

AN ARCHITECTURE TO RECORD TRACES IN INSTRUMENTED COLLABORATIVE LEARNING ENVIRONMENTS

Christophe Courtin, Stéphane Talbot

▶ **To cite this version:**

Christophe Courtin, Stéphane Talbot. AN ARCHITECTURE TO RECORD TRACES IN INSTRUMENTED COLLABORATIVE LEARNING ENVIRONMENTS. IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA), Dec 2005, Porto, Portugal. hal-02885190

HAL Id: hal-02885190
https://hal.science/hal-02885190

Submitted on 30 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AN ARCHITECTURE TO RECORD TRACES IN INSTRUMENTED COLLABORATIVE LEARNING ENVIRONMENTS

Christophe Courtin
Laboratory Systèmes Communicants – University of Savoie
73376 Le Bourget-du-Lac Cedex, France
Christophe.Courtin@univ-savoie.fr

Stéphane Talbot
Laboratory Systèmes Communicants – University of Savoie
73376 Le Bourget-du-Lac Cedex, France
Stephane.Talbot@univ-savoie.fr

ABSTRACT

Supporting observation activity in a digital workspace environment is an idea that holds promise for improving groupware activity. In this paper, we focus on non-constraint equipped collective learning situations, that is to say without a given structure to the working process. We present a meta-model and an architecture of an observation station, to collect activity traces and to analyse them at different levels of abstraction. System traces, such as logs, are generally inadequate to identify pedagogical activities. Therefore, we have been working on a more accurate technique for observation instrumentation. Our initial experience has shown that this technique can significantly improve event trace processing for final user-oriented analysis.

KEYWORDS

trace, observation, instrumentation, awareness, collaborative learning.

1. INTRODUCTION

Over the last few years, there has been a new orientation of e-learning devices. Initially, the most common models were based on making resources available in an autonomous way for the learner. Today, very few e-learning technologies take advantage of the collaborative aspect in the learning process, because it enables actors (learners et teachers) neither to observe group activity and to act consequently, nor to evaluate the benefits for the learning process. Progressively, these rather static devices have become inadequate for collective learning processes, and to reach the declared pedagogical goals. When equipped collective learning situations (ECLS) are set up, the teacher can plan several sequential activities with a workflow system for instance, but she/he can also wish to observe actively a non-constraint ECLS: the sequences of actions, the interactions, the exchanges, the resources used (contents, materials, tools, software), and so on. In the latter, the teacher can intervene during the process to help, to correct, or to stimulate the learner if necessary. When the session is finished, she/he may analyse a posteriori the activity, the usability of the available tools (statistics), the nature of the tasks performed (production, communication, etc.). From this, she/he carries out the activity process assessment, to adjust its organization, and thus improve it.

When equipped collective learning situations (ECLS) are set up, Pernin (2004) highlights the necessity for the automation of observation and regulation activities traditionally performed by human beings. Both of these activities are essential for pedagogical monitoring to assist learners, and to tailor the environment to the individual.

In this study, all the tools used (production, communication, coordination) are integrated into a digital workspace environment (DWE). This means they are interoperable with the environment: roles and rights applying in the arena, DWE functionality use in tools (e.g. add an image or a hypertext link).

In this paper, we briefly set out the objectives we seek to reach. Then, we present the problem of collection in digital workspace environments, and we highlight their shortcomings in terms of traceability with a simple use example. We attempt to counter this weakness by means of an instrumentation technique applied to environment tools and functionalities. We then present a meta-model and an architecture for an observation station inside a DWE. We adapt the aforementioned example to an equipped collective learning situation. Finally, we measure the potential of such a system to satisfy our given objectives. This study leads us to a conclusion, and we plan to place our work in a more general perspective.

2. OBJECTIVES

There are numerous and complementary objectives of our work about observation.

The first objective consists in providing participants with a view of group activity, which is called group awareness. In the context of our study, we want particularly to provide the actors (learners and teachers) with the possibility of following an equipped collective learning situation (ECLS) in progress, which is considered as part of workspace awareness (Gutwin 2002). From the pedagogical point of view, we must provide teachers with the means of helping learners in real time, and of evaluating the recorded learning process.

