An interaction-aware design process for the integration of interaction analysis into mainstream CSCL practices

Alejandra Martínez-Monés[1], Andreas Harrer[2], Yannis Dimitriadis[1]

[1] GSIC-EMIC Group. University of Valladolid. Spain. amartine@infor.uva.es, yannis@tel.uva.es

[2] Catholic University of Eichstätt, Germany. andreas.harrer@ku-eichstaett.de

Abstract The potential capabilities of computers to support analysis of interaction data have attracted the attention of the CSCL research community. This has led to the proposal of a number of *interaction analysis tools*, which process interaction data to meet different purposes. These may range from supporting researchers in ethnographic studies to providing advice to the students. However, after several years working with classroom-based CSCL experiences, we have found that both researchers and practitioners meet many difficulties to apply these potential benefits to their CSCL settings. Thus, the first goal of this chapter is to provide a systematic analysis of the problems that can be found when trying to apply interaction analysis tools to CSCL settings, which are then classified at into three levels, namely: application, architecture and design levels. Then, we outline the path for possible solutions to face these problems. According to this, the issues identified at the design level call for an IA-aware design process where we distinguish between *co-design* approaches that directly integrates the diverse needs of learning and analysis, and *multi-perspective* approaches that treat them independently at an initial stage. On the other hand, the problems at the application and architecture levels must be faced by technology-driven solutions, such as the use of decoupled architectures, either based on inter-process communication or on interchange of log file information. Several open issues have also been detected that need adequate solutions, as e.g., the semantic integration of log-files when multiple self-contained learning tools are used for an integrated analysis.

1 Introduction

The massive use of computers to support learning has brought the possibility to apply e-research capabilities to the learning sciences (Markauskaite & Reimann,

Sadhana Puntanbekar, Gijsbert Erkens, and Cindy Hmelo-Silver (Eds.). *Analyzing interactions in CSCL: Methodologies, Approaches and Issues*. Springer (Enviado)

2008). Computers can store large amounts of interaction data that can be then analyzed by automatic or semi-automatic means to serve different purposes, from pure research to formative evaluation or monitoring approaches. This challenge is even stronger in CSCL scenarios, where the complex and multimodal interactions among participants are totally or partially mediated by computers, and thus, not directly observable by traditional means in the remote scenarios. This has raised the interest of at least two different trends: those coming from ethnographic or ethnomethodology traditions, that see the computer as a tool to help researchers store and analyze detailed accounts of the interactions (Guribye & Wasson, 2002), and those coming from engineering fields aiming to produce automatic or semi-automatic results that can directly help researchers or practitioners in their work (Moreno & Ventura, 2007). Tools coming from these two trends have been conceptualized in Soller et al., (2005) as a continuum from mirroring tools, that store and reproduce interaction data to facilitate their analysis, to guiding tools that perform themselves the analysis and give direct advice to their users. Due to their emphasis in the analysis of interactions among participants, we will refer to all these tools as computer-based Interaction-Analysis (IA) tools, or IA tools for short. Therefore, an IA tool in this work must be understood in a broad sense as any (software) system able to take interaction data as an input, process it, and show the results of the analysis to its users. These users may be researchers, teachers or students, depending on the specific case. The format used to display the results may be very different, depending on the target user and the purpose of the analysis.

The potential benefits of computer-based analysis of interactions among participants has led to a significant growth of interest in terms of research papers (Harrer et al., 2009), meetings (Dwyer et al., 2008; Law et al., 2009), and projects (see for example, those funded by the European Commission as *Argunaut* (De Groot, 2007) or *Kaleidoscope* (Kaleidoscope, 2007), or other international initiatives such as several international research collaborations located in the Pittsburgh Science of Learning Center (PSLC). However, in spite of the major impact in the research community, IA tools have not found yet their place in the classroom, and they have not been incorporated in real-life CSCL scenarios beyond pilot studies directly guided by researchers (Martínez-Monés et al., 2008a).

Several reasons hinder IA tools to move into mainstream educational practices (Dimitracopoulou, 2005; Soller et al., 2005). Some of them deal with the difficulty of having educators designing and using IA tools in CSCL. First, there is a general resistance to adoption of technological innovations in classroom, especially when these require a major shift in pedagogy, as it happens in collaborative learning. On the other hand, there is an intrinsic complexity of the concepts and indicators involved in interaction analysis of collaborative learning. Thus, it is really difficult to advance in the definition and selection of appropriate indicators, as well as their visualization to the different actors involved in the teaching/learning process (teachers, students, evaluators). Finally, several technological problems have been reported with their origin in a significant mismatch between the learning management systems, be them generic or specific for CSCL, and the IA tools or services.

In that latter case, it is not straightforward to integrate all these tools and services in platforms and put them in practice (Markauskaite & Reimann, 2008). These difficulties have been experienced by the authors in several international and local research projects, and constitute the main motivation of the reflections shared in this chapter.

We present a systematic analysis of the problems and the eventual paths to solutions related to a wider adoption of IA tools and practices in CSCL environments. The approach followed here is mainly data-driven or bottom-up, illustrating both problems and solutions with examples drawn from the mentioned research projects and classroom-based case studies in which the authors have been directly involved. Although such an approach does not guarantee any generalization, it is expected that the theoretical analysis and discussion presented in this document may foster a major reflection and eventually an approach adopted by the global community of researchers, educational practitioners and technology providers.

Section 2 presents a general IA process model that allows a common understanding of the field, as it emerged through various projects and the joint effort of several research teams within the European Kaleidoscope Network of Excellence. Then, Section 3 proposes a classification of problems for the integration of interaction analysis into mainstream CSCL practices, illustrated through several examples drawn from practice. The following section provides a set of solutions that can be considered as design or technology-oriented. The last section includes an overall discussion based on the main conclusions and suggests a series of orientations and future steps that may be taken into account by the global community.

2 Overview and model of the Interaction Analysis process

As mentioned beforehand, computer supported interaction analysis has raised a growing attention in the learning sciences. A good example of this interest is the fact that it was a prominent theme in the mentioned Kaleidoscope Network, where several projects and initiatives took place in order to integrate and leverage current research on computer-supported interaction analysis (Kaleidoscope, 2007). However, from the beginning, it became clear that the different research perspectives that converged in these projects did not share their understanding of the involved processes, tools, and methods. Thus, a major goal of the projects was to define shared conceptualizations of the interaction analysis process. The first model was produced in the ICALTS project, and consisted on a framework to describe the main concepts underlying computer-supported interaction-analysis. This model illustrates the close interplay between context, the CSCL environment and the IA tools, and helps to conceptualize single interaction analysis methods. Later, in the CAViCoLA project, the scope of this model was extended, by defining a general framework for the whole process of computer-supported collaboration analysis, considering aspects such as data and method triangulation, appropriate for the analysis of complex learning scenarios (Harrer, et al., 2007). Their main purpose was to enable researchers from different traditions share their concepts and eventually their methods and tools to support these processes. With this objective, but on a more technical level, in the IA project we proposed a *common format* to enable interoperability among CSCL and IA tools (Harrer, et al., 2009). This format will be discussed later, as it is an example of the possible solutions we propose in this chapter for a major adoption of IA tools in CSCL.

The *ICALTS* interaction analysis model is especially useful to understand the interplay between learning environments and IA tools, and will be shortly described here, as it is a useful model to frame the discussions included in this chapter.

