Monitoring Collaboration in Flexible and Personal Learning Environments

María Jesús Rodríguez-Triana, Alejandra Martínez-Monés, and Juan Ignacio Asensio-Pérez

GSIC-EMIC, University of Valladolid, Spain
{chus@gsic, amartine@infor, juaase@tel}.uva.es
http://www.gsic.uva.es

Abstract. Both virtual and personal learning environments (VLEs and PLEs) are technologies widely used with educational purposes. Moreover, there also exists a pedagogical trend towards the development of high-level competences, such as the capacity to work collaboratively in a group. These two trends come together within the field of computer supported collaborative learning (CSCL). The evaluation of CSCL situations requires to take into account, among other things, the process of collaboration. In response to this need, a number of collaboration analysis tools have been developed. However, the integration of these tools in the decentralized scenarios typically found in PLEs is a complex matter: the information retrieved and the access to it are heterogeneous; and the lack of data sharing standards between the learning software and the analysis tools reduces the probability of compatibility. The present work aims to delve into the problem of the integration of collaboration analysis tools and learning software. A solution is proposed to support monitoring in an architecture that already integrates third-party external tools in PLEs and VLEs named GLUE!. The proposal is illustrated by means of an example based on a real CSCL experience that took place in a course at our University.

Keywords: CSCL, collaboration analysis, VLEs, PLEs, GLUE!

1 Introduction

Currently, there exists an increasing trend in education towards a more learner-centered and decentralized learning. There are initiatives aiming towards the integration of third-party external tools, mainly Web 2.0 tools, in personal and virtual learning environments (VLEs and PLEs) [2], making them more flexible and configurable. Hereafter, we will refer to these PLEs and VLEs as flexible and personal learning environments (F&PLE). Computer-Supported Collaborative Learning (CSCL) scenarios are an example within TEL where these technologies are employed in order to improve learning and teaching [7] [12].

When evaluating a CSCL situation, the analysis of the collaboration among participants becomes a central issue [11]. This has been a strong trend in CSCL in the last years, mostly focused on detailed interaction analysis methods, which provide highly detailed accounts of the collaboration, appropriate for the research purposes [6]. However, teachers need a more abstract idea of collaboration, easier to interpret so that they

can react on time if needed. But monitoring collaboration in the decentralized contexts found when using F&PLEs is not trivial [9].

The present work aims to delve into the problem of data gathering in decentralized and heterogeneous contexts for collaboration monitoring purposes. A solution is proposed to support monitoring in an architecture that already integrates VLEs and external tools, GLUE! (Group Learning Uniform Environment) [2]. An illustrative example, based on a real CSCL experience in higher education, is presented. This example provides initial evidences about the feasibility of monitoring using the proposed architectural solution.

The rest of this paper is structured as follows: section 2 identifies some problems that hinder the collection of data for analysis purposes in decentralized and heterogeneous CSCL contexts; section 3 presents our proposal of how a particular architecture for the integration of external tools and learning environments (GLUE!) can be adapted to facilitate monitoring of collaboration; section 4 is devoted to illustrate the proposal by means of an example; and finally conclusions and future work are summarized in section 5.

2 Monitoring collaboration in F&PLE: analysis of problems

There are several problems that impede the application of collaboration analysis techniques in real CSCL scenarios [9]. Among these problems, some depend on the decisions taken during the design of the learning activity [10], such as the need to consider in advance whether or not the tools chosen to support (collaborative) learning will also support monitoring, others are caused by technological reasons. We will focus on these last ones. Technological issues can be classified into three types, which in many cases are inter-related:

- *Data gathering issues:* on the one hand, some tools do not register any kind of data about the user activity and, on the other hand, there are also difficulties with those tools that store this kind of information. Since does not exist a standard format to store it, each tool/environment follows its own approach, for example by means of logs or data bases. Frequently, tools do not provide documentation explaining how the information can be obtained if possible at all, since the information may be located in a system where the access is not allowed.
- *Data interpretation issues:* though there is no common format that models these data, it could be solved defining an ontology or a taxonomy that facilitates the data sharing and interpretation. Frequently, applications do not provide ready-to-use data, such as streamed data or low level events (e.g. interface events), which meaning can not be obtained automatically. In many cases, the problem is due to data are not stored for analysis purposes, but for others such as debugging. These problems highlight the need of taking into account the monitoring requirements when designing and developing collaboration support tools [11].
- *Data integration issues:* at this level, several difficulties have to be addressed, for example, data synchronization. First of all, data has to be timestamped, but there might be additional problems due to time gaps between systems. Another problem

met at integration deals with the identity of the actors and objects manipulated by them. If these identities do not have a common representation shared by all the tools used in the F&PLE, additional mechanisms should be put into practice to be able to integrate them for analysis.

These obstacles increase when the technological context is heterogeneous and decentralized, as it happens in F&PLEs, since it is necessary to process and take into account the information of each source in order to obtain a more general and realistic view of the CSCL activities.

3 GLUE-CAS: collaboration analysis support for GLUE!

In order to present a solution to the aforementioned problems, it is necessary to start from an existing system that allows to show how these issues can be addressed. We have chosen GLUE! for three main reasons; first, this architecture enables the enactment of a wide range of learning situations; second, it provides educators with enough functionality to control the life cycle of external tools within VLEs; and finally because GLUE! supports the enactment and realization of collaborative learning situations.

To understand and formulate the proposal, this section introduces the original GLUE! architecture. Then, the required data and the elements that of the learning scenario that are expected to provide these data are described. The section ends with the description of the extension proposed to the GLUE! architecture in order to support interaction monitoring.

3.1 GLUE! description

GLUE! is an architecture that aims to integrate multiple third-party external tools in multiple VLEs with just a few restrictions and promotes a many to many loosely coupled integration [1].

GLUE! design rationale aimed at reducing the development effort needed to integrate multiple existing external tools in multiple existing VLEs. As Figure 1 depicts, GLUE! follows a three-tier architecture with loosely-coupled distributed services. On the one hand, the core is in charge of managing instances of external tools. It supports the life cycle of tools integrated in VLEs by providing the functionality to find, create, configure, update or delete tool instances. On the other hand, the learning environments and tools located on the left and right of Figure 1, make use of the adapter pattern. VLE adapters enable educators and students to use external tools as if the one of the VLE's built-in tools. To do so, these adapters map activities, users, groups, and roles to tool instances, as with the VLE own tools, but delegate the creation, update and removal of these instances to the GLUElet Manager. Finally, tool adapters translate requests from the GLUElet Manager to the contracts imposed by external tools.

GLUE! defines an integration contract for the communication between the core and the adapters. This contract imposes three mandatory features on VLEs and tools, which most providers already fulfill: they must be web-based platforms, they must offer an extension interface, and additionally VLEs must understand the concept of tool.

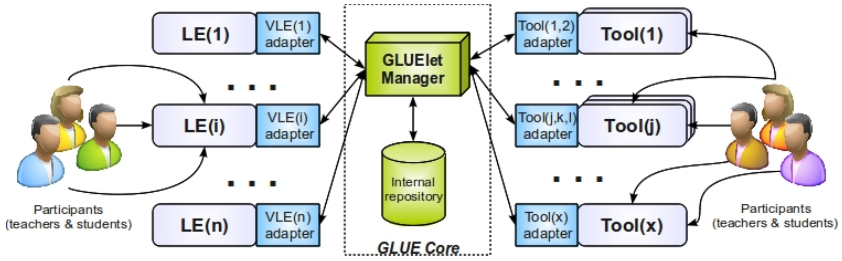

Fig. 1. Overview of the GLUE! architecture.