As a consequence of the latter point, we define the second objective which concerns reporting on the usability and adequacy of the tools with the learning activity. The analyst will be able to improve the ECLS, or even identify new tool uses.

The long term objective consists in measuring the impact of instrumentation on a collective learning situation.

3. THE COLLECTION PROBLEM

One of the software artefact features which support collective learning activities, is the capacity to produce activity traces for the different actors. On the other hand, this information is rarely relevant or even insufficient to enable activity interpretation, and analysis tools are generally absent. Most of the time, these difficulties lead us to make the observations in an empirical manner, by applying a limited set of generally manual techniques: this implies that the observation corpus and resulting analyses are quite poor.

Observation from raw traces of server logs enables analysis of tools and services use inside DWE (Chabert 2005). On the other hand, it does not enable activity interpretation inside tools used in the arena, which is defined as change of granularity. Moreover, it is necessary to complete the abovementioned traces with those coming from the use of non-centralized tools on the server, such as Instant Messenger for communicating (Heraud 2005). Lastly, possible actions carried out outside the software tool, as for direct human-human interactions, must be included in traces. A general approach of observation consists in gathering all the possible flows of events generated by activity. A quantitative collection of traces highlights the heterogeneity of event sources, which may be completed by qualitative data (interviews, etc.). The problems of this bottom-up approach are both the huge information volume (Smith 2000), and the high level of interpretation when activity is multi-user and multi-tool.

We are going to describe an example of collaborative activity to illustrate the last case, which features a communication tool and a production tool. This activity is divided into several stages: (1) discussion in pairs (x_1, x_2) and (y_1, y_2) of two subjects, (2) writing each subject as single members (x_1) and (y_1), (3) at the same time, discussing each subject successively one by one (x_2) and (y_2), (4) synthesizing each subject in the initial pairs (x_1, x_2) and (y_1, y_2), (5) changing the single members (x_2) and (y_2) to finish the writing of each subject. In this example, actions (1) (3) and (4) are performed with the communication tool, and actions (2) and (5) with the production tool. In the experiment, the production tool is a wiki (an asynchronous cooperative editor for web pages) and the communication tool is a coffee-room (a chat room where communication spaces are represented by tables).

(1) discussion in pairs (x_1, x_2) and (y_1, y_2) of one of the two subjects = a member of each pair (x_1) and (y_1) creates a thematic table + the other members (x_2) and (y_2) join their respective table + each pair (x_1, x_2) and (y_1, y_2) exchanges messages at its table

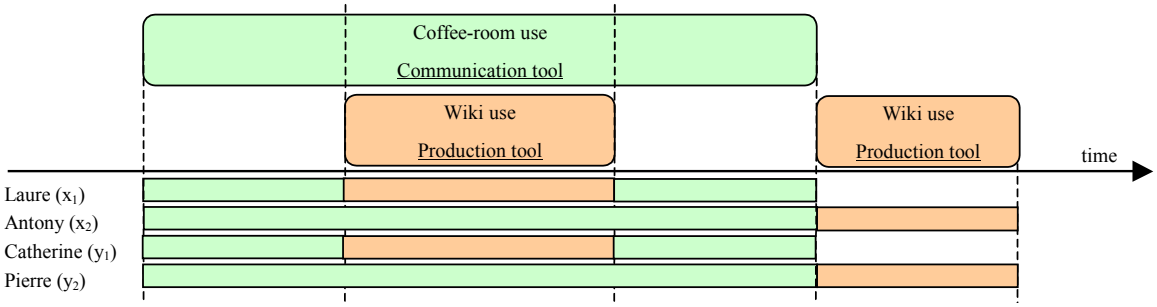
(2) writing on each page as single members (x_1) and (y_1) = each single member (x_1) and (y_1) joins one of the two pages + edits it + saves it

(3) discussing one of the two subjects as single members (x_2) and (y_2) = a member (x_2) of one pair and a member (y_2) of the other pair discuss both subjects successively

(4) synthesizing each subject in the initial pairs (x_1, x_2) and (y_1, y_2) = all the members (x_1, x_2) and (y_1, y_2) join their respective table