As shown in Figure 1, the analysis of participants' interactions is usually driven by some sort of hypothesis which shall be proven or rejected by certain observation. So the first question is "What are the important questions to ask?", "What do I want to analyze?". The answer to this question influences (indicated by the large arrow in Figure 1) the choice of *indicators* used to conduct further analysis. The questioner will choose an indicator able to express the concept to be analyzed, i.e., the choice of an indicator is influenced by the target group and will vary with the interest of the questioner. For example, researchers might want to obtain a detailed view of the process, while students might benefit from visualizations of their participation in a forum as compared to other students in the same classroom.

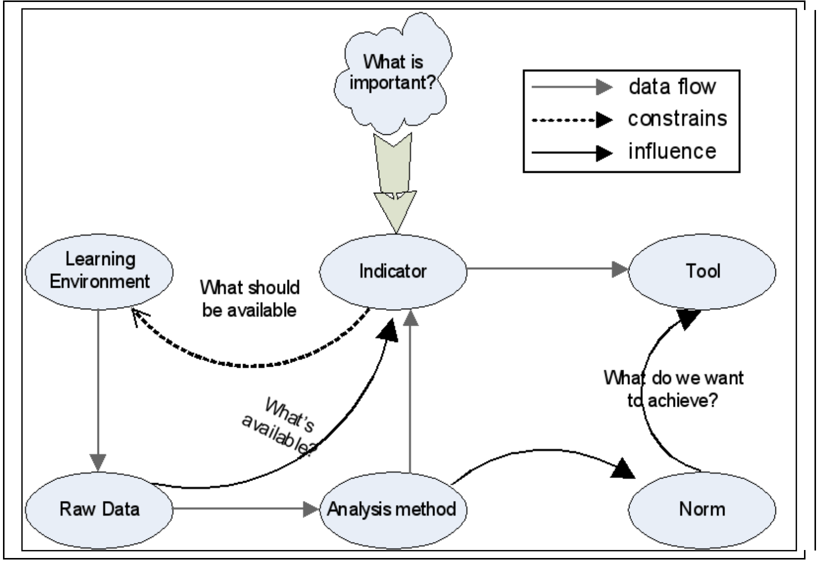


Fig. 1. Schema of the interaction analysis process

The choice of an indicator determines certain constraints a *learning environment* has to fulfill (indicated by the dashed arrow in Figure 1). Thus, each indicator determines "what should be available to compute the indicator's values?", e.g., to measure social structures or patterns of interactions it is necessary to capture the information related to “who is sending messages to whom”, etc. The learning environment is responsible for generating the *raw data* (e.g. log files) used in further analysis steps to compute the chosen indicators. Sometimes the data required by an indicator cannot be supplied by the used learning system causing the set of indicators to be narrowed or re-assembled. Thus, it can be seen that the availability of appropriate raw data influences the choice of indicators as well (shown by the curved arrow). These mutual constraints between the learning environment and the IA tool will be illustrated in the next section with examples taken from the authors' experience.

In the end the *analysis method* relays the indicators to a certain *tool* that uses these indicators and eventually presents them to the intended target users (researchers, teachers, students, etc). In some circumstances, for the concrete utilisation of the indicator a *norm* can be applied. This norm defines desired values and behaviour, such as “less than 10% participation of one student is too low for good collaboration”, and can be employed for providing specific messages or visualizations. In CSCL, the use of these norms is not always possible or even desirable, and thus, this element can be considered an optional aspect of the process.

3 Main problems for the integration of interaction analysis into mainstream CSCL practices

As already mentioned in the introductory section, the first objective of this chapter is to present, classify and analyze the problems that impede a wider adoption of IA tools by end users, be them researchers, practitioners or students. Taking into account the brief general overview of the interaction analysis process presented above, we present in this section a structured description of the problems met by the authors while designing and enacting CSCL scenarios. Eight of these experiences will be used to exemplify the listed problems. As it is not possible to describe in detail all the experiences that have been analyzed, we will employ one of them, from the *MosaicLearning* project (denoted as the *Mosaic* experience) to illustrate most of the problems listed in this section before we describe them with some detail in section 3.2.

3.1 An illustrating example: The Mosaic experience

A clear example of the problems that appear when trying to apply computer-based IA in real practice was experienced by the authors in the *MosaicLearning* research project[1], where several groups from three Spanish universities set out to share

---

[1] http://mosaic.gast.it.uc3m.es

their learning-support tools and methods (Fuente et al., 2008). More concretely, the overall objective was to study the issues that arise when designing, deploying and enacting a fully collaborative learning experience with remote students. The authoring tool *Collage* was used to design the learning script, which was enacted by *Grail*, an IMS Learning Design run-time environment fully integrated with the .LRN Learning Management System. (.LRN, 2008). Although a final general evaluation phase was also foreseen, the case study was not planned taking analysis purposes explicitly into account. This fact, together with the complexity of the final setup convert the *Mosaic* experience in a particularly good example of the many problems that can be found when researchers try to apply analysis methods to CSCL settings.

Table 1 describes the main aspects of this experience which is briefly introduced here. The course was an undergraduate program on Grid computing in three geographically distant higher-education institutions to a total of 12 students divided into groups of 3-4 members each (Fuente et al., 2008). After an initial phase of individual work, the students had to collaborate remotely in order to produce a joint conceptual map visualizing the main topics addressed by two technical reports on grid services and the service oriented computing paradigm.

Table #-1. Context of the *Mosaic* experience

Dimensions	**Characteristics**		
Scope	**Category**	**Size**	**Number of groups**
	Workgroup of 3-4 people	12 students	4 groups
Type of interaction	Distance		
Educational level	University (Graduate course)		
Tasks	**Period**	**Task**	
	Individual phase	Reading of two technical reports about the same topic and construction of a conceptual map with the main topics	
	Collaborative tasks	Build a joint conceptual map on the basis of the two previous ones	
Tools	**Tool**	**Usage**	
	Cmaptools	Construction of conceptual maps	
	Kedit	Text processing	
	Kolourpaint	Image edition	
	Skype	Remote discussions	
	Grail+.LRN	LMS engine for the deployment of the Unit of Learning	
Collaborative experience	Students had different previous level of experience of computer-supported collaborative learning. There was no teacher, with the participants regulating tasks, time, discussions and solutions.		

In order to provide the students with a collaborative infrastructure, the team set up a VNC (*Virtual Network Computing*) server combined with single-user applications. VNC enables sharing of applications running in a remote server, which allowed us to use single user applications to support the collaborative tasks. To carry out these tasks, the students were provided with instances of *Cmaptools*[2], *Kedit*[3] and *Kolourpaint*[4] (see Table 1 for further details on their use). Remote communication was mediated by *Skype*[5]. It is important to note that this setup was chosen for practical reasons among other possible configurations. For example, *Cmaptools* has a collaborative version, but the design team had to discard its use, due to licensing and performance problems, forcing us to use an individual version, which was shared with the aforementioned VNC server to enable collaboration. *Kedit* and *Kolourpaint* were also a convenient choice, as they are already provided by the Linux-based environment where the experience took place.

