CARTE: An Observation Station to Regulate Activity in a Learning Context

Christophe Courtin

Chapter 13
CARTE: An Observation Station to Regulate Activity in a Learning Context

Experiments in a Practical Class

Christophe Courtin

Abstract This chapter discusses the introduction of a new concept called "regulation" into a use model, which is part of a theoretical observation model called trace-based system (TBS). This concept defines a retroaction mechanism in an observation station. We present the results of experiments, in a learning context, with a prototype observation station called Collection, activity Analysis and Regulation based on Traces Enriched (CARTE).

13.1 Introduction

In this chapter, we present in greater detail a prototype of the observation system Collection, activity Analysis and Regulation based on Traces Enriched (CARTE) designed in our laboratory (Courtin, 2008) and especially the feasibility of a regulation mechanism, which is based on an observation instrumentation.

An observation station is a system based on a theoretical observation model that provides observers (e.g., teachers, learners, and experts) with observation services. We present a prototype which is coupled with collaborative learning software tools that enable participants to communicate with each other, to organize the group work, and to produce content collectively. We use such a system to keep track of learning activities in order to provide observers (e.g., learners, teachers, and experts) with information which helps participants in their own practice.

The final research question is how may regulation influence the learning process to take into account the social dimension? However, it is worth noting that our

C. Courtin (✉)
Laboratory Systèmes Communicants, UFR Sciences Fondamentales Appliquées, University of Savoie, 73376 Le Bourget-du-Lac Cedex, France
e-mail: christophe.courtin@univ-savoie.fr

This work is part of the research cluster ISLE (*Informatique, Signal, Logiciel Embarqué* or Informatics, Signal, Embedded Software), in the project "customization of the technology-enhanced learning systems (TELS)", supported by the region Rhone-Alpes in France.

J.M. Spector et al. (eds.), *Learning and Instruction in the Digital Age*,
DOI 10.1007/978-1-4419-1551-1_13,

professional practice as teachers allows us to manage instrumented collective learning situations (ICLS), but it is advisable to be cautious about the semantic value of the results obtained by a system of observation of corresponding activities. Therefore, we will focus on computing aspects without forgetting the final objectives which are centered essentially on computing environment uses in a collective learning context. In other words, we attempt to evaluate the feasibility of our regulation mechanism, implemented in the prototype of the CARTE observation station, based on the theoretical model called trace-based system (TBS) described below. The evolution of this work will consist of evaluating how relevant this regulation is to support the social dimension in collaborative learning situations.

In this chapter, we first restate briefly the concepts of the theoretical observation model, which underlines the CARTE observation station model, and we shall attempt to find answers to the activity regulation problem with the help of retroaction. From this, we describe the corresponding working process with the architecture of the CARTE system. Finally, we present experiments which highlight the feasibility of a regulation mechanism in an observation station with the technical results of the CARTE prototype. We compare the last experiment to a previous one that was similar, but without the retroaction mechanism, in order to evaluate the benefits of regulation in collaborative learning activities.

13.2 Traces

This chapter provides a computing point of view about the concept of regulation in instrumented collective learning situations (ICLS), which widely covers other scientific fields, especially those in human sciences. We have introduced the term "regulation" (Martel et al., 2004) to represent all mechanisms that enable participants to organize themselves in a shared environment. The objectives of the regulation are to ensure social control (i.e., respect of social rules such as not talking to other participants) and to fit the activities to the pedagogical objectives according to a specific context (e.g., suggesting new exercises to students having difficulty). We propose a more precise definition of regulation in the following part.

We define an instrumented collective learning situation (ICLS) such as a traditional collective learning situation (i.e., in situ and in real time, such as a classroom) supported by a computing system which enables participants to communicate with each other, to coordinate their own activities, to produce content, and to regulate this situation as explained above. In our research work, we use the computing device to obtain clues about the learning context. From this, we chose to set up a specific instrumentation based on the computing events in order to obtain more elaborate traces (than the log system) of the participants' learning activities. From the point of view of a human observer (e.g., a teacher), the final traces, that is to say those with an abstraction level close to that of the human observer, are called indicators (i.e., they determine success or failure in an exercise). These indicators enable the description of actions as they are defined by the observer him/herself. The instrumentation technique we use is equivalent to the log system technique, except for the fact that it takes place at the level of the software tools themselves and close to their use model.

We would accept the idea that going from a traditional to an instrumented situation would benefit all the participants if the latter maintained the activities' human dimension. It is worth noting that computing environments' techno-centered design, which is too widely spread (Rabardel, 1995), limits use observation. The difficulty is increased by the singular nature of the use for which a model cannot be anticipated. With that in mind, we maintain that it is essential to involve interactants very early in activity analysis. This involvement occurs in the observation system design stage and thus in that of corresponding models as explained below.

We observe mediated situations with the help of organized knowledge which stems from computing environment instrumentation. The choice of instrumentation for observation is based on the idea that observation objectives are predefined by the observer him/herself (teacher, sociologist, ergonomist, etc.) (Carron, Marty, AQ1 Heraud, & France, 2005). We then define the concept of "observed," which represents data relative to an activity observation. An activity trace, which is built from a use model, is composed of time-situated "observed" sequences. The use model supplies the necessary semantics to interpret the elements that compose the activity trace. The association between a trace and the corresponding use model (Settouti, AQ2 Prié, Mille, & Marty, 2006) is termed MTrace. In order to interpret the MTrace, the human observer has to participate in the definition of the semantics of the use model elements.