Besides, the adapters have to meet four restrictions based on widespread standards with a high degree of adoption, aiming at promoting the development of adapters by interested third-parties. First, to facilitate invocation of tools by the learning environment, both the VLE and tool adapters must be *RESTful* compliant. Second, in order to enable educators to configure the tools, tool adapters must follow an *XForms* format, and VLE adapters have to use the *Atom* syndication format, to enable the propagation of forms filled out by educators, and to provide additional information, such as which users are sharing a tool instance. Third, tool adapters must be prepared to retrieve this information from Atom feeds. Finally, tool adapters have to provide URLs representing tool instances.

Several adapters have already been developed: two VLE adapters for Moodle and LAMS; and five tool adapters for Google Docs (Documents, Spreadsheets and Presentations), MediaWiki, Dabbleboard, W3C widgets deployed in Apache Wookie servers, and any URL representing a web content. Besides, current work deals with the development of adapters for PLEs, such as the Southampton Learning Environment (SLE) [13].

3.2 Data and data sources for monitoring collaboration

As it has been previously described, the first step in collaboration analysis is the collection of data representing user interactions from the learning environments and tools. This section describes which data can be logged by the system.

Within the European Kaleidoscope Network of Excellence [8], CSCL data collection issues were considered and previous research works in this topic were reviewed in depth. To enable interoperability among CSCL and collaboration analysis tools, many data formats used in these tools were studied. From this analysis arises what was called the ”Common format“ [4], data format that models the elements involved in the learning activities.

This ”Common format” was described by means of a DTD (Document Type Definition). It is conformed by two main branches: the first branch refers to the **context**, describing the set of *users*, *groups*, *roles*, *learning resources*, etc.; the second branch is devoted to describe the learning activity from a ”dynamic” point of view, in other words, the **actions** carried out by participants, identifying *who* has done *what* and *when*.

”Who” will be one user previously identified in the context, ”what” will be an action among those allowed within the specific learning environment or tool, and ”when” will time-stamp the event.

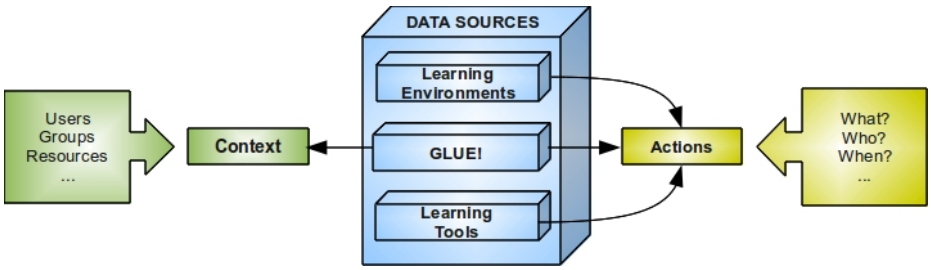

Fig. 2. Data and data sources in learning scenario supported by GLUE!.

As figure 2 illustrates, GLUE! will provide the context description since it is the one who knows all the elements involved in the learning scenario. Then, there will be registered three types of ”user actions”: the ones reported by the learning environments, those provided by the tools, and finally, the accesses to the tools.

3.3 Proposed architecture

The proposal is an extension of the GLUE! architecture. Following this approach, the original architecture has been left untouched or minimally changed, and the design decisions has been respected. Figure 3 depicts the proposal. New elements have been represented with red boxes, and those that required changes have been highlighted with a star.

As it was done with the learning environments and the external tools, to incorporate collaboration analysis tools (CATs) in GLUE!, we will make use of the ”adapter“ pattern. Thus, the new architecture will include **adapters** to communicate the CATs with the GLUE! core. Following the restrictions of the previous architecture, these adapters must be prepared to invoke a *RESTful interface* provided by the GLUE! core.