Besides the actual design and enactment of the collaboration script, the most relevant aspect of this case for this chapter relates to the difficulties that appeared when trying to analize the experience. The team aimed to study it following the mixed evaluation method defined in Martínez et al. (2003). This method describes how to combine qualitative, quantitative and social network analysis techniques in a semi-automatic analysis based on data coming from different sources. Among these sources, the method includes data logs representing computer-mediated interactions between the participants in the experience. However, several obstacles were met when trying to collect the interactions mediated by the system in order to analyze them. First of all, .LRN only manages the use of the applications, and thus provides data on applications usage, but not about the users' actions on these applications. In fact, these actions took place through the single-user applications shared by the VNC server. However, with this VNC-based setup, the information about who originates which action was lost, which is a major obstacle to carry out any **analysis on the actors' performance**. Moreover, the actual implementation of VNC used (*tightvnc*) provides for screen capturing, which could have been combined with the audio recorded from the Skype sessions. However, it was not possible to analyze these data, because no **synchronization** mechanism between the videos from VNC and the audio from Skype had been prepared.

This experience comprises many of the problems that are normally found when researchers or educators try to apply IA to real practice. In the next subsection we describe these problems in a structured way, providing further examples taken from the authors' experience in CSCL-based projects and classroom situations.

---

[2] http://cmap.ihmc.us/

[3] http:// www.kedit.com/

[4] http://www.kolourpaint.org/

[5] http://www.skype.com

3.2 A data-driven analysis of problems regarding integration of IA tools and CSCL environments

A systematic clustering of the cases reflecting authors' experience in the design and enactment of CSCL scenarios indicates that the problems found to integrate learning environments and IA tools can be classified at three levels. They may be due to the characteristics of the applications (application level), to the actual architecture used to enable collaboration (architecture level), or to deficiencies in the overall design process (design level). This subsection elaborates on these three levels and provides further examples of cases (see also case codes in Table 2) where the authors met them in real practice.

Most of the cases originate from projects within the *Kaleidoscope* European Network of Excellence. In the *IA* project we worked in the integration of the teams' systems in a shared library of IA tools (Martínez et al., 2005) ([Kal-IA] in Table 2) where we met some of the problems discussed here. Later, in *CAViCoLA*, we had a number of experiences to share tools and data between the participating teams. One of these experiences between Universities of Duisburg and Patras ([Du-Pa dyads] in Table 2) also illustrates specific problems found when working towards this integration (Harrer et al., 2006). In parallel to *CAViCoLA*, in the *CCI-IA* project ([Kal-CCI] in Table 2) we were asked to integrate our IA tools in the network's web-based communication and collaboration platform, in order to provide support to the members of the network. This gave us further experience in the real problems that are met when trying to apply interaction analysis tools to existing environments (Bratitsis et al., 2008). Besides these international projects, we also provide examples of local experiences where these problems or their solutions were clearly manifested, namely: the *Mosaic* experience ([Mosaic] in Table 2), reported in the previous subsection, the use of Group Scribbles in real classrooms ([GS] in Table 2), and the use of *BSCW* and *wikis* to support project- and inquiry-based learning ([BSCW] and [Wikis] in Table 2), and finally, an experience run in Germany with students' and teachers' tagging behavior of learning material ([Tagging study] in Table 2). These will be shortly described along with the problems they illustrate at the three levels that structure this subsection.

At the *application level*, it is frequent that applications **do not provide ready-to-use data** about their interactions, as was the case in the *Mosaic* experience with *Kolourpaint*, *Kedit* and the single-user version of *Cmaptools* that was employed. Another typical scenario occurs when some kind of **data** is provided, but it **does not give enough information** to perform the required analysis. For example, in the *Mosaic* experience, the data provided by *.LRN* was insufficient to perform an

Table #-2. Problems found in integrating interaction analysis and learning environments, together with the cases in which these problems were encountered

Problem		Mosaic	Kal - CCI	Kal - IA	GS	BSCW	Wikis	Du-Pa dyads	Tagging study
Application level	Data logs are not provided or are insufficient	X				X	X		
	Data is not documented	X			X	X	X		
	Data is not processable	X							
Architecture level	Not all relevant data is captured	X	X			X	X	X	
	Data are not synchronized	X							
	Data formats are not compatible			X				X	
	The architecture does not allow to get crucial data	X							
Design level	The design did not take into account the need to analyze data	X							
	The intended object of analysis has not been integrated into the teaching practice								X
	The IA tool constraints the choice for the CSCL application					X			

analysis. We have met this difficulty in several other authentic classroom experiences, where *BSCW* and *wikis* were employed to mediate collaboration. *BSCW* is a shared workspace system that provides an awareness information service to its users that can be employed for analysis. However, not all relevant data, such as repeated document readings, is provided by this service. *Wikis* usually provide ready-to-use accounts of the history of page modifications, but no data about other actions, such as readings, is recorded. Readings are usually relevant for analysis and therefore, we had to find a workaround in order to be able to analyze their data with our IA tools. In the experience based on BSCW, we could use system logs, which are not meant for end-users, but that provide useful data for analysis at a good level of detail (Martínez et al., 2003). In the *wiki-based* experience, we had to code a specific *MediaWiki* extension to be able to record readings and consider them in our analysis (Martínez-Monés et al., 2008b). Besides this, we also met difficulties to access and understand the data due to the **lack of documentation**. Thus, the use of these data for analysis requires a demanding effort and the participation of programmers before they can be employed for interaction analysis purposes. Moreover, this set of problems illustrates the fact that application builders do not think that logs are useful for end-users, and for this reason, they do not provide an easy access to them. Another example of this lack of documentation was met by the authors in their experiences with *GroupScribbles* ([GS] in Table 1), a tool that enables students and a teacher to share contributions on sheets similar to post-it notes and to jointly manage the movement of these electronic notes within and between public and private spaces (Roschelle, et al., 2007), (Dimitriadis, et al., 2007). The first version of this tool, based on the *TupleSpaces* architecture, provided usage logs only for internal technical debugging purposes, and therefore the lack of an adequate documentation impeded the use of these logs for analysis. The net result of this situation is that valuable information on the interactions among students and teachers could not be used for analysis. The last problematic issue that can be found at the application level is that the **data** itself might **not be directly processable** by computer tools in order to extract indicators. This is the case with streamed data such as audio, video, etc., like the data provided by *Skype* and *VNC* ([Mosaic]). These data is appropriate for a thorough review of the experience, but it is not directly understandable by an automatic data analysis application. It requires human intervention, in the form of coding or labeling before this data is usable for computing IA indicators. In fact, this is an approach followed by many IA tools aimed to supporting research, like *ActivityLens* (Fiotakis et al., 2007) or Tatiana (Dyke, 2008) but it becomes an open issue how these data is meant to be prepared and handled by practitioners, which do not have the time and resources needed to process them.