We have placed traces at the heart of the theoretical observation model called TBS, presented in Fig. 13.1, to the definition to which we have contributed with other actors in the Technology Enhanced Learning and Teaching (TELT) field (Settouti, Prié, Mille, & Marty, 2006). We add a retroaction module to this model that we will describe in the next section. We propose a generic trace structure, presented in Fig. 13.2, in order to take into account the various trace sources (logs, snapshots, annotations, etc.). Common elements among all these sources (e.g., description) define the meta-data and specific elements of each source (e.g., a *page* in a text editor, a *table* in a structured chat room) define optional parameters. The observer defines these parameters' semantics in the use model. As specified above, we have chosen to collect traces by means of instrumentation of the computing environment's tools. We introduce two types of templates: signals and sequences (see Fig. 13.2). A signal generally represents basic information corresponding to an elementary action on software tools (e.g., saving a page). A sequence represents enriched information, which makes sense for an observer and which stems from composite actions represented by signals or even by other previously created sequences (e.g., edit_page + save_page = edition). We distinguish several levels for sequences according to how they are composed: signals and/or sequences of level n (Courtin & Talbot, 2007).

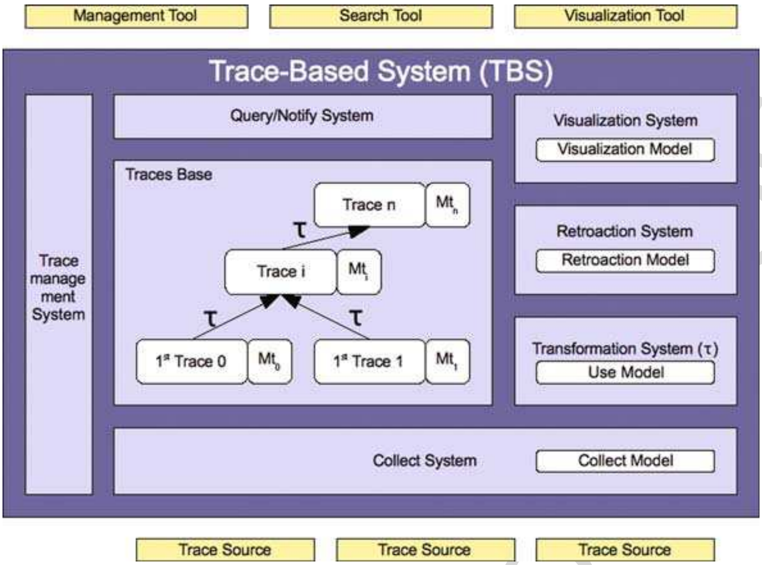

Fig. 13.1 Trace-based system (TBS)

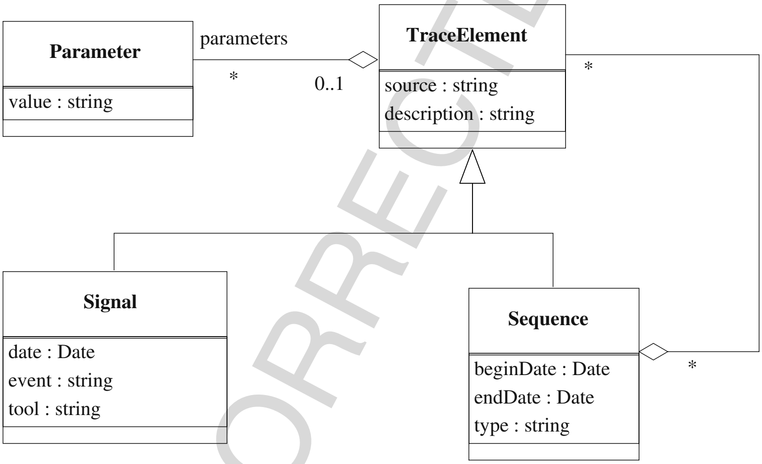

Fig. 13.2 Trace structure

13.3 Retroaction and Regulation

The introduction of regulation concerns taking into account a fourth dimension in the classical clover model (collaborative work model), which is divided into three spaces: communication, coordination, and production (Ellis & Wainer, 1994). A fourth space entitled "regulation," which is orthogonal to the other three spaces, AQ4

is added to this model in order to take into account the activities' social dimension (Chabert, Marty, Caron, Vignollet, & Ferraris, 2005). Regulation is a very broad concept and it is necessary to define it in this chapter before presenting our research work. We consider all the mechanisms for activity configuration in the computing environment to ensure coherence between work in progress and activity objectives (e.g., workgroup management during the activity, modification of the choice of proposed exercises according to the situation, etc.) included in regulation.