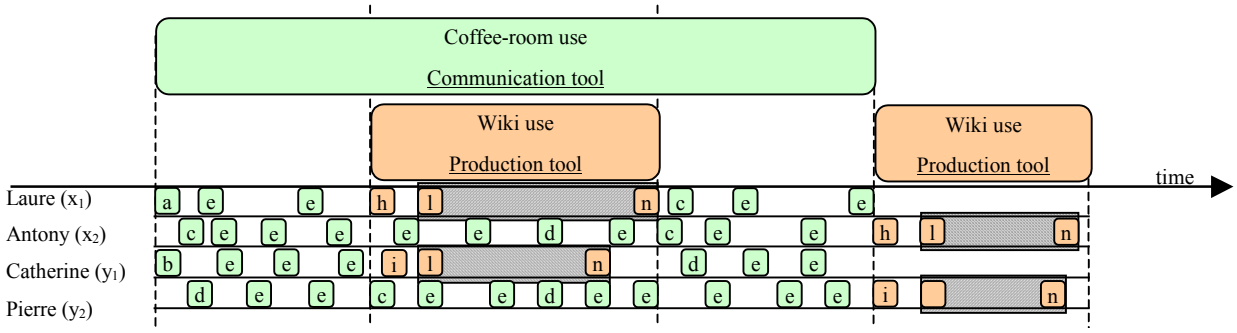
(5) changing the single members (x_2) and (y_2) to finish the writing of each subject = every new single member (x_2) and (y_2) joins their respective table + edits it + saves it

Figure 1. Observation with logs



As shown in Figure 1, we only draw up statistics on collaborative tool use with the logs, but we obtain nothing about actions carried out inside tools. For instance, we cannot determine that the first pair consists of Laure and Antony, and that the second one consists of Catherine and Pierre. We can identify single member location in the DWE, but not in the tools themselves.

Figure 2. Observation with instrumentation



Therefore, it is necessary to identify all the actions carried out inside tools (cf. Figure 2), and to highlight correlation between them to achieve identification of a higher level activity.

Action (1) division:

(a) creating a table for the first subject and joining it (inseparable actions)
(b) creating a table for the second subject and joining it (inseparable actions)
(c) joining the table of the first subject
(d) joining the table of the second subject
(e) sending a message

Actions (2) and (5) division:

(h) joining the page of the first subject
(i) joining the page of the second subject
(l) editing the page
(n) saving the page

Actions (3) and (4) division:

(c) joining the table of the first subject
(d) joining the table of the second subject

(e) sending a message

The event sequence (h,l,n) means writing on the first page, and (i,l,n) writing on the second one, in the same way that (c,e*) means discussion on the first table, and (d,e*) on the second one. Tool equipment enables one to identify more precisely achieved operations inside tools, and to interpret activity at a higher abstraction level.

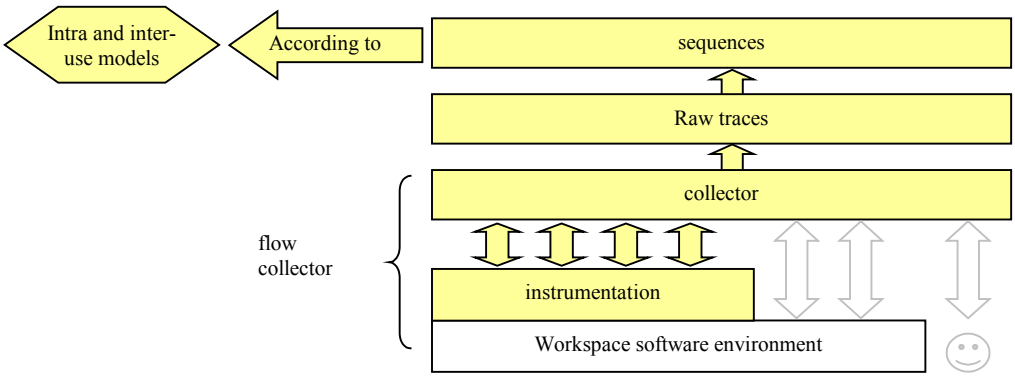
We use a top-down approach which implies the use of several collective tools. We strive to reduce information volume generated by the system by explaining possible actions in the arena, and in particular inside the latter's tools. This implies the recognition of sequences of events corresponding to the said actions thanks to intra-tool collaborative use models, which are extrapolated to define an inter-tool collaborative use model, explained later.