Some of these problems at the application level are translated to the *architecture level*, i.e., to the actual configuration of teaching/learning tools set up in order to support the collaborative tasks. For example, even if the experience uses an application with a complete and well documented data log, it might happen that this application covers only part of the interactions, and therefore, the **data provides only a partial view** of the collaborative tasks carried out by the students. In the Mosaic experience, we could not use *.LRN* to store and analyze the participants' interactions because this system was used to launch applications, but not to mediate interactions. Examples of this problem can be found very frequently, as students' communication is normally not controlled, and happens outside the system many times. This happened in the aforementioned experiences, where we employed *BSCW* and *wikis* to support the collaborative tasks. In spite of the mentioned difficulties, we were able to analyze interactions mediated by these platforms, but the most of the students' internal communication was mediated by e-mail or instant messaging, which was not possible to record, and was thus not included in the analysis. Another case where this problem was met was in several cross-site studies between universities of Duisburg and Patras ([Du-Pa dyads] in Table 2) (Harrer, et al., 2006), where the remote interaction, especially the use of their external chat tools, between students was not always possible to register, and thus, it could not be analyzed. A third example of this was experienced by the authors in the CCI-IA project, where we were asked to provide an IA service for the Kaleidoscope site ([Kal-CCI] in Table 2), based on the analysis of the interactions on the platform. Again, the main problem found in this project was that most of the interactions among teams happened outside the system (Bratitsis, et al., 2008).

A second problem to consider at the architecture level happens when the data is stored, but it is not possible to integrate them because they are not **semantically** compatible. A very clear example of this issue was met by the authors in the aforementioned effort to build a library of shared IA tools ([Kal-IA] in Table 2). This effort was hindered by the fact that their data models were not compatible. A more concrete example of this problem was met in the mentioned cross-site studies between Duisburg and Patras, where it was difficult to integrate process data of the students' interaction with their products, etc. Related to this integration problem we can find the difficulties to **synchronize** data from different sources. Synchronization is feasible provided some previous requirements are met, such as that the data is time-stamped with a common time reference. Then, there exist software packages specifically oriented to synchronize these data and show it to the researchers for its coding and analysis. However, as shown by the *Mosaic* experience, this data-stamping requires an extra effort to the designers which is not always possible. Finally, it might happen that the **overall architecture** used to enable collaboration **hinders analysis**. This was clearly reflected in the *Mosaic* experience previously described. The VNC-based implementation used did not allow distinguishing who executed the actions, which is a major problem for the later analysis of these data.

A major source of problems related to the integration of IA in real practice deals with the *design level*. First, when setting up a scenario to enable and study collaboration, the aspects related to interaction analysis are not normally considered a first priority, and many times this means that they are neglected in benefit of more immediate requirements, such as the need to provide for collaboration-enabling functionalities and an acceptable performance. This was the main reason why the Mosaic setup did not facilitate the recording of interactions, and there are many other examples where this lack of priority ends up with a setup that does not allow researchers or educators to study the case in its fully extent. This is a very common situation in every innovative project, but it would be a minor problem if the tools provided ready-to-use logs. Therefore, this shows that the problems at the tool level are also reflected at this one. Another issue that must be considered at design level is whether or not the actual analysis objectives are possible to reflect in the teaching practices. The need of using real scenarios to analyze CSCL practices often meets the obstacle that it is **difficult to integrate these aspects in normal teaching practices** and environments, which tend to be slow in innovation. An example of this problem was found by the authors in an experience ([Tagging study] in Table 2), where the phenomenon of social tagging of learning material was explored (Lohmann et al., 2008): since the e-learning platform used at that university did not integrate a tagging feature, the experiment was emulated outside of the learning platform in a pen-and-paper pre-study. Even when this integration has been achieved and an authentic scenario is found where teaching practice meets research objectives we can find a new problem. As illustrated by the *ICALTS* model described in Section 2, the learning environment can be restricted to those that provide the data needed by the analysis, thus restricting the choice of the collaboration-supporting tools to those that might be not the most appropriate ones for the planned tasks. An example of this case was met in the *BSCW*-based experience, where we could not upgrade the system to a newer version, which did no longer provide the system logs we were using for analysis. This problem has its roots on the fact that not all collaboration-support tools provide ready-to-use log files, or that these logs are not compatible among each other.

The above data-driven analysis provides an overview of the problems encountered, when researchers or educators want to integrate interaction analysis techniques and tools in CSCL environments. The conceptual assignment of problems to three levels aims at analyzing the common issues and providing a global explanation of their origin. However, the above analysis provides sufficient hints on the existing correlation among levels, as well as the actors involved. In this sense, the design level originates many of the problems related to the educators or researchers, i.e., the practitioners or end-users, while the application and architectural levels deal with computer scientists and engineers, i.e., the technology providers of CSCL environments and IA tools. Apparently, there is a significant gap between the two worlds which impedes the solution of the observed problems.

The following section points to some solutions that may bridge the gap and allow for a seamless integration, taking into account trends and advances in different fields. These may focus either at the design level (integrated or multi-perspective scripting or learning design), application (IA-aware tools and learning environments) or at the architectural level (decoupled and service-oriented architectures and common data protocols).

4 Towards an integrated perspective on learning and analysis activities in CSCL

From the problems described in the previous section we can derive a need to address the issue of interaction analysis early in the preparation of the learning activities. Our first ideas on that have been described in Martínez-Monés et al. (2008a) and will be expanded in the following. As a first differentiation we propose solutions that are mainly based on direct consideration during the *design* process and solutions that are mainly based on preparations of the learning *technology*, i.e., the tools used. The design-driven solutions already try to take into account the design level issues of the previous section, i.e. they explicitly make the need for analytical activities visible in the design process. The technology-driven solutions mainly tackle the application and architecture level by means of providing well-defined interchange formats between learning and analysis tools.

4.1 Design-driven solutions

For the first type of proposals the needs of interaction analysis are directly integrated into the design process, which means that the activities related to the analysis (like assessment, evaluation, research or monitoring) are explicitly modeled and specified, together with the rest of the learning process before the activity is implemented. Otherwise the problems raised in the previous section might arise, when studying the collected data, possibly finding shortcomings in the richness, availability etc.

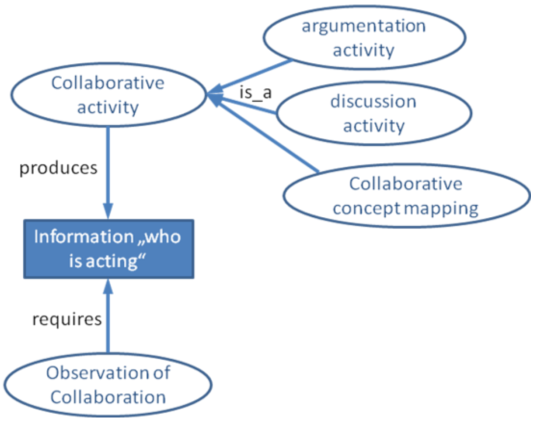


Fig. 1 An example of joint study and design of learning and interaction analysis activities**.**

Since the concerns of the learning process and those of analysis can have different needs and issues, multiple and potentially conflicting aims have to be addressed. A usual means to tackle this integration problem in computer science is either to follow a *co-design* approach that directly integrates the diverse needs or to use a *multi-perspective* approach where each facet is represented individually and relations between the different perspectives are made explicit by meta-rules or constraints. Prominent examples for these different approaches are hardware/software co-design or the multi-perspective software modeling language UML (unified modeling language with multi-perspective diagrams). Interestingly, this differentiation has also been discussed in the area of learning design (Botturi & Stubbs, 2007) where the dimension "perspective" has been used to differentiate between single-perspective and multiple-perspective approaches to represent learning designs. For the integration of the needs that interaction analysis brings forth we use a similar differentiation.