The CARTE observation station we have produced matches the theoretical observation model specification, presented in Fig. 13.1, where a use model is associated to traces. MTraces thus created may be combined by means of transformations (τ) in an analysis tool to produce higher abstraction level MTraces, or language level ones, in order to help a human observer to interpret activities in a given application domain. Therefore, our application context constrains us greatly. Indeed, users are not involved in the design of the generic software tools which compose the computing environment used for their activities. This constraint hinders use analysis significantly. We get around it by involving the observer in computing environment use modeling. The resulting model, built by the observer, defines the software tools' working process without having to modify them. This approach's originality comes from the independence of the various use models and the software tools and their adaptability to match the various observation objectives. In other words, it is a matter of adapting a generic tool to a specific use by modifying the working process from traces, without modifying the tool itself. These different use models enable one to specify social rules (e.g., forbidding unauthorized intrusion in a chat-room discussion), which imply different uses that from the user's point of view are interesting for his/her activity.

The analyzer tool we have associated to the CARTE observation station, of which the working process is described in Fig. 13.3, provides "awareness" and/or

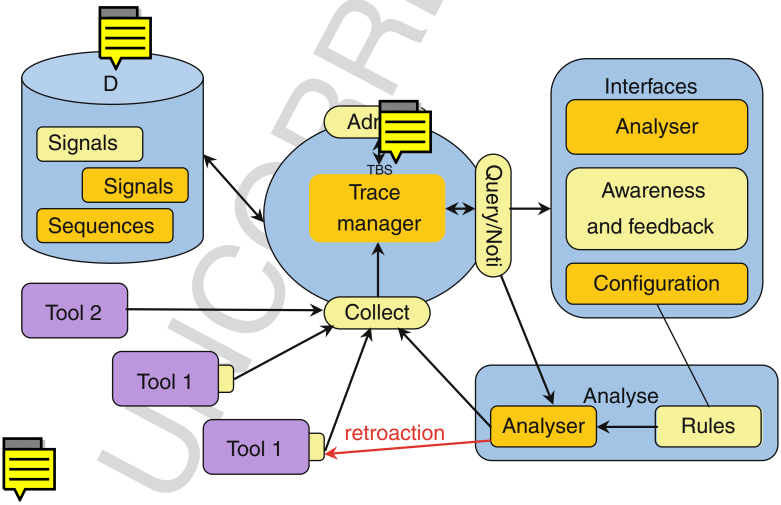

Fig. 13.3 CARTE architecture

This figure will be printed in b/w

"retroaction" type traces. In the first case, the observer receives information relative to "observeds" described with the use model. By interpreting this information, the user/observer (teacher, student, etc.) will be able to act on the computing environment to move the activity toward predefined objectives. In the second case, the traces trigger the execution of actions on the tools in order to facilitate their use (e.g., task linking) or even to enrich their working process (e.g., automatically warning a user of the modification of a document that concerns him/her) without adding new functionalities.

In this subsection, we describe the principle of the retroaction mechanism used for the regulation of collaborative learning activity. When the observation station is started up, a base of rules (with XML format) dedicated to the regulation is loaded. The analyzer is thus ready to receive signals either collected by means of the collect Application Programming Interface (API) (e.g., the *save a page* action) or produced by the analyzer itself (e.g., the *beginning of a co-writing* action). The left-hand part of the rules that represent the specification of the use model is based on the use of operators of logic (AND, OR, NOT), of temporal relation (THEN), and of priority (brackets). Each time a signal is received, we verify whether it contributes to the condition of one or several rules. When the condition of a regulation rule is complete, the latter produces a sequence of actions which will be carried out on tools by means of the retroaction API (containing all the predefined retroaction methods).

13.3.1 Use Model to Support Regulation

We have seen previously that, from our point of view, the reading of traces corresponded to the expected interpretation, which was anticipated by the observer's involvement during the definition stage of the use model of the learning software tools. In their first version, the use models were represented by a set of rules that produced templates (signals or sequences), providing awareness-type information and thus allowing the users to operate explicitly on the tools (Courtin & Talbot, 2006). AQ5 We have proposed an evolution of the model by integrating retroaction execution in the rules in order to operate directly on the learning software tools.

We have considered the postulate that the concept of regulation covers all the mechanisms that ensure social control and that adapt the learning activities to the pedagogical objectives according to the context. Furthermore, as far as our experiments were concerned, the teacher had integrated the collaborative dimension (exchanges within pairs) into his/her pedagogical objectives. The objectives suppose a strong involvement of the learner in the group work mediated by the learning software tools. Below, we will attempt to highlight the effects of the increase in coupling between these tools on the participants' behavior. In other words, we wish to be able to appreciate the effects of the retroaction mechanisms and thus appreciate the increase in interactions on participants' involvement in the instrumented collaborative learning situation (ICLS).

13.4 Experiments

We have used an experimental approach to assess the CARTE observation station architecture and especially the regulation (R) mechanism. Furthermore, we agree with Mille, Caplat, and Philippon (2006) that user actions depend on reactions to the environment in a given activity context. The experiments we present in this chapter complete those described by Courtin and Talbot (2006), with a new type of use model featuring a retroaction mechanism. We restate briefly the context of experiments: the previous one without regulation and the last one with regulation. Both experiments were carried out during a practical class in an English course (foreign language) with students at the university. The students, in pairs, were supposed to define a set of situated English words by means of a specific collaborative text editor (multilingual dictionary). They communicated in pairs via a specific chat room to organize their work to their liking. The teacher was using the same production tool to check students' definitions and the same communication tool to answer students' questions. Furthermore, the teacher coordinated the groups by means of a group structuration tool and everyone visualizes the actions with the awareness tool. The CARTE observation station may provide participants/observers with these observation tools.