4. TRACE MODEL

4.1. Meta-model

The observation problem may be considered from different angles according to the fields concerned (computer science, sociology, educative sciences, etc.). This leads us to consider several abstraction levels (raw traces, actions, activities). Thus, in our experiment, we present a meta-model (cf. Figure 3) which comprises other existing models, such as MUSETTE (Champin 2002) from the LIRIS laboratory. Traces from several abstraction levels may be produced to make analyses easier with different points of view and granularities. This approach enables one to emphasize variable coupling of collaborative activities, which implies several shared artefacts in the DWE. This meta-model allows one to define relationships between signals (generated events) and tool use models. In the instrumentation activity, it is possible to distinguish clearly between the use model development and the observed tools' equipment. This ensures the creation of real banks of explained activity traces.

Figure 3. Meta-model for observation



The meta-model emphasizes the various event sources: the tools, the system, or external artefacts such as a human observer. Within this study, where we have shown that the logs' granularity does not enable us to achieve our objectives, we will focus on tool and DWE functionality traces.

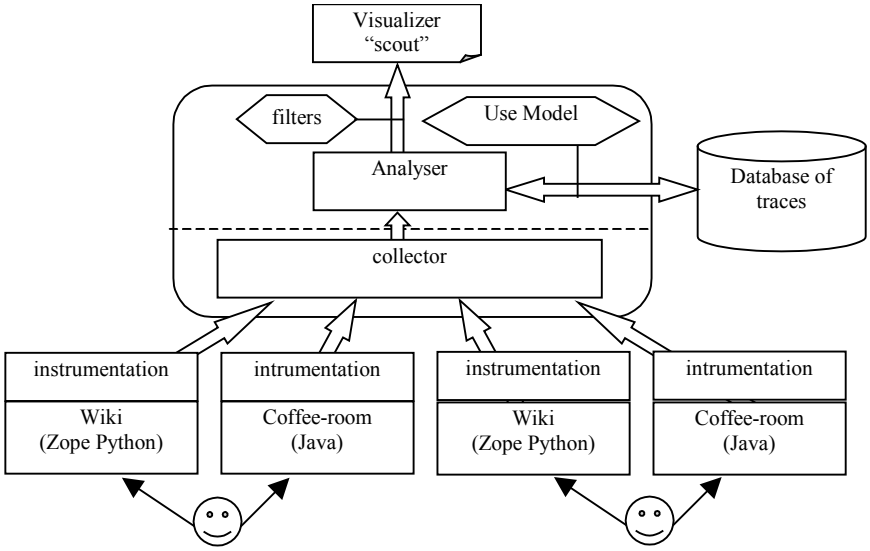
4.2. Context of experience

In our experience, we focus on how small groups of participants perform remote collaborative tasks in a non-constrained way, that is to say without determining different work stages beforehand. In the equipped collaborative learning situation concerned, we focus on a remote co-writing exercise by four students on a given subject, with a given deadline. The students do their work by means of a communication tool called coffee-room, and a writing tool, called wiki. In real time, a teacher observes at a distance the work in progress, and may intervene if necessary whenever she/he wants: to stimulate or to help the students. At the

end of the session, the teacher can analyse the resulting writing, the students' work management, and DWE tool use a posteriori. In order to carry out this experiment, we suggest setting up a generic observation station in a DWE with an awareness tool (cf. Figure 4).

The generic observation station includes a central kernel, called the trace manager, which is divided into two main parts: the collector and the analyser. The inputs of this central kernel are events from equipped tools and from digital workspace environment functionalities, and there are several outputs depending on the chosen abstraction level. According to our goal, which is to support part of shared workspace awareness, we present a final user-oriented output called the "scout".