In a co-design approach of the learning and analysis processes, the aspects of the *learning/teaching* activities and the *observation/analysis* are taken into account simultaneously, posibly by a single designer. One design process using this approach is described in Villasclaras et al. (2009) where both aspects are represented by means of a pattern language: collaborative learning flow patterns (CLFP) and assessment patterns describe at an abstract level the essential characteristics of the activities. While a CLFP describes the learning/teaching activities, the assessment patterns describe the activities related to the collection of information for assessing the students, i.e., an observation/analysis task. According to this approach, the designer of a learning scenario chooses and configures both types of patterns in an integrated process. Since there might be constraints between specific learning and assessment activities, these constraints have to be represented in a pattern language to inform the designer of potential problems when using patterns of different type in combination. Figure 1 shows an example of how learning activities (at the top of the figure) and analysis activities can be related to each other by high-level dependencies helpful for coordinated planning of both aspects: the choice of a collaborative concept mapping creates a potential information "who is acting" that can be used by an observation activity.

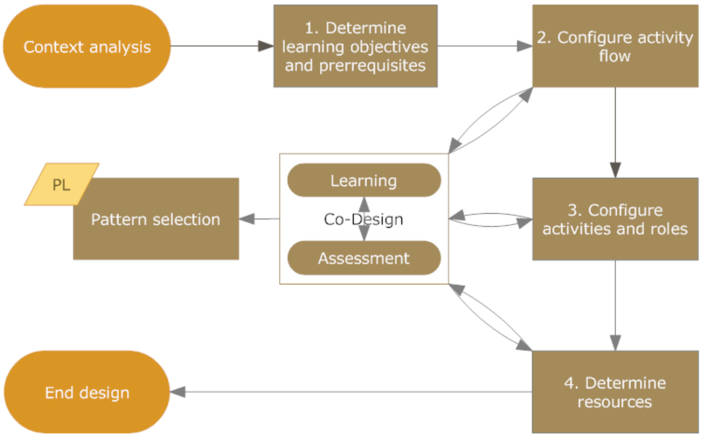


Fig. 2. A unified view of the co-design process. Adapted from Villasclaras et al., (2009)

Thus, it is reasonable to think of following a co-design process such as the one shown in Figure 2, in which learning and analysis needs are taken into account in an integrated way. An example of a concrete solution that could be produced with this co-design process is shown in Figure 3, where the peer-review assessment pattern (that can be seen as a specific kind of interaction analysis activity performed by a peer student) is designed to be used together with the pyramid, jigsaw and think-pair-share patterns, which are specific cases of CLFP, i.e., learning patterns. For more information on these pattern-based solution, see Villasclaras et al, (2009).

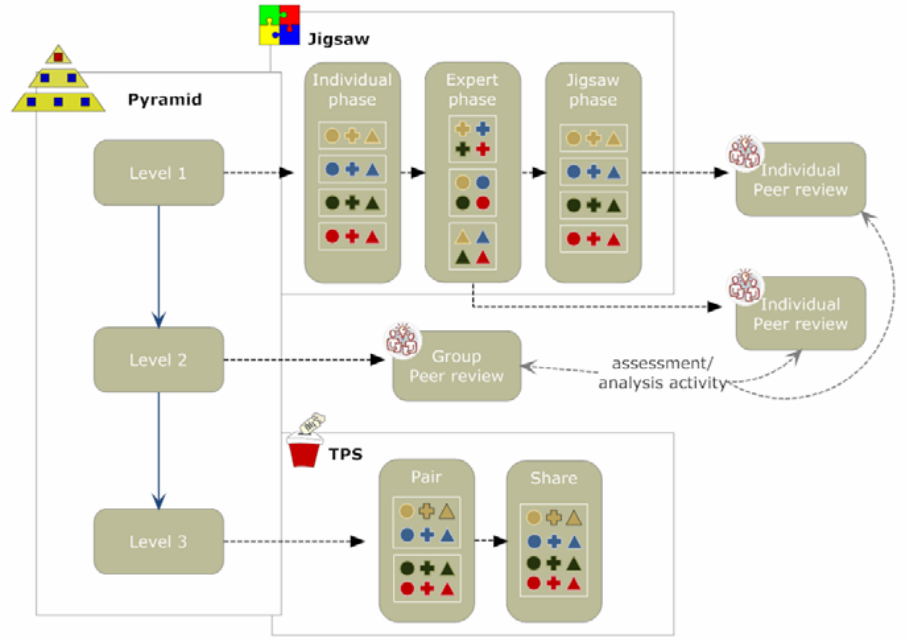


Fig.3. An example of co-design that takes into account learning and assessment activities

An advantage of this approach is that the information given in the relations between learning activities and analysis activities helps to get a harmonized and coherent process model immediately, because inconsistencies would be visible to the designer automatically. Drawbacks of this co-design approach are on the one hand that the expertise in both fields is needed at once, so a division of labor is hard to achieve, and on the other hand, that the methodology of design is “closed” in that respect, that other approaches for learning design or analysis cannot be combined with it.

In the multi-perspective approach the learning process and the analysis process are modeled separately with a method of the designers' choice, where each aspect can be designed by a different expert of the respective field. This potential of using division of labor between experts in the different aspects and also allowing each expert to use a method of her/his choice is a substantial advantage of this approach. The main challenge with this approach is the integration of the learning and analysis processes into one model that takes into account both perspectives appropriately. For this end, high-level constraints between the two perspectives are needed to allow a meaningful integration of these. This gap between the two different perspectives can be bridged by an abstraction level for learning tools and analysis tools, so that no concrete tools are defined in the respective processes: when both the learning activities and the analysis activities are modeled with their abstract purpose / goals, a suitable combination of learning and analysis tools can be searched for using a categorical representation of tools. One representative of this approach is the *OntoolCole/Ontoolsearch* (Vega-Gorgojo et al., 2008) environment, where an ontology of learning tools helps to categorize specific tools according to their general purpose. The extension with a similar categorization schema for analysis tools would help to inform the designers of a multi-perspective modeling approach if their modeled design can be conducted with the available selection of learning and analysis tools and what a recommended combination could be.

As a reflection on the *Mosaic* example from the previous section, one of the problems for analysis was the lack of logfile information with respect to the actor of an action. Using the *OntoolCole/Ontoolsearch* approach a designer would specify "collaborative concept mapping tool" and "tool for observation of collaboration" as required for the experience. Table 3 gives a simple list for several tools available for the activity that could be provided by *OntoolSearch*:

Table #-3. Selection of appropriate tools for the Mosaic experience based on categorization of learning and analysis tools

Tool	Category	Format produced	Format consumed
Cmaptools+VNC	Collaborative concept mapping	VNC log [without user info]	
FreeStyler + concept map plug-in	Collaborative concept mapping	Common format logs [with user info]	
.LRN monitoring	Observation of student progress		IMS Learning Design
Argunaut system	Observation of collaborative activities		Common format logs

Based on the categories assigned to available tools, a possible recommendation would be the collaboration tool *FreeStyler* with its concept mapping functionality and logfile information in the *common format* defined in the Kaleidoscope IA project, and the *Argunaut* observation tool that consumes logfiles in this *common format*. The combination of the tools would be a recommendation, because the tools not only comply to the specified categories (this would also fit for the combination of *Cmaptools/VNC* and *.LRN*) but also with respect to the logfile format, which facilitates the integration of both into an integrated learning/analysis process (which was not possible in the *Mosaic* experience with the used tools).