Our experiments consist in setting up a computer-based collaborative learning session (Talbot & Courtin, 2008). From a pedagogical point of view, a collaborative learning activity is one that requires mutual learner involvement and which implies educative intentions that aim to assist or to facilitate the learning process. The system is composed of the following: a production tool called Jibiki (a specific collaborative text editor to create multilingual dictionaries, created for a research project in linguistics, whose working depends on trace exploitation, Mangeot & Chalvin, 2006), a communication tool called CoffeeRoom (a chat room in which communication spaces are represented by tables), a group structuration tool, an activity monitoring tool (awareness), and the activity trace observation station CARTE.

As the activity scenario is very simple, it is not necessary to specify it with a scripting tool. The activity is carried out as follows:

Preliminary phase

- The teacher generates groups thanks to the group structuration tool.
- The teacher distributes the subjects with the words that have to be defined.

Work phase

- The students are free to organize their work in pairs with the CoffeeRoom.
- The students define the words with the Jibiki and submit them to the teacher.
- The teacher verifies the definitions, annotates them, or validates them directly.
- Sometimes the students ask the teacher for help.

The work is finished when the class is over or when all the definitions have been completed. The main objectives of our experiments are

- to inform about other participants' actions to which an interactant is subscribed (e.g., partner actions in a pair),
- to enrich the functionalities without modifying the software tools (e.g., sending an e-mail automatically after having recognized a specific action or asking for help),
- to set controls to ensure that the social rules necessary for the learning session to take place in the right way are respected (e.g., forbidding students to change tables in the CoffeeRoom), and
- to simplify the carrying out of the operations by automating certain tasks (e.g., automating the configuration of the learners' tools from the definition of the pairs by the teacher).

It should be noted that the above-mentioned aspects deal essentially with general group activity regulation (social rules) and not directly with the pedagogical activity itself, for example, to modify the way the collaborative learning session proceeds according to educative activities that are carried out by the students (nature and tempo). However, we think that some actions thus regulated improve interactions with the computing environment and can have an influence on learning conditions. It is what we attempt to show with the experiments results.

13.5 Description of the Software Learning Tools

We consider the postulate that an instrumented collaborative learning situation (ICLS) has to fit the specification of a groupware model, which takes into account all aspects of the collaborative work. We thus use as a base the previously presented augmented clover model, namely the classical model with a regulation space. The experiments that we have described have been carried out by means of learning software tools, which are initially designed for independent uses, that is to say which do not support intertool collaborative activities in real time. The adaptation of these tools, which are able to interact with each other and offer functionalities that are adapted to the collaborative learning situation, has been made possible thanks to the use of the traces they produce. Taking into account the objectives of our experiments, described above, we have thus selected learning software tools which cover the various spaces of the augmented clover model.

The production tool used is the Jibiki editor (see Fig. 13.4), which allows the creation of multilingual corpuses by means of a Web client. In this tool, each participant is registered with a casual role, that is to say a role that is associated with a set of possible actions (e.g., setting a new definition, validating a definition, and administrating the corpus). Because Jibiki is a Web client, we have chosen to use HTTP requests to retroact from the observation station and thus from the retroaction module. An API (application programming interface) has thus been created to process the retroaction requests in a generic way. A request is associated with a generic method which receives the following parameters:

- ActionType which represents the action to be triggered (the actions are located by means of constants).
- ParametersList which contains all the necessary parameters so that the request is sent.

For example, deconnection corresponds to an HTTP request with the following parameters:

- Action type: Jibiki deconnection
- Jibiki address: http://yeager.univ-savoie.fr
- Port used by the Jibiki: 8999
- Corresponding Java method corresponding to the deconnection: LoginUser.po
- Session identifier: ;jsession=KBK4LJQ9zTQa13uAasQTcf63
- Parameter: ?Logout=yes

In the context of our experiments (see Fig. 13.4), we have instrumented the editor in order to detect the following predefined set of actions:

- The (de)connection of a user (e.g., the student Albert),
- The creation of an entry (e.g., the word "gender"),