Besides, in the GLUE! core there will be two new elements in charge of the Collaboration Analysis Support (CAS): the *GLUE-CAS Manager* and the *CAS Repository*. The GLUE-CAS Manager will have to collect and store the data from the learning tools and environments, answer the requests of the CATs, and communicate with the GLUElet Manager. All the data propagated through the architecture will be compliant with the *Atom syndication format*. The CAS-Repository will store the information related to collaboration analysis purpose. This repository is a MySQL data base (as well as the internal repository) that models the learning scenario, following the description given by the ”Common Format”.

The changes required from the previous architecture are two. On the one hand, to register the information about the context and the accesses, the *GLUElet Manager*

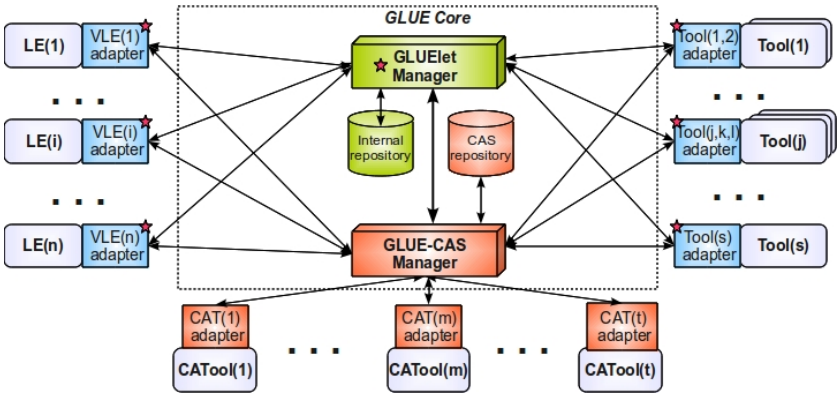

Fig. 3. Overview of the proposed architecture based on GLUE!.

will send a copy of the requests received from the LE adapters. The ones related to *create, configure, update or delete tool instances* will inform the GLUE-CAS Manager about the **context**, providing information about the users, instances and groups assigned to each tool instance. Those requests where a *user asks for a tool instance* will be registered as an **action** of type "**access**". On the other hand, LE and tool adapters must include a module to answer the requests for "event histories". This module will be in charge of the translation of the information from the tool-specific to the common format.

4 Illustrative example

In this paper, the authors address one main research question: "what requirements are imposed to the learning environments and tools to monitor collaboration among participants in a heterogeneous and decentralized CSCL scenario?" In our attempt to understand in depth this research question, this section presents an example based on an authentic CSCL scenario. The authors' purpose is to show whether the proposed architecture allows to collect useful information about the user activity in CSCL scenarios involving several learning environments and tools.

This section is structured as follows: first, we present briefly the main characteristics of the learning scenario; then, we explain how, applying the architecture proposed in section 3.3, information about the users' interaction is retrieved; and finally, the section ends with the discussion of the obtained results.

4.1 Context

The example presented here is inspired by a learning activity developed in Spring 2010, and took place within a fifth year course (out of five) on "Telematics II" of Telecommunications Engineering degree, at the University of Valladolid (Spain), with 22 students attending the course. During this course, one student was in charge of writing down all

the acronyms and terms related to telematics that appeared in lectures. In order to help students to understand and review these concepts, they were asked to elaborate a concept map using them. To elaborate this diagram, students worked in a blended CSCL situation.

The collaboration script implemented a two-level *Pyramid* CLFP [5]. At level-1, *groups* of 2 or 3 participants attended to a face-to-face lab session to carry out the first activity. In this activity, students had to draw a preliminary version of the concept map and write a report with a summary of the main decisions and open issues. At level-2, groups joined to conform *super-groups* (composed of 2 groups) that had to accomplish both a distance and a face-to-face activity. During the former, each *group* had to review and provide feedback on the reports produced by their *super-group* mates; in the latter, they had to discuss and produce a joint version of the map, as well as to perform an oral presentation with a common version of the conclusions and open issues.

