough to propose other indicators to the students. The reason why we have currently dedicated the indicators to the teacher is linked to the design of the indicator. In fact, we have noticed in our previous experiments that the indicators must be coconstructed with their future users in order to be really accepted and useful for them [18], [31]. There is thus a need to acquire this specific knowledge and we find it easier to do the knowledge acquisition from the teacher than from the students, especially because they were not familiar with that GBL environment.

In order to provide the teacher with several useful indicators from the traces generated within the GBL environment, we need to apply the process explained in Section 2: we need to define what to observe, how to transform the “rough” traces, and what to visualize.

4 Applying the Observation Features to the Pedagogical Dungeon

In this part, we describe how all these observation features are included in the pedagogical dungeon. The three identified phases of observation are 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 the list of the basic facts to be observed through an Application Programming Interface (API). 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>, <Comment>
7. WorkshopCorrectlyAnswering <UserName>, <WorkshopName>, <Comment>
8. StudentCorrectlyAnswering <UserName>, <WorkshopName>, <Boolean>
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>.

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 ongoing activity.

As stated previously, 17 basic indicators (simple probes) have been extracted from the pedagogical dungeon and three 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. Fig. 9 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.

Fig. 9. Complex probes creation interface.

The possibility of having basic or complex probes in the platform helps solving the problem of lack of feedback for the users. 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 personalization of the teachers’ expected perception.

4.3 Visualization Phase

We set up experiments with this probe visualization and reached the conclusion that it was very difficult for the teacher to understand what was going on, due to the cognitive overload caused by the huge amount of messages arriving in a dedicated window. Although it was possible to color the messages regarding their categories, the teacher was unable to remain immersed in the dungeon (answering students’ questions) while looking at the probe visualization. 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 ongoing 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 visualized through pieces of armour gained when answering the quizzes correctly. 2) behavioral indicators represented by auras directly on the avatar. For instance, in Fig. 10, the spiral symbol was chosen by the teacher to display a talkative indicator and white snow for empathetic behavior.

Fig. 10. Two auras representing collaborative indicators.

4.4 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 artifacts for regulating a learning session [45].

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 ongoing 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 artifacts 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. 11). 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.

All roles are concerned by observation: the teacher for monitoring/regulation purpose; the researchers when they need to understand the process of collaboration, as it is the case in [29] for argumentation; the students for reflexivity purposes [14], [35]. 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, or when s/he takes a specific role in the group (same approach as in gStudy when using the “guided chat” [20]). 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 memorization, verify prerequisite(s), and magnify the transfer between knowledge and learned abilities.

5 EXPERIMENTS AND DISCUSSION

5.1 Description of an Experiment

Similarly to [2], our approach is very pragmatic and mainly based on empirical experiments. We now illustrate how a training session can be immersive and flexible in such an environment, through an experiment in the pedagogical dungeon.

5.1.1 Conditions and Methodology of the Experiment

This experiment was carried out in our university with colocated settings. During the experiment, six groups of 15 students with their teacher were present successively in the classroom equipped with 15 computers. Concerning the social presence perception [9], players were oriented away from each other, limiting mutual eye contact, natural reciprocation of approach or avoidance cues and mirroring. The students were 18 years old and familiar with computer use. In the six groups, there was an almost equal distribution between male and female. Each student accessed the virtual environment through his/her workstation, and had a personal (adapted) view on the world. These students used the environment for approximately one and a half hours.

They were explicitly allowed to communicate through the chat tool provided with the system (integrated within the Dungeon) and were warned that they would be observed concerning the use of the system. Two observers were present in the classroom. The students were free to refuse this observation (the same practical work was available outside the learning environment), but everyone agreed to follow the proposed protocol.