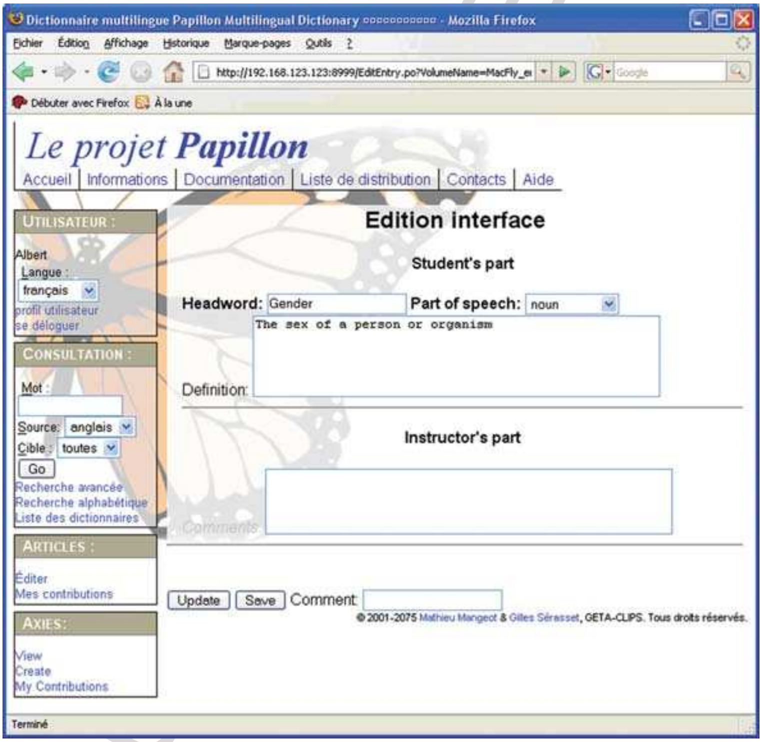

Fig. 13.4 Jibiki

- The editing of a definition (e.g., “the sex of a person or an organism”),
- The submission of a definition (to be checked by the teacher),
- The setting on standby of a definition to be corrected,
- The reviewing of a definition (to take into account the teacher’s annotations),
- The validation of a definition (when it is correct), and
- The integration of the definition in the corpus (when it is validated).

A pair management tool has been specifically implemented to match the experiments’ needs (see Fig. 13.5) by respecting a strict specification, according to which the functionalities for the collaboration are based on the use of traces. This specification guarantees the independence of this coordination tool’s implementation with regard to other tools, while allowing functional interdependence. In other words, the interactions with other tools are carried out solely by the retroaction rules defined in the analysis module of the observation station. In practice, after the teacher has set the pairs, the traces collected by the observation station trigger rules in which

This figure will be printed in b/w

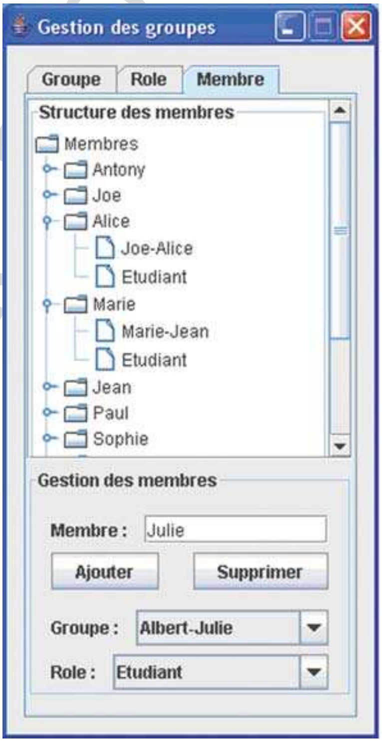

Fig. 13.5 Pair management tool[1]

[1]Translation:
“gestion des groupes/membres” means “group/member management”
“ajouter/supprimer” means “add/delete”

retroactions enable the creation of the corresponding tables in the CoffeeRoom. It should be noted that with this intermediate mechanism replacing the CoffeeRoom by another structured communication tool would not affect the implementation of the pair management tool.

As has been explained above, the specification of the learning software tools implies no modification of their code. Thus, an awareness tool (see Fig. 13.6) has been designed to gather information relative to the context of the activity in progress (e.g., the group structure, the name of a participant in difficulty, the action that has just been carried out, etc.). This information stems from "observeds" defined by the observer him/herself, with a level of interpretation very close to that of the latter.

The CoffeeRoom is a structured chat-room tool (the conversation spaces are represented by tables, presented in Fig. 13.7) used for communication among participants according to rules defined by the teacher at the beginning of the session. For example, during the preparation of the practical class, the teacher defines the pairs by

This figure will be printed in b/w

Fig. 13.6 Awareness tool[2]

---

[2]Translation: the most recent action is at the top of the list
"Conscience de l'activité (des groupes)" means "(group) awareness"
"Sans groupe" means "without a group"
"Valeur du paramètre" means "parameter value"
9:21: Albert has written the word "gender" without talking at his table
9:21: Albert has finished the word "gender"
9:9: Albert is trying to join another table

This figure will be printed in b/w

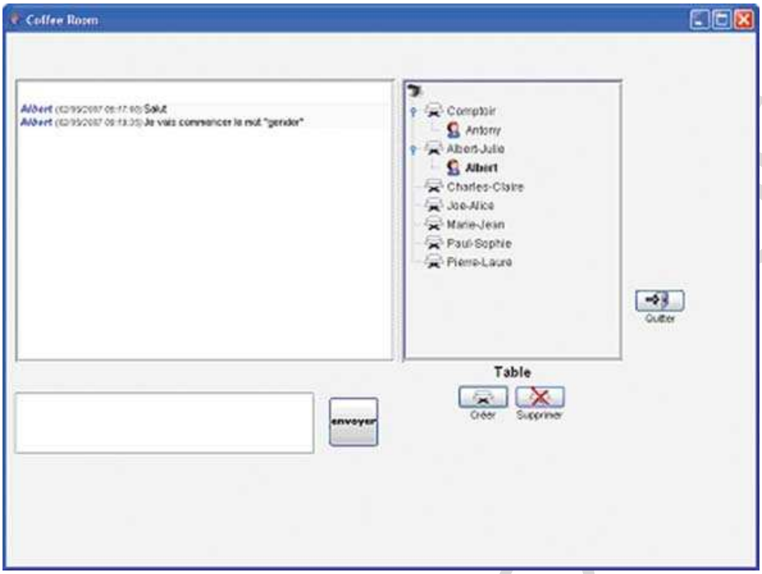

Fig. 13.7 CoffeeRoom[3]