Regarding the technological support, teachers used the VLE *Moodle* [1] to centralize the access to all the resources and activities. To accomplish the drawing tasks, students were provided with a shared board (*Dabbleboard* [2]), and in order to explain, review and discuss, they had at their disposal shared documents and presentations (*MediaWiki* [3] and *Google Presentations* [4]). Besides, since the learning tools can not be automatically integrated in Moodle, *the extension of the GLUE! architecture* presented in section 3.3 was used to integrate them into the VLE.

As mentioned beforehand, the scenario depicted in this section is inspired on a real experience that took place at our University. The authors of this paper want to highlight that the choice of the software involved in the example does not attend to reasons related to its suitability for future collaboration analysis. Indeed, this example is intended, first, to reflect the capabilities of the proposal to support collaboration monitoring in real CSCL scenarios; and second, to identify the limitations and constraints.

4.2 Gathering data througout the lifecycle of the learning experience

In this section we will follow the different tasks that teachers and students perform during the learning activity life-cycle, showing how they are dealt with by the architecture in order to retrieve information about the context and the users' activity. We will consider that the phases of this life-cycle are divided into design, instantiation, enactment and evaluation [3].

Once the learning experience has been designed (see section 4.1), it is necessary to proceed to instantiate activities to address the concrete tool instances, participants and groups that will participate in their execution. According to the characteristics of this example, the first task will be to provide the teacher with a Moodle and GLUE! instance. Then, the teacher will register the users that participate in the activity, create as many instances of MediaWiki and Google Presentations as needed, and allocate the participants to the corresponding tool instances.

[1] http://www.moodle.org
[2] http://www.dabbleboard.com/
[3] http://www.mediawiki.org
[4] http://docs.google.com/

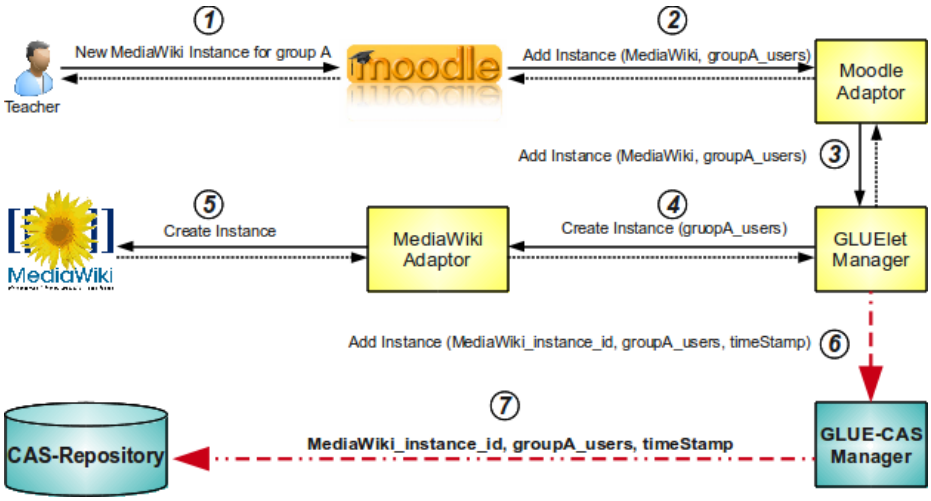

Fig. 4. Data flow during the assignment of users to tool instances.

Figure 4 depicts the process where a MediaWiki instance is assigned to a set of users. To distinguish between the functionalities provided by the original GLUE! architecture and the extension proposed here, the components and connections included in the former are represented by means of yellow boxes and the black arrows, while those components and connections added in the proposal are represented with blue boxes and dashed red arrows.

After registering the 22 students, 10 groups and 5 super-groups; creating the tool instances; and specifying the correspondence between participants and instances, all the information about the context of the learning scenario will be collected in the **CAS-Repository**: *users description, VLE and tool instances description, which instances are integrated in each VLE, and who has access to each instance*.