Figure 4. Architecture of an observation station



Intra-tool use model enables one to annotate a tool action (e.g. to edit a text in a wiki tool, or to send a message in a coffee-room tool). From such annotations we can deduce higher abstraction level activities, such as collaborative editing or discussion on a specific topic. We will use a formalism, presented in Table 1, to describe intra or inter-tool use traces. An elementary annotation (signal) is an association of a tool name, a date, a description, an event, an user and parameters: [tool, date, description, event, user, param1, param2,…]. In the examples, we will denote variables with a question mark. Thus, in the example described in the previous parts, [wiki, $?t_0$, description, edit-page, $?x_1$, ?page1], [wiki, $?t_1$, description, save-page, $?x_1$, ?page1] and [wiki, $?t_2$, description, edit-page, $?y_1$, ?page1], [wiki, $?t_3$, description, save-page, $?y_1$, ?page1] signals where $t_0<t_1<t_2<t_3$, imply collaborative editing on the same text fragment (or page), and [coffee-room, $?t_0+\Delta t$, description, send-msg, $?x_1$, ?table1], [coffee-room, $?t_0+\Delta t'$, description, send-msg, $?y_1$, ?table1] signals describe a discussion on a specific topic (because it occurs at the same table).

Extrapolation of intra-tool models to inter-tool ones enables one to highlight links between activities carried out with several tools. In the previous example, [[wiki, $?t_0$, description, edit-page, $?x_1$, ?page1], [wiki, $?t_1$, description, save-page, $?x_1$, ?page1], [wiki, $?t_2$, description, edit-page, $?y_1$, ?page1], [wiki, $?t_3$, description, save-page, $?y_1$, ?page1], [coffee-room, $?t_0+\Delta t$, description, send-msg, $?x_1$, ?table1], [coffee-room, $?t_0+\Delta t'$, description, send-msg, $?y_1$, ?table1]] trace sequences suppose that both participants x_1 and y_1 communicate at the same table "table1" and are writing the same fragment "page1" as a result. As far as our study is concerned, the teacher may consider this activity as collaborative writing with discussion. She/he will be able to join the table possibly to participate in the discussion.

5. IMPLEMENTATION

The goal of the implementation consists in testing how realistic our approach is. In particular it is important for us to verify how tractable it is to define such an architecture as the one described in section 4.2. with different levels and standardized API between each of them.

We are still in the early stages of development. Currently we have concentrated

- essentially on the implementation of a generic collector and on the instrumentation of tools,
- a little on the analyser which is still very basic and inadequately standardized,
- and not at all on the visualizer which at this time can hardly display the signals received and the sequences identified.

5.1. Collection

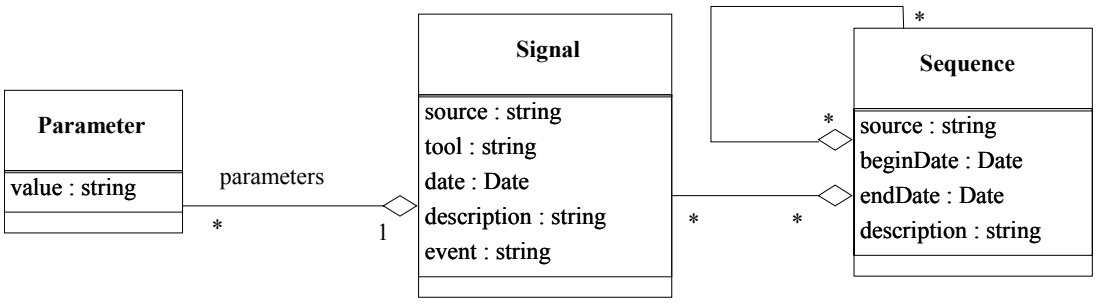
Collecting information on what happens in tools is not an easy task. Even when tools can generate logs, these ones have different formats. Moreover the data of interest to us will also be very different in nature and even in form. So for our model to be usable, it has to very generic but also allow us to access the different parts of a trace.

5.1.1. Model

With our model, we can represent two kinds of objects as activity traces:

- signals which correspond to time pinpoints and elementary elements;
- sequences which split into a chronological succession of signals or sub-sequences. Obviously a sequence also has some duration.