means of the workgroup management tool previously presented, then enacts rules of social control by means of the observation station's rule editor. From then on, any action which breaks these social rules (e.g., Albert is attempting an unauthorized move to another table in the CoffeeRoom; see Fig. 13.6) will cause retroactions, such as sending warnings in the awareness tool (e.g., Antony, the teacher, is warned about participant Albert's attempted intrusion into another pair) or acting on the tool in which the transgression has been committed (e.g., the participant is returned to the table he/she came from). Therefore, this communication tool, when it is instrumented, has its functionalities enriched by the use of the activity traces of a group within a set of tools (taking into account part of the context).

In the context of our experiments, we have instrumented the CoffeeRoom to allow its automatic configuration from the session specification, which is carried out by the teacher (i.e., pair definition, setting-up of social rules, etc.). The retroactions thus identified are creating or deleting a table (from traces stemming from the configuration of the group management tool); connecting or disconnecting a user, adding a user to a table (according to the defined pairs); and sending a message to a table (e.g., to warn the partner automatically of the validation of an action). Following the example of the Jibiki, the CoffeeRoom has an API to carry out the aforementioned retroactions. In the same way, a generic method receives parameters such as the reference of the server, which centralizes the messages and on which

[3]Translation: "envoyer" means "to send"

there are already the methods corresponding to the various (retro)actions. This technique allows one to be both independent of the communication tool (if it contains separate discussion spaces) and to take into account the specificities of the chosen tool (here, the CoffeeRoom). The retroaction API, which is linked to the server of the CoffeeRoom, enables remote access via the Remote Method Invocation (RMI) protocol. The regulation module then has a Java interface listing all the possible retroactions which will be called up remotely at that time.

We have presented above an instrumentation of the various tools, which allows the analyzer to implement both types of regulation, namely human interventions from information which stems from the "observeds" (awareness) and retroactions triggered by one or several rules predefined by the observer him/herself. In both cases, the system is based on the exploitation of activity traces that are collected, then analyzed in the observation station by means of predefined use models.

13.6 Results

Our research objectives consist of transforming information that stems from computing traces in order for it to be relevant for a human observer and in testing our models' strength. In the first experiment, the software tools used in the collaborative work were independent of each other. However, we were defining a use model to recognize actions performed on one or several software tools. The observation system was able to provide participants with awareness information, but it was not able to trigger actions directly on the software tools. Thus, the latter were limited to their own functionalities. We briefly reiterate below certain limitations observed during the first experiment:

- Lack of information on activity within a pair (e.g., a student is not warned about the acceptance of a definition sent by his/her partner),
- Difficulty in identifying those who need help,
- Difficulty in having the rules respected (for example, pair separation) and in taking into account pedagogical context features,
- Presence of certain repetitive tasks specific to tools (e.g., in Jibiki, the definition validation by the teacher is always followed by a corpus introduction; two actions are necessary), and
- Lack of synchronization for actions performed within different tools (e.g., group management by teacher and corresponding configuration of the CoffeeRoom).

Therefore, it is a matter of analyzing the effects resulting from this new use model feature by means of the experiments presented below.

With regard to the last experiment, a computer-based study alone does not allow us to evaluate the regulation's effects on learning quality. On the other hand, this experiment has revealed the use model's capacity to specify the regulation actions of the activity in progress during an instrumented collaborative learning session. In order to present these results, we propose to describe the retroaction mechanism and present the related effects on the learning activity. To carry out retroaction on tools, we have modified the analyzer tool (see Courtin & Talbot, 2006) to add a list of actions to perform on the right-hand part of the rules. These actions are triggered after the sequence is recognized and added to the TBS. The use model allows us to define an abstraction level between actions to be carried out and the various tools' API (application programming interface). In concrete terms, a name is associated with every action as well as parameters for which the semantics are defined at the use model level. In our case, for example, the actions on Jibiki are sent to it in an HTTP request form, whereas those dedicated to the CoffeeRoom or to the awareness tool are carried out with remote methods invocation (RMI) in Java language.

From a functional point of view, the use model integrating the retroaction mechanism implies an incoming instrumentation for the tools concerned, which completes the outgoing instrumentation that ensures the connection between the trace sources and the first trace of the collect module of the CARTE system (see Figs. 13.1 and 13.3). From the point of view of the system architecture, there is no direct link between the tools; all the actions are triggered from the MTraces, which are produced by the TBS. A major advantage of this level of abstraction that is the use model is that it preserves the independence of the tools which are thus interchangeable. For example, the action that consists of putting a participant back onto his/her table in the CoffeeRoom is associated with the event *join* with two parameters: the name of the discussion space (here, the table) and the user's identifier. With another communication tool, whose structure features communication spaces bearing different names (e.g., rooms), the rule would then be the same, which would exempt us from modifying the use model at the observer level. We also note the relevance of going through this level of abstraction, which places the trace at the heart of the system and which then guarantees the persistence of the trace of all the stages that have led to a retroaction, the latter itself generating a new trace. For example, when a participant tries to change tables (event *join* with another table name), the trace produced by this attempt triggers a rule which executes an action on the CoffeeRoom (join his/her own table) and another one on the awareness tool (notification for both the teacher and him/herself), which will generate two new traces.