During the enactment, participants proceed to the execution of the activities themselves. To carry out the activities, they will have to use the tool instances that have been assigned to them in Moodle. Each time a user accesses to an instance (see figure 5), the event that represents this access flows across the architecture and it is registered in the **CAS-Repository**, specifying *who* has accessed to *which* instance and *when*.

During the course of the learning activity, the **GLUE-CAS Manager** will call the adapters of Moodle, MediaWiki and Google Presentations periodically to obtain the *event history* of their corresponding instances. As it is shown if figure 6, Moodle and MediaWiki provide this information about the recent events and changes, which does not happen with Dabbleboard and Google Presentations. However, Google Presentations provides some information about the ”history of changes“ through the user interface, so maybe, developing a specific module for this purpose in its tool adapter, this information could be retrieved. The information obtained from Moodle, MediaWiki and GLUE! is interpreted and each event is stored in the **CAS-Repository**, specifying de-

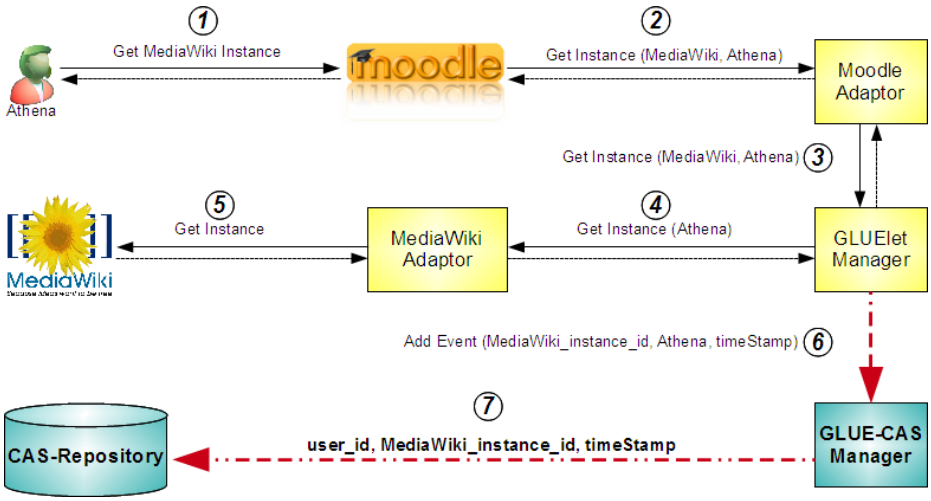

Fig. 5. Data flow when a user accesses to a MediaWiki instance.

tails such as the instance, the user, the time, the action type and the action description (if it is given).

Once the data is retrieved, it can be used for carrying out collaboration analysis, which, as mentioned at the beginning of the paper, is a major issue in the evaluation of CSCL experiences. The example described in this section illustrates the problems one may encounter when trying to get this data from the heterogeneous sources typical in F&PLEs, and proposes a possible solution to meet them.

4.3 Findings and discussion

As mentioned in the Introduction, the main goal of the example was to get evidence on whether the presented architecture can contribute to support automated or semi-automated collaboration analysis in CSCL scenarios. The proposed extension of GLUE! supports the enactment of a wide range of CSCL situations such as the one presented in the previous section. In addition, it adds to the existing functionalities the possibility to collect data about the users' collaborative interactions.

In order to assess the proposal, we will discuss the results with respect to the problems identified in section 2. On the one hand, the example shows the impact of not taking into account the needs of the collaboration monitoring during the design phase. Choosing tools that do not provide information about the user activity limits the analysis, allowing just a partial view about the collaboration among participants. On the other hand, regarding the technological issues, we will review each type in depth:

– *Data gathering:* the example illustrates that it has been possible to integrate the tasks related to the data gathering with no impact on the learning software and transparently to users. Besides, though Dabbleboard and Google Presentations do