Figure 5. UML model



We have deliberately decided to use a simple model. For the signals we have attributes only to define what we were sure to find in all signals:

- a source (the people or the tool which has generated the signal)
- a tool (the tool or the instance tool in which the event has taken place)
- a date (when the event happened)
- an event description (the list of possible events will of course depend on the tools we use)
- a list of parameters (which contains the variable parts of signals). In this list we should find everything that is needed to understand what has happened. For example who and what is involved in the event. Obviously, this part also will change with events and tools.

Normally we will obtain signals from the tools either directly, or indirectly by mean of logs. The sequences are more complex than signals. So we will rather obtain them from the analyser part of the observation station (that is its job). However nothing forbids one tool to also generate sequences. So it is possible that a "smart" tool generates signals and, moreover, sequences.

5.1.2. The collector

The collector is in charge of gathering all the traces from the different sources available, a priori the logs or the tools. It is also responsible for stocking them in a persistent storage area, in a common format. In our implantation the format is the one described above (cf. Figure 5) and we are using a relational database for the persistent storage.

As stated the collector should be able to obtain data from logs or directly from tools. With logs we should use filters to interpret the raw logs and render them as signals. Because there is in our University another group which is currently working on this subject (the log usage problem), we will not cover this part, although their work is fully compatible with ours.

In the second instance, the tools and the collectors have to come to an agreement with each other on how they will communicate. To achieve this purpose, we have defined a given set of API. Using them, first we are

able to establish a connection between one tool and the collector, then the tool can send the collector the signals which describe what is happening.

Actually the observation station is incorporated into a Web portal built using Zope/Python (Latteier 2001). We also have some tools written in java and want to have the possibility of using the observation station and monitoring them. So XML-RPC (Winer 1999) has been chosen as the reference implementation for our API. Indeed many programming languages, including java, Zope and python can speak transparently XML-RPC. So XML-RPC is also an element which contributes to the genericity of the implementation.

5.1.3. The tools

As suggested above, two different tools have been instrumented and can be used with the observation station. The first one is a Zope/python tool (a wiki, an asynchronous cooperative editor for web pages) well integrated into the Zope Portal we use. The second one is a Coffee room implemented using the Java programming language. This choice was guided by two main targets. First we wanted to demonstrate that we could instrument tools whether they are neatly integrated into our Zope portal or not. Secondly we had planned to use these tools in different experiments, but until the instrumentation we lacked usable information on what happened inside during these experiments.

Of course we generally have to access and modify the source code of the tools if we want to instrument them. In our example it is not a problem, because the wiki is OpenSource and the Coffee room has been implemented by some of our students.

We have used two different approaches for the instrumentation. With the wiki we have slightly modified the HCI (Human-Computer-Interface). So now when the user clicks on buttons or links, the extra code we have inserted in the HCI sends signals to our collector.

For the wiki five signals are actually generated and transmitted to the collector:

- when a page is created (event create-page)
- when a page is deleted (event drop-page),
- when a page is visualized (event load-page),
- when the edit mode is activated on a page (event edit-page),
- when a modified page is saved (event save-page).

In the signal two parameters are also associated with all these events: the user who performed the action and the name of the page which is either created, deleted, visualized, edited or saved.

The HCI of the Coffee room is much more sophisticated than the wiki one. So inserting code at the HCI level was not very easy to manage. On the other hand, the Coffee room implementation implied one centralised server and one client on the computer of each user. The server receives information from each client and dispatches them to the others. So we just had to instrument the Coffee room server to get what we wanted.

Table 1. Signals received by the collector during the 2nd phase of the example experiment

source	tool	date	Event	param_1	param_2	param_3
Wiki	wiki	12005-06-15 15:07:12	load-page	Laure	page-subject1	
Coffee-Room	coffee-room	12005-06-15 15:07:25	join-table	Pierre	subject1	
Wiki	wiki	12005-06-15 15:07:35	load-page	Catherine	page-subject2	
Coffee-Room	coffee-room	12005-06-15 15:07:56	send-mesg	Antony	subject1	Aren't you supposed to work with Cathy ?
Coffee-Room	coffee-room	12005-06-15 15:08:20	send-mesg	Pierre	subject1	Hi Antony, join us for 5' at the other table…
Wiki	wiki	12005-06-15 15:08:36	edit-page	Catherine	page-subject2	
Wiki	wiki	12005-06-15 15:08:41	edit-page	Laure	page-subject1	
Coffee-Room	coffee-room	12005-06-15 15:10:33	send-mesg	Antony	subject1	OK, give me just 1'
Coffee-Room	coffee-room	12005-06-15 15:11:02	send-mesg	Pierre	subject1	OK, I'm going back