5.1.2 Pedagogical Objectives

The course was dedicated to the Operating Systems field. The learning content dealt with “Unix basic shell commands.” The aim of the session (role-playing game) was to assess the knowledge and know-how of the students about the latter. A story guided the knowledge quest thanks to metaphors: teleporting, cloning and destroying objects in the world representing, respectively, moving, copying, and deleting files in the file system. The challenge is encouraged through NPCs who propose a coherent contest, to test new futuristic developments. Immersion is reinforced when the users’ actions have a direct impact on the objects of the world. In this experiment, an artifact corresponding to the system console was present in the game. Actions correctly achieved on the file system were visible in the game: a book (object) was moved, duplicated, or deleted in the game thanks to effective actions on the file system book (file). Eventually, the teacher was present in the game via an avatar: it was possible to chat with him, to ask for help for example.

5.1.3 Evaluation and Results

From the questionnaires from students. At the end of the experiment, the students were asked to fill in questionnaires to give feedback about their feelings regarding their work session. Twelve questions (ranking and open-ended) were
used. The questionnaire evaluated aspects relating to several parts of the learning game (pedagogical content, scenario-story, collaborative activities, and accessibility of the user interface). The final question let the students propose improvements concerning weak points of the game.

The initial objective concerning motivation was reached. The students were unanimous in preferring to work with this environment rather than do conventional practical work on workstations, and more generally were very enthusiastic about this kind of experiment. In a 5-level ranking (ranging from definitely not (1) to strongly agree (5)), the results were mainly agree (4) or even (5), concerning the addition of more collaborative aspects. They liked play to learn, see each other and be able to chat together, to discuss their solution. We must say that the chosen activities matched very well with the game. The results obtained from the questionnaires were confirmed by the fact that the students did not take a break.

Contrary to some preliminary experiments with video-games, our pedagogical resources or background information [12] were well integrated in the game and adapted for the game experience. Nevertheless, some questionnaires suggest possible improvements, mainly linked to the interface or to the scenario (girls did not like the scenario very much). Apart from their perception of the scenario, male and female students appear to react similarly to this learning game, but this may be due to the small sample size which is not a true representation of the population, as it is now well known that specific gender aspects have to be taken into account [22].

Finally, we must precise that several experiments have been set up, each time with precedent suggestions or improvements taken into account. For instance, a “loss of positioning” reported by the students in a previous version has been solved through a minimap (small window allowing to see the global map and our current position).

**From an interview with the teacher.** It was also important to interview the teacher (short interview, with open questions). From his point of view, in the light of a previous experiment, it was mandatory to have tools supporting him in the monitoring task (which have to be prepared or anticipated). Several indicators elaborated from transformed traces were therefore set up in our environments in order to meet this requirement. However, the teacher reported the cognitive overload problem that we have explained in the Section 4.3 and asked for indicators directly in the game. This led us to develop specific tools to represent the results of some indicators through objects in the game. He was also asking for a “replay” facility in order to analyze more precisely the students’ behavior in the game. Finally, he was impressed by the motivation of the students.

**5.1.4 Discussion Concerning These Results**

We are aware that many bias exist regarding these results. For instance, the fact that the students enjoy this approach is probably due to its playful aspect and its novelty. The students declare that they have learned from the GBL environment, but we have no measure of the gain (if any) compared with a traditional approach.

We have also noticed that students are not always aware of the regulation process. Most of the time, they are amused by some “strange things” happening to them, as it is due to random (frequent in games). The accessibility of new pedagogical resources or the impossibility to activate a tool is very efficient but very disturbing if no satisfying explanation is given. We thus believe that explanation is an important part of the regulation process for the people involved in the process (as the definition of new rules in a group that must be clearly stated).

The most positive point is probably the motivation induced by the use of these tools. This is an encouraging factor for constructing elaborated scenarios with more collaborative activities (as the one mentioned in the next paragraph) to coresolve problems and allow the students to monitor some of the processes by themselves. Regarding these future experiments, we believe that the skills to be improved through the GBL environment must be identified before and the observation data set up to measure the effectiveness of the tools properly. We can find in [19] a large overview of such measures dedicated to CSCL.