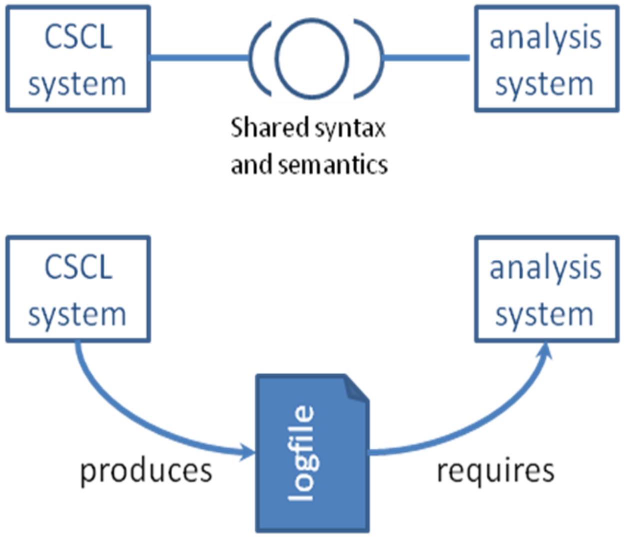
Besides this challenge how to integrate the two different perspectives, the latter multi-perspective approach has the benefit that each perspective can be modeled separately by an expert in that field using the design method of her/his choice, i.e. the flexible combination of different methods for the learning design and the analysis process is possible, which would also make comparative analyses feasible, e.g. using two different analyses processes with the same learning design method or vice versa.

4.3 Technology-driven solutions

Technological infrastructure can be a show-stopper or enabler of analysis processes in computer-based learning scenarios. As motivated in the previous section, the analysis process can be severely compromised by the technology used, if important data is missing (such as in the *Mosaic* example), heterogeneous data cannot be synchronized and/or integrated etc.

Yet, there are several provisions that can be made to prepare the infrastructure for a support of both learning and analysis processes: **interoperability** is the main principle that allows conducting learning and analysis processes in a coordinated manner. Besides some systems that have tightly-coupled learning tools and analysis tools (e.g. the *Synergo* system (Avouris et al., 2004)) that has a built-in analysis tool for the teacher/researcher), usually the learning tools and the analysis tools are not within the same codebase and developed by the same teams. Thus, data exchange and semantic interoperability between the different tools are a prerequisite to conduct the analysis process.

Fig. 4. Schema of the two decoupled architectures. On top of the diagram the inter-process communication via communication of shared syntax and semantics. At the bottom the coupling via logfiles



The idea of having independence between the CSCL and IA-related codebases leads to **decoupled architectures**, where both systems are able to run independently of each other, but where the semantics and syntax of the interactions are shared, so that the analysis processes can get the most of the solutions. We can distinguish two approaches. The first is based on inter-process communication, while the second is based on interchange of log file information. Figure 4 shows the schema for both types of decoupled architectures.Examples for the first approach have been proposed for educational systems that combine the functionality of different stand-alone applications. In Ritter & Koedinger (1997) the combination of simple learning tools with the diagnostic capabilities and feedback messages of Cognitive Tutors has been discussed, which opens up the possibility of on-the-fly analysis and tutoring feedback for the integrated system. Service-oriented approaches are well-suited for this type of integration, because a (web) service provides a well-defined interface description for the data exchange and its granularity is well-suited to compose complex educational applications from several services. The use of (web) services for educational systems has been discussed early in Chen (2003) and Vaquero-González et al. (2005) and has been followed up by several implementations of educational systems, such as GridCole (Bote-Lorenzo et al., 2008) and Finesse (Allison et al., 2005). Currently, these systems still have a limited scope, because the number of existing learning services and especially of analysis services is too low to allow a variable mix-and-match of services for flexible construction of learning scenarios. Given the expected larger set of educational services, the instantiation of services for learning scenarios via graphical editors is a promising approach to relieve the scenario designers of deep technical knowledge. A similar approach to this can be found in existing learning environments, like Moodle, that enables its users (mostly teachers) to instantiate specific tools, such as chats, wikis, etc.

Log files can also be the base of decoupled architectures, provided that the **semantics** of the log events are known and shared between the CSCL and the IA sub-systems. This would be a step forward in the use of log files, so that different IA tools would be able to be used with different CSCL systems and vice versa, providing for a flexible mix-and-match. To facilitate the flexible combination of different analysis tools during the process, the international initiatives ICALTS, IA, and CAViCoLA between several European research teams defined the standardized common data format that captures the relevant information of collaborative learning activities for follow-up analyses. A detailed description of this format is out of the scope of this chapter, but the interested reader can find it in Harrer et al., (2009). The *Argunaut* (De Groot, 2007) system is an example where the standardization of log files has been used as the mediating vehicle to allow moderators (e.g., teachers) of synchronous discussions the monitoring and evaluation of the ongoing discussion(s). Different discussion environments can be integrated into the system, if they provide the log file format as output (Harrer et al., 2008b). The technical framework of the *Argunaut* system is shown in Figure 5, where arbitrary discussion environments can be integrated into the system, given that they comply with the defined logfile format (used by the Protocol Processor) and implement the desired moderation features (Remote Intervention API) to allow the moderator of e-discussions the intervention into ongoing discussions. Indeed the framework provides a solution that uses inter-process communication via the proxies to allow on-the-fly observation and intervention of discussions based on the exchange of common format events in the well-defined logfile format.

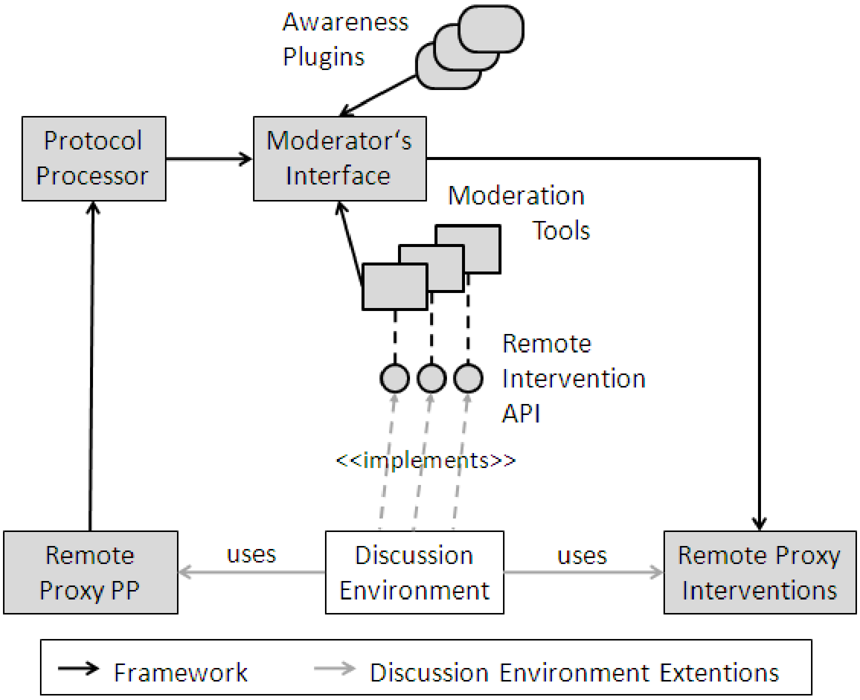


Fig. **5.** Argunaut framework, showing how different discussion environments can be plugged into the environment

A generic processing scheme for analysis has been enabled due to this common data format: Its validity has been tested through several CSCL and IA tools within the Kaleidoscope network (Kaleidoscope, 2007), including heterogeneous indicators based on Social Network Analysis (SNA) or qualitative methods. Other initiatives in this direction are the MULCE project, that aims to define a learning data corpora for sharing purposes (Chanier et al., 2009), and the *Centralised Research Data Repository*, another Kaleidoscope initiative, which aimed to define a common ontology to share learning materials among researchers (Centralized Research Data Repository, 2007).