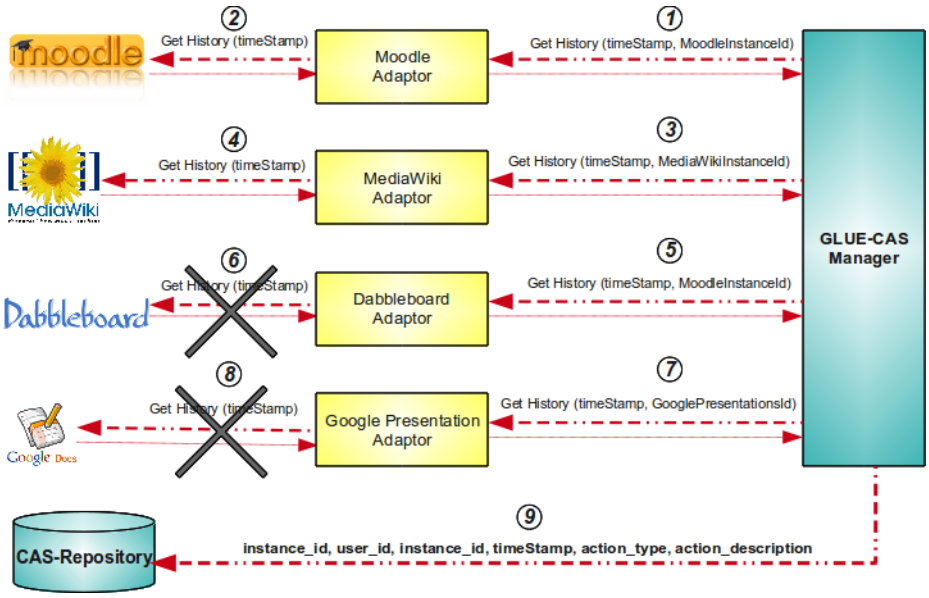

Fig. 6. Data gathering from the environment and tools used in the learning scenario.

not offer programatically any kind of data about the user activity, by means of the architecture, it has been possible at least to register when users access to the tool instances. Regarding Google Presentations, the format that this tool uses to display recent changes in documents (through the interface instead of via files or logs), has not allowed to include the information. In this first approach to the architecture, it would be necessary to develop a specific module for this purpose in its tool adapter, but this is not the ideal solution because of the cost of developing specific gathering modules for each tool.

- *Data interpretation:* since each learning environment/tool adapter is in charge of interpreting the events registered and translating them into the ”common format“ described in section 3.2, once the information arrives to GLUE-CAS manager to be stored, the interpretation tasks have finished. This approach distributes the developers efforts since each adapter will only need to include a module for mapping the data from the proprietary format to the common one, instead of developing a translator for each collaboration analysis tool.
- *Data integration:* by translating all the data collected to a common data format we are providing the basis for data integration, but a full integration requires to deal with problems of identity that require a broader approach. However, the architecture proposed can help to partially solve the problems posed by the identities of the users and objects in these heterogeneous systems. For example, knowing when a user asks for a tool instance, and comparing this information with the one provided by the tool, a matching between the identities of the user in the two environments can be made. Further work has to be done to test the feasibility of this approach.

These findings provide initial evidences on the capabilities of the proposed architecture to gather relevant information about the user activity during the learning process. In order to obtain an educational value, this information has to be still integrated and analyzed, but it already represents a step towards the collaboration monitoring of CSCL activities in real practice.

5 Conclusions and future work

This paper addresses the problem of evaluating CSCL experiences supported by F&PLE, and analyzes the technological issues that appear when trying to apply collaboration analysis techniques in such decentralized and heterogeneous contexts. Also, an architectural proposal based on GLUE! has been presented, and its viability has been illustrated by means of an example based on an authentic learning experience where several external tools are integrated in a VLE.

This proposal shows how to solve some technological challenges of monitoring collaboration in F&PLE, especially those related to data gathering. Besides, some other problems related to the impact of the learning design over the collaboration analysis are highlighted. This line of research is also relevant and its study has being addressed in other paper presented at this conference [10].