Retroaction increases coupling between the various software tools, which tends to change the way the tools are perceived by the interactants. For example, the fact of joining the awareness tool with Jibiki, which was initially designed to do collective editing with a very low coupling, allows us to warn the pair partner about definition validation and to visualize the right version. We have noticed the analyzer tool's efficiency, which enables it to be used online during the learning sessions. The rules we have specified in these experiments enable the following:

- To have feedback about the other participants' activities (e.g., a student is warned in the awareness tool when his/her partner has completed a definition not to do the same),
- To change the way the tools are perceived because of the increased interactivity,

- To automate certain tasks, in particular for the teacher (e.g., in Jibiki, with only one action, the definition validation by the teacher is automatically followed by a corpus introduction, what is an ergonomic improvement without modifying the software tool), and
- To reinforce social control at the group level (by restricting, for instance, the possibility of communication between students belonging to different groups).

The fact that retroactions are useful traces for recognizing other activities reinforces the idea that the sequences from which they originate have to be stored in the TBS. Likewise with the traces produced by these retroactions. This illustrates the trace-centered nature of our observation system.

As a result, the retroaction mechanism enriches the use models by adding virtually new functionalities without modifying the software tools themselves. From this, we have also observed new students' behavior, like discussion after a word has been validated in order to understand the right definition. In other words, these new virtual functionalities may increase the collaboration within student pairs.

We can notice that through these experiments, the enrichment of interactivity, on the basis of information emanating from the context (activity in progress, difficulties encountered, etc.), gives rise to new behaviors. For these experiments, the collaboration between participants within a pair was part of the pedagogical objectives defined by the teacher. It remains to be seen whether these new behaviors meet the expectations of the latter. The CARTE system already allows us to perceive very promising answers to this question which will be the basis of new experiments in the context of the setting up of a test-bed project (customization of the technology-enhanced learning systems [TELS] of the research cluster ISLE of the Rhône-Alpes Region, 2008–2011), dedicated to researchers in computer science, social sciences, didactics, etc. The observation station will then be able to interact with this activity's trace-centered benchmarking platform.

The expectations of the observers (e.g., experts, teachers, and learners) are the production of indicators, that is to say traces with a level of abstraction close to their interpretation. During the preparation of the learning session, the teacher defined explicit rules that specify the use model for the analyzer. For example, when a user writes "help" on a table, a message is sent directly to the awareness tool for the teacher's attention. This trace is then calculated from the event "send a message on a table" containing the single text message, *help*. For the teacher, it is an indicator of the type "awareness," confirming any difficulty encountered by a learner. Another example of an indicator, namely the warning message about an attempt to intrude on another table, may be interpreted by the teacher as being a test of navigation in the CoffeeRoom, or the wish to communicate with another pair, or the search for solutions elsewhere in spite of the instructions, etc. The goal of this regulation (the fact of thwarting the attempt automatically) is to introduce social control into the session to guarantee the respect of the pedagogical objectives.

With the experiments, it appears that the CARTE system seems to support basic regulation in ICLS. However, we have noticed some limits in the current version of the prototype to express specific actions of regulation such as the following: to warn the teacher when a pair of students has completed *n* definitions in a given period of time. Indeed, iteration and time dimension are not already supported by the rules editor. In fact, we get around the iteration drawback by repeating *n* times the corresponding MTrace (signal or sequence). These technical drawbacks may be solved by adding a new analyzer and then by applying a new use model, working on previously generated MTraces (i.e., with the first analyzer). We plan to carry out this work for the next version of the prototype.

13.7 Conclusion and Perspectives

We have described an improvement of the trace-based system (TBS) to allow and take into account regulation in collaborative activities in a learning context. A prototype observation station called CARTE has been presented, and the results of the experiments carried out with students at the university have enabled us to test our models' strength and have highlighted the added value of such a trace-centered system with a retroaction mechanism.

We are currently working on how to operationalize our observation system in a project about multilingual dictionary asynchronous co-construction (Mangeot & Chalvin, 2006). An important first stage of this work consists in defining relevant indicators for this new type of analysis.

AQ9 AQ10 We are also thinking of enhancing pedagogical scenarios (Martel, Vignollet, Ferraris, David, & Lejeune, 2006; Talbot & Pernelle, 2003) with our activity trace models, particularly by exploiting the analyzer retroaction mechanism.

Generally speaking, we plan to work in a full-service groupware perspective. This implies orienting our work toward generic observation services for collection, analysis, awareness, retroaction, etc.

Given that the CARTE system strictly respects the specification of the theoretical model of TBS, this observation station has been selected for the study of an activity trace management platform in technology-enhanced learning systems, in order to create corpuses of reference that are open to the international community of researchers.[4] The goal of such corpuses is to be found at the intersection of numerous problems centred on a research object, called a trace, which derives from various sources and is able to take different forms, while respecting a specification to allow interoperability with other traces and to take into account all the specificities inherent in its use (ownership, ethics, classification, etc.). This benchmarking platform is dedicated to research in computer science, social sciences, pedagogy, didactics, etc.