Because in some cases the modification of the original logfile formats is not desirable or brings a substantial effort with it, the use of adapter components is a potential technical solution for this problem: with the help of adapters the data sources / logs can be made compatible with each other so that analysis tools can be used with learning tools of a third party. The practical usage of adapters in heterogeneous educational scenarios (i.e. using several independent learning tools) has been demonstrated and discussed recently in Harrer et al. (2008a). The effort needed to generate an adapter component is relatively low compared to refactoring tools into full-fledged web or grid-services, which makes this proposal a good alternative for initial rapid development. The use of a mediator was also proposed by the aforementioned *centralized data repository* initiative, in order to enable tools with different underlying ontologies to access a **common ontology** representing a wide range of possible learning objects.

In spite of their partial success, these proposals still represent local efforts, and show that there is a need for an agreement by the various stakeholders of the community that would allow for a generalized sharing of tools and data among teams.

5 Conclusions

Computer-supported interaction analysis tools and methods have the potential to leverage research and practice in CSCL. This fact has raised the interest of the research community, which has been reflected in a growing number of research projects, meetings and papers focused on these themes. However, current practice is not benefiting from the potential advantages of applying IA tools to their settings. Among the reasons that explain this mismatch we can identify problems at the design level, where end users such as researchers or practitioners do not plan in advance for configurations that allow for interaction analysis. Sometimes, these problems are related to issues at the application and architectural levels, where (learning) system developers do not provide for ready-to-use interaction data or if they do, they do not worry about their interoperability or synchronization with other sources of data.

Several lines of work can help to overcome these problems. At the design level, interaction analysis issues must be integrated in the overall design process. This can be done following a co-design approach or a multiple perspectives approach, taking into account the trade-off between a consistency control and a division of labor at design time. At the application and architecture levels, technology-driven solutions based on decoupled architectures are feasible. These architectures can be implemented following an inter-process communication or a log-file interchange approach, aiming at enhancing interoperability while fostering integrated use of CSCL and IA tools.

The problems stated in this chapter reflect a large variety of situations in both European and national projects, and heterogeneous or homogeneous design teams. These problems are expected to be even worse in the case of a wider adoption of the IA tools and techniques by practitioners. Then, this review aims to raise the awareness of all the implied actors on the issues that must be taken into account to increase the use of IA tools in real CSCL settings. This is a noteworthy effort; as it would allow researchers and practitioners improve their experiences by being able to reflect on them and by adding new monitoring and assessing capabilities to their CSCL settings.

However, several issues have been detected that remain unsolved, and call for further efforts in the area. First of all, we need to increase interoperability among our data, but the complexity and richness of CSCL settings make it difficult to reach a common agreement on these data. This interoperability might be achieved following different paths, from minimalist approaches that define the minimum set of data needed by tools to inter-operate, or by means of practical approaches that propose the use of adapters to match local formats (enabling for specific types of analysis) to a generic one (enabling for sharing data and tools). Of course, these are not easy approaches and many issues remain unsolved that need further discussion and agreements among the community which has been started in recent years with several international workshops and initiatives on analysis methods and integration of methods with different tools (Dwyer et al., 2008; Law et al.,2009)

Improvements at the technological level are also needed. Especially, there is a need to increase the number of IA tools and services that could then be chosen by researchers and practitioners in the interaction-aware design processes outlined in this chapter. A major adoption of these design processes could also be benefited by tools, such as the one proposed by Villasclaras et al. (2009), for integrating assessment into the learning design processes.

Acknowledgments

The authors would like to thank their colleagues for their intensive cooperation carrying out the cases mentioned in this chapter. This work has been partially supported by the Spanish Ministry of Science and Education (project TIN2008-03023/TSI) and the autonomous government of Castilla and León, Spain (VA106A08, VA107A08).

References

.LRN (2008). http://dotlrn.org/ Accessed 23 August 2009.

Allison, C., Nicoll, J. R., Bateman, M., & Ling, B. (2005). From Web To Grid: Evolution of a Learning Environment. In *The 4th IASTED International Conference on Web Based Education WBE'05*. 2005. Grindelwald, Switzerland: ACTA Press.

Avouris, N., Margaritis, M., & Komis, V. (2004). Modelling interaction during small-group synchronous problem-solving activities: The Synergo approach. In *2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction at ITS 2004, 7th Conference on Intelligent Tutoring Systems*. Maceio, Brazil.

Bote-Lorenzo, M.L., Gómez-Sánchez, E., Vega-Gorgojo, G., Dimitriadis, Y., Asensio-Pérez, J.I., Jorrín-Abellán, I.M. (2008) Gridcole: A tailorable grid service based system that supports scripted collaborative learning Computers & Education. 51(1):155-172.

Botturi, L., & Stubbs, S. T. (Eds.). (2007). *Handbook of Visual Languages for Instructional Design: Theories and Practices*. Hershey, PA: Idea Group Inc (IGI).

Bratitsis, T., Dimitracopoulou, A., Martínez-Monés, A., Marcos-García, J. A., & Dimitriadis, Y. (2008). Supporting members of a learning community using Interaction Analysis tools: The example of the Kaleidoscope NoE scientific network. In *8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008), Santander, Spain:* IEEE Computer Society.

Centralized Research Data Repository (2007). http://www.noe-kaleidoscope.org/group/datarep/ Accessed 23 August 2009.

Chanier, T., Reffay, C., Lamy, M. N., & Ganascia, J. G. (2009). Project MULCE (Multimodal Learning Corpus Exchange) http://mulce.univ-fcomte.fr/. Accessed 23 August 2009.

Chen, W. (2003). *Web Services in Web-based Education Systems*. In *Proc. of Computers and Advanced Technology in Education CATE/WBE ' 2003*. Rhodes, Greece.

De Groot, R., et al. (2007). Computer supported moderation of e-discussions: the ARGUNAUT approach. In C. Chinn, G. Erkens, & S. Puntambekar (Eds.) *Computer- Supported Collaborative Learning: Mice, Minds, and Society. Proceedings of the Seventh International Computer Supported Collaborative Learning Conference (CSCL 2007)*. Mahwah, NJ: Lawrence Erlbaum Associates.

Dimitracopoulou, A. (2005). Designing collaborative environments: Current Trends and Future Research Agenda. In T. Koschmann, T.-W. Chan & D. D. Suthers (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years!* Taipei, Taiwan: Lawrence Erlbaum Associates.