In its simplest version the Coffee room can be thought of as a chat in which the communication area would be seen as tables. A user can go inside or outside the Coffee room. When s/he is inside s/he is able to create and join tables, go from one table to another or send messages. When a message is sent it can only be seen by the users who are at the sender's table.

After our instrumentation the Coffee room server now sends a signal for each of the following actions:

- on connection (event connect),
- on exit (event disconnect),

- when somebody creates a table (and joins it) (event create-table),
- when somebody goes to a table (event join-table),
- when somebody sends a message (send-msg).

The signals related to the connection and the disconnection have just one parameter, the name of the user who is connecting or disconnecting. When somebody sends a message then the signal has three parameters: the user who sends the message, the table at which s/he stands and the message. The other signals have only two parameters: a user and a table.

The table above (see Table 1) gives some idea of the kinds of signals which could be collected in the case of the experiment described in section 4.2.

5.2. Analysis and display

With our approach, it is now possible to change the granularity of use traces. The given elements thus enable us to increase the abstraction level of activity analysis. Then, analysis consists in rebuilding activity traces from these elements, and from a representation of activity by means of meaningful sequences of signals (or sub-sequences), described in intra and inter use models. Although there already exist such models (Champin 2002), we have chosen a simple and explicit representation with patterns. A sequence model is made up of a set of partially instantiated patterns (e.g. [wiki, $?t_0$, description, edit_page, $?x_1$, ?page1]). These patterns correspond either to signals or to sub-sequences. Insofar as we wanted to focus on the collector, this choice simplifies recognition mechanisms.

For now, the display, called "scout", is very simple and basic: it just shows recognized sequences. Regarding our equipped collective learning situation, the teacher can now recognize action sequences presented in part 3, and s/he can interpret them according to the work given.

6. CONCLUSION AND PERSPECTIVES

The work presented is a first step towards the long term objectives we have set ourselves. The non-constraint context of our study enables us to exploit less structured learning situations. The simplicity of the instrumentation technique and of the observation station ensure great interoperability between the tools themselves, and the digital workspace environment.

Today in our laboratory, we are working on the convergence of oriented-pedagogical scenario approaches (Heraud 2005) and artefact use of the DWE. Therefore, we will be able to combine learning objectives with effective and predicted collaborative use of tools. In the near future, qualitative methods (semi-directed interviews) will join these quantitative methods to also observe new uses (Chabert 2005).

REFERENCES

Chabert, G. et al., 2005, L'observation des usages dans un contexte éducatif : pratiques pédagogiques et nouvelles formes de communication éducatives émergentes ?. *ISD.* Tunis, Tunisie.

Champin, P.-A. and Prié, Y., 2004. MUSETTE: a Framework for Knowledge Capture from Experience. *Short paper in EGC*. Clermont-Ferrand, France.

Gutwin, C. and Greenberg, S., 2002. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *CSCW.* New-Orleans, USA.

Heraud, J.-M. et al., 2005, Helping the Interpretation of Web Logs: Application to Learning Scenario Improvement. *AIED.* Amsterdam, The Netherlands.

Latteier, A. and Pelletier, M., 2001. *The Zope Book.* New Riders Publishing, Indianapolis , Indiana, USA.

Pernin, J.-P. and Lejeune, A., 2004. Modèle pour la réutilisation de scénarios d'apprentissage. *Colloque TICE Méditérannée*. Nice, France, p48.

Smith, R. and Korel, B., 2000. Slicing Event Traces of Large Software Systems. *In M. Ducasse (ed), proceedings of the Fourth International Workshop on Automated Debugging (AADEBUG)*. Munich, Germany.

Winer, D., 1999. *XML-RPC Specification*[online]. XML-RPC.com. Available at: http://www.xmlrpc.com/spec [Accessed 20 July 2005].