Future work lines include the implementation of the proposal and its evaluation in authentic CSCL scenarios, as well as the integration of multiple collaboration analysis tools.

Acknowledgements

This research has been partially funded by the Spanish Ministry of Science and Innovation Projects TIN2008-03-23 and IPT-430000-2010-054, and the Autonomous Government of Castilla and León, Spain, Project VA293A11-2.

References

1. Alario-Hoyos, C., Bote-Lorenzo, M., Gómez-Sánchez, E., Asensio-Pérez, J., Vega-Gorgojo, G., Ruiz-Calleja, A., Velasco-Villanueva, D.: GLUE!: An Architecture for the Integration of External Tools in Virtual Learning Environments. Internet Computing (2011), (submitted)
2. Alario-Hoyos, C., Wilson, S.: Comparison of the main Alternatives to the Integration of External Tools in different Platforms. In: Proceedings of the International Conference of Education, Research and Innovation. pp. 3466–3476. ICERI'10, Madrid, Spain (2010)
3. Gómez Sánchez, E., Bote Lorenzo, M., Jorrín Abellán, I., Vega Gorgojo, G., Asensio Pérez, J., Dimitriadis, Y.: Conceptual framework for design, technological support and evaluation of collaborative learning. International Journal of Engineering Education. 25(3), 557–568 (2009)
4. Harrer, A., Martínez Monés, A., Dimitracopoulou, A.: Technology-enhanced learning. Principles and products., chap. Users' data: collaborative and social analysis., pp. 175–193. Springer, UK (2009)

5. Hernández-Leo, D., Villasclaras-Fernández, E., Asensio-Pérez, J., Dimitriadis, Y.: Handbook of Visual Languages for Instructional Design: Theories and Practices, chap. Diagrams of learning flow patterns' solutions as visual representations of refinable IMS Learning Design templates. IGI Global (2007)
6. Kahrimanis, G., Avouris, N., Komis, V.: Technology-Enhanced Systems and Tools for Collaborative Learning Scaffolding, chap. Interaction analysis as a tool for supporting collaboration. An overview. Springer (2011)
7. Koschmann, T.: CSCL: Theory and Practice of an Emerging Paradigm, chap. Paradigms Shift and Instructional Technology, pp. 1–23. Lawrence Erlbaum Associates, Mahwah, NJ USA (1996)
8. Martínez Monés, A.: Library of Interaction Analysis Tools (2005), Kaleidoscope Noe. JEIRP IA. Deliverable D.31.2
9. Martínez-Monés, A., Harrer, A., Dimitriadis, Y.: Analyzing Collaborative Interactions in CSCL: Methods, Approaches and Issues, chap. An interaction-aware design process for the integration of interaction analysis in mainstream CSCL practices, pp. 269–291. Springer (2010)
10. Rodríguez-Triana, M., Martínez-Monés, A., Asensio-Pérez, J., Jorrín-Abellán, I., Dimitriadis, Y.: Monitoring pattern-based CSCL scripts: a case study. In: 6th European Conference on Technology Enhanced Learning: Towards Ubiquitous Learning. EC-TEL'11, Palermo (Italy) (September 2011)
11. Soller, A., Martínez, A., Jermann, P., Muehlenbrock, M.: From Mirroring to Guiding: a review of the state of the art in interaction analysis. International Journal on Artificial Intelligence in Education 15, 261–290 (2005)
12. Stahl, G., Koschmann, T., Suthers, D.: Cambridge handbook of the learning sciences, chap. Computer-supported collaborative learning: an historical perspective., pp. 409–426. Cambridge University Press, Cambridge, UK (2006)
13. White, S., Davis, H.: Making it rich and personal: crafting an institutional personal learning environment. International Journal of Virtual and Personal Learning Environments 2(3) (2011), http://eprints.ecs.soton.ac.uk/22030