Dimitriadis, Y., Asensio-Pérez, J. I., Hernández-Leo, D., Roschelle, J., Brecht, J., & Tatar, D. (2007). *From socially-mediated to technology-mediated coordination: A study of design tensions using Group Scribbles.* In C. Chinn, G. Erkens, & S. Puntambekar (Eds.) *Computer-Supported Collaborative Learning: Mice, Minds, and Society. Proceedings of the Seventh International Computer Supported Collaborative Learning Conference (CSCL 2007).* Mahwah, NJ: Lawrence Erlbaum Associates.

Dyke, G. (2008). Tatiana: Trace Analysis tool for interaction ANAlysts, http://tatiana.emse.fr/Presentation/summary.html. Accessed 23 August 2009.

Dwyer, N., Law, N., Rose, C.P., Suthers, D. (2008). Workshop A Common Framework for CSCL Interaction Analysis at ICLS 2008, Utrecht, The Netherlands. http://engaged.hnlc.org/stories. Accessed 23 September 2009.

Fiotakis, G., Fidas, C., & Avouris, N. (2007). *Comparative usability evaluation of web systems through ActivityLens*. In *Proc. of the 11th Panhellenic Conference on Informatics (PCI 2007), Patras, Greece*.

Fuente, L. d. l., Pardo, A., Delgado, C., & Asensio, J. I. (2008). Collaborative Learning Models on Distance Scenarios: A Case Study. In *8th IEEE International Conference on Advanced Learning Technologies, ICALT 2008*. Santander, Spain: IEEE Computer Society.

Guribye, F., & Wasson, B. (2002). *The ethnography of distributed collaborative learning*. Paper presented at the Computer Supported Collaborative Learning 2002.

Harrer, A., Kahrimanis, G., Zeini, S., Bollen, L., & Avouris, N. (2006). Is there a way to e-Bologna? Cross-National Collaborative Activities in University Courses. In *Proc. of the 1st European Conference on Technology Enhanced Learning (EC-TEL)*, Crete, Greece.

Harrer, A., Zeini, S., Kahrimanis, G., Avouris, N., Marcos, J. A., Martínez-Mones, A., et al. (2007). Towards a Flexible Model for Computer-based Analysis and Visualization of Collaborative Learning Activities. In C. Chinn, G. Erkens, & S. Puntambekar (Eds.) *Computer-Supported Collaborative Learning: Mice, Minds, and Society. Proceedings of the Seventh International Computer Supported Collaborative Learning Conference (CSCL 2007)*. Mahwah, NJ: Lawrence Erlbaum Associates.

Harrer, A., Pinkwart, N., McLaren, B. M., & Scheuer, O. (2008a). The Scalable Adapter Design Pattern: Enabling Interoperability Between Educational Software Tools. *IEEE Trans. Learn. Technol., 1*(2), 131-143.

Harrer, A., Ziebarth, S., Giemza, A., & Hoppe, U. (2008b). A framework to support monitoring and moderation of e-discussions with heterogeneous discussion tools. In *Proc. of the 8th IEEE International Conference on Advanced Learning Technologies ICALT 2008,* Santander, Spain: IEEE Computer Society.

Harrer, A., Martínez-Monés, A., & Dimitracopoulou, A. (2009). Users' data: Collaborative and social analysis. In N. Balacheff, S. Ludvigsen, T. de Jong, A. Lazonder & S. Barnes (Eds.), *Technology-Enhanced Learning. Principles and products*. UK: Springer.

Kaleidoscope (2007). Kaleidoscope NoE. Projects: ICALTS (Interaction & Collaboration Analysis' supporting Teachers & Students' Self-regulation) and IA (Interaction Analysis' supporting Teachers & Students' Self-regulation), and the European Research Team CAViCoLA

(Computer-based Analysis and Visualisation of Collaborative Learning Activities), from http://www.noekaleidoscope.org/. Accessed 23 August 2009.

Law, N., Suthers, D., Lund, K., Rosé, C., and Teplovs, C., (2009). *Common Objects for Productive Multivocality in Analysis*, Workshop at the Computer Supported Collaborative Learning (CSCL) Conference 2009, Rhodes, Greece.

Lohmann, S., Thalmann, S., Harrer, A., & Maier, R. (2008). *Learner-Driven Annotation of Learning Resources - Lessons from Experiments on Tagging*. In Proc. of the 8th International Conference on Knowledge Management I-KNOW 08, *Graz, Austria.*

Markauskaite, L., & Reimann, P. (2008). Enhancing and Scaling-up Design-based Research: The potential for E-Research. In *Proc. of the International Conference of Learning Sciences, ICLS 2008*. Utrecht, The Netherlands.

Martínez, A., Dimitriadis, Y., Rubia, B., Gómez, E., & de la Fuente, P. (2003). Combining qualitative evaluation and social network analysis for the study of classroom social interactions. *Computers and Education, 41*(4), 353-368.

Martínez, A., Harrer, A., Barros, B., & Vélez, J. (2005). *Library of Interaction Analysis Methods*: Kaleidoscope NoE.

Martínez-Monés, A., Dimitriadis, Y., & Harrer, A. (2008a). Interaction-aware design for learning applications. Reflections from the CSCL field. In *Proc. of the 8th IEEE International Conference on Advanced Learning Technologies, ICALT 2008*. Santander, Spain: IEEE Computer Society.

Martínez-Monés, A., Villagrá-Sobrino, S., Santos-Fernández, R., Anguita-Martínez, R., & Jorrín-Abellán, I. M. (2008b). Social network analysis support for an IBL wiki-based course. Paper presented at the *Workshop Real-Time methods at International Conference of the Learning Sciences, ICLS 2008*. Utrecht, The Netherlands.

Moreno, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications, 33*, 135-146.

Ritter, S., & Koedinger, K. (1997). An Architecture for Plug-In Tutor Agents. *Int. J. of Artificial Intelligence in Education, 7*(3/4), 315-347.

Roschelle, J., Tatar, D., Chaudhury, S. R., Dimitriadis, Y., Patton, C., & DiGiano, C. (2007). Ink, improvisation, and interactive engagement: Learning with tablets. *Computer, 40*(9), 38-44.

Soller, A., Martínez, A., Jermann, P., & Muehlenbrock, M. (2005). From Mirroring to Guiding: A Review of the State of the Art Technology for Supporting Collaborative Learning. *International Journal on Artificial Intelligence in Education, 15*, 261-290.

Vaquero-González, L. M., Hernández-Leo, D., Simmross-Wattenberg, F., Bote-Lorenzo, M. L., Asensio-Pérez, J. I., Dimitriadis, Y., et al. (2005). The Opportunity of Grid Services for CSCL-Application Development. In *Proc. of the 13th Euromicro Conference on Parallel, Distributed and Network-based Processing, PDP'05*. Lugano, Switzerland.

Vega-Gorgojo, G., Bote-Lorenzo, M.L., Gómez-Sánchez, E., Asensio-Pérez, J.I., Dimitriadis, Y., and Jorrín-Abellán, I.M. (2008) Ontoolcole: Supporting Educators in the Semantic Search of CSCL Tools**,** *Journal of Universal Computer Science (JUCS)*. 14(1):27-58.

Villasclaras, E., Hernández, D., Asensio, J. I., & Dimitriadis, Y. A. (2009). Incorporating assessment in a pattern-based design process for CSCL scripts. *Computers in Human Behavior, 25(5)*, 1028-1039.