[4]This work in progress is part of the research cluster ISLE, project "customization of the technology-enhanced learning systems (TELS)", supported by the "Region Rhone-Alpes" in France.

References

Carron, T., Marty, J. C., Heraud, J. M., & France, L. (2005). *Preparing an observed pedagogical experiment* (pp. 526–531). International conference on cognition and exploratory learning in digital age (CELDA'05), IADIS, Porto, Portugal.

Chabert, G., Marty, J. C., Caron, B., Vignollet, L., & Ferraris, C. (2005). The electronic schoolbag: A CSCW workspace. *Review AI & Society: The International Journal of Human-Centred Systems and Machine Intelligence, Springer*, *20*, 403–419.

Courtin, C., & Talbot, S. (2006). *Trace analysis in instrumented collaborative learning environments* (pp. 1036–1038). 6th IEEE International Conference on Advanced Learning Technologies (ICALT'06), Kerkrade, Netherlands.

Courtin, C., & Talbot, S. (2007). *Une station d'observation pour des situations d'apprentissage collaboratif instrumenté* (pp. 371–376). Environnement Informatique pour l'Apprentissage Humain (EIAH'07), ATIEF, Lausanne, Switzerland.

Courtin, C. (2008). CARTE: an observation station to regulate activity in a learning context. In Kinshuk, D. G. Sampson, J. M. Spector, P. Isaías, & D. Ifenthaler (Eds.), *Proceedings of the IADIS international conference on cognition and exploratory learning in the digital age* (pp. 191–197). Freiburg, Germany: IADIS.

Ellis, C., & Wainer, J. (1994). *A conceptual model of groupware*. Computer supported cooperative work (CSCW'94), ACM conference, Chapel Hill, NC, pp. 79–88.

Mangeot, M., & Chalvin, A. (2006). *Dictionary building with the Jibiki platform: The GDEF case*. Proceedings of LREC'06, Genoa, Italy, pp. 1666–1669.

Martel, C., Ferraris, C., Caron, B., Carron, T., Chabert, G., Courtin, C., et al. (2004). *A model for CSCL allowing tailorability: Implementation in the "electronic schoolbag" groupware: Vol. 3198/2004. Groupware: Design, implementation and use* (pp. 322–337). Berlin/Heidelberg: Springer.

Martel, C., Vignollet, L., Ferraris, C., David, J. P., & Lejeune, A. (2006). *Modeling collaborative learning activities on e-learning platforms* (pp. 707–709). 6th IEEE international conference on advanced learning technologies (ICALT'06), Kerkrade, Netherlands.

Mille, A., Caplat, G., & Philippon, M. (2006). Faciliter les activités des utilisateurs d'environnements informatiques: qui, quand, comment? *Intellectica, 2*(44), 121–143.

Rabardel, P. (1995). *Les hommes et les technologies: Approche cognitive des instruments contemporains*. Paris: Colin.

Settouti, L., Prié, Y., Mille, A., & Marty, J. C. (2006). *Systèmes à base de trace pour l'apprentissage humain* (p. 8). Actes du colloque de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise, Conference TICE 2006, Toulouse, France.

Talbot, S., & Courtin, C. (2008). *Trace analysis in instrumented learning groupware: An experiment in a practical class at the university* (pp. 418–422). Proceedings of the seventh IASTED international conference on web-based education. Innsbruck, Austria.

Talbot, S., & Pernelle, P. (2003). Helping in collaborative activity regulation: modeling regulation scenarii. In T. Baudel (Ed.), *15th French-speaking conference on human–computer interaction (IHM'03): Vol. 51. IHM 2003* (pp. 158–165). Caen, France: ACM Press.

Chapter 13

Q. No.	Query
AQ1	"Carron, 2005" has been changed to "Carron, Marty, Heraud, France, 2005" as per reference list. Please confirm.
AQ2	"Settouti, 2006" has been changed to "Settouti, Prié, Mille, Marty, 2006" as per reference list. Please confirm.
AQ3	"Ellis, 1994" has been changed to "Ellis, Wainer, 1994" as per reference list. Please confirm.
AQ4	"Chabert, 2005" has been changed to "Chabert, Marty, Caron, Vignollet, Ferraris, 2005" as per reference list. Please confirm.
AQ5	"Courtin, 2006" has been changed to "Courtin, Talbot, 2006" as per reference list. Please confirm.
AQ6	"Mille, 2006" has been changed to "Mille, Caplat, Philippon, 2006" as per reference list. Please confirm.
AQ7	"Talbot, 2008" has been changed to "Talbot, Courtin, 2008" as per reference list. Please confirm.
AQ8	"Mangeot, 2006" has been changed to "Mangeot, Chalvin, 2006" as per reference list. Please confirm.
AQ9	"Martel, 2006" has been changed to "Martel, Vignollet, Ferraris, David, Lejeune, 2006" as per reference list. Please confirm.
AQ10	"Talbot, 2003" has been changed to "Talbot, Pernelle, 2003" as per reference list. Please confirm.
AQ11	Please provide forename for the author "Kinshuk."