**5.2 Discussion: How Trace Data Can Help in Other Kind of Collaboration**

In this paper, the trace data are useful mainly for the teacher regulation of the activity. This approach can also operate on other kinds of collaboration.

In Learning Adventure [1], another game developed in our team, we have designed a collaborative tool called the post-it wall (see Fig. 12) and we would like to relate quickly what happened related to socially shared regulation [20].

In one of the activities for learning Object-Oriented Concepts in this environment, the students were asked to use this wall to provide the class diagram of an ecosystem whose components were present in the game. The solution had to be the result of the common thinking from a group of 15 students. From the data traces, it was possible to see the main participants in this collaborative activity, through the pie chart in Fig. 12, each division showing the percentage of actions on the wall for one student. It came out that six students started this activity before the others (that were still exploring the world). These six students were first manipulating the post-it notes inefficiently, moving notes that someone else just positioned elsewhere. But they quickly understood how to regulate the activity through the chat and obtained an acceptable diagram before the other students joined them.

No precise measure has been performed in order to know the impact of the indicator, which was also available for the students. It seems that for these students, the motivation to participate was higher, since everyone wanted to stay at the same level of participation, showing thus that the result.
really belongs to the group. The problem encountered by the teacher here is that the construction of the solution by these students is correct but too many students did not have the chance to participate. This has been easily corrected by setting up a new group doing the same activity, although this was not part of the initial scenario.

Another remark concerns the plans made by these six students. Obviously, these students wanted to go as fast as possible in this activity. This is mainly a cultural problem that leads to this kind of behavior, since the teacher had given no constraint of time. It is one of our thoughts to have a reward policy concerning collaboration to be included in the game and to measure its effectiveness.

Another quite simple way of using trace data is to replay them. For metacognition purpose, we have the possibility in the dungeon to activate “ghosts” of previous activities. These ghosts replay what the avatar of a student had done during a session. The idea is that in a session, you can “follow” your avatar, redo the same actions or decide to try something else. The teacher can also give some students access to other students’ ghosts having reached the objective using a different path.

The observation through trace data is also a powerful tool for codesign of a collaborative application. We believe that a collaborative tool in order to be used properly must be codesigned with the users. Too often, as computer scientists, we develop new collaborative tools with very nice features, but whose functionalities are not obvious for the users. In that case, tracing the collaborative activities allow to understand better how group of users use the new tools and can help to redesign the tool when needed.

6 CONCLUSION AND PERSPECTIVES

In this paper, 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. More precisely, we have described many actions on modeled elements of a pedagogical session that are relevant for regulation of the activity.

These concepts have been illustrated through a Game-Based Learning Management System called a pedagogical dungeon. Although the assets 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 [31]. As we are also working on collaboration and work within groups, we are currently elaborating a group model. This group model should be more than the sum of the models of users composing the group. For instance, Group efficacy, Interpersonal attraction in collaboration, Complementary skills in the group, and Preferred ways of communication are characteristics of the group rather than of a single person.

In order to understand more deeply the cooperation strategies of a group, we need to extend our model of TBS and introduce an additional layer for group traces. This layer will avoid costly transformation for grouping the traces generated by the different members of a given group.

Finally, immersion is an important motivation factor [3], and we are developing a new 3D version of the learning game providing new collaboration tools (e.g., the post-it wall presented earlier). We are also working on innovative tools (such as, for example, “brain storming,” “sketch storming,” or “model storming” tools) in order to enhance collective creativity inside the game.

ACKNOWLEDGMENTS

The authors would like to particularly thank their colleagues and friends from ESC Chambéry, Laure France and Jean-Mathias Heraud. They were in their group from the beginning and they discussed extensively with them most of the ideas presented in this paper. The authors 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, J. Depoil, Y. Henri, L. Kepka, M. Michard, J. Pavlou, J. Roche, V. Saissat, J.C. Serra, and 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


