Metadata of the article that will be visualized in Online

1	Article Title	**A three-level analysis of collaborative learning in dual-interaction spaces**	
2	Article Sub- Title		
3	Article Copyright - Year	**International Society of the Learning Sciences, Inc.; Springer Science + Business Media, LLC 2009 (This will be the copyright line in the final PDF)**	
4	Journal Name	International Journal of Computer-Supported Collaborative Learning	
5	Corresponding Author	Family Name	**Lonchamp**
6		Particle	
7		Given Name	**Jacques**
8		Suffix	
9		Organization	LORIA-Nancy Université
10		Division	
11		Address	BP239, Vandœuvre-lès-Nancy Cedex 54506, France
12		e-mail	jloncham@loria.fr
13	Schedule	Received	6 September 2008
14		Revised	
15		Accepted	28 May 2009
16	Abstract	CSCL systems which follow the dual-interaction spaces paradigm support the synchronous construction and discussion of shared artifacts by distributed or colocated small groups of learners. The most recent generic dual- interaction space environments, either model based or component based, can be deeply customized by teachers for supporting different collaborative learning tasks and different ways of performing them. This work stresses the importance of basing customization decisions on a socio-cognitive interpretation of how learners interact in a given learning situation. The central contribution of this article is a methodological approach for conducting qualitative interaction analysis oriented toward the improvement of the supporting environment which can be applied to any learning task and any environment configuration. This “generic analysis approach” is organized into three levels. At the dialog level, a task-independent dialogical model is proposed for analyzing action/communication traces as “generalized conversations.” A graphical notation is provided for visualizing the syntactical characteristics of collaborative sessions. At the knowledge level, a typology of task-independent collaborative knowledge-building episode types that can occur during such generalized conversations is proposed. Thanks to that classification scheme, recurrent meaningful elements that structure the low-level descriptions can be detected and characterized. These regularities help to pass from local interpretations to a global interpretation of the whole process. At the action level, task-dependent socio-cognitive interpretations of why the collaborative learning process unfolds as observed are proposed. They constitute a firm basis for improving the customization of the generic environment in order to support learners more efficiently.	
17	Keywords separated by ' - '	Dual-interaction spaces - Interaction analysis - Generic environment - Generic analysis approach	
18	Foot note information		

JrnlID 11412_ArtID 9068_Proof# 1 - 30/05/2009

Computer-Supported Collaborative Learning
DOI 10.1007/s11412-009-9068-6

A three-level analysis of collaborative learning in dual-interaction spaces

Jacques Lonchamp

Received: 6 September 2008 / Accepted: 28 May 2009

Abstract CSCL systems which follow the dual-interaction spaces paradigm support the synchronous construction and discussion of shared artifacts by distributed or colocated small groups of learners. The most recent generic dual- interaction space environments, either model based or component based, can be deeply customized by teachers for supporting different collaborative learning tasks and different ways of performing them. This work stresses the importance of basing customization decisions on a socio-cognitive interpretation of how learners interact in a given learning situation. The central contribution of this article is a methodological approach for conducting qualitative interaction analysis oriented toward the improvement of the supporting environment which can be applied to any learning task and any environment configuration. This "generic analysis approach" is organized into three levels. At the dialog level, a task-independent dialogical model is proposed for analyzing action/communication traces as "generalized conversations." A graphical notation is provided for visualizing the syntactical characteristics of collaborative sessions. At the knowledge level, a typology of task-independent collaborative knowledge-building episode types that can occur during such generalized conversations is proposed. Thanks to that classification scheme, recurrent meaningful elements that structure the low-level descriptions can be detected and characterized. These regularities help to pass from local interpretations to a global interpretation of the whole process. At the action level, task-dependent socio-cognitive interpretations of why the collaborative learning process unfolds as observed are proposed. They constitute a firm basis for improving the customization of the generic environment in order to support learners more efficiently.

Keywords Dual-interaction spaces · Interaction analysis · Generic environment · Generic analysis approach

J. Lonchamp (✉)
LORIA-Nancy Université, BP239, 54506 Vandœuvre-lès-Nancy Cedex, France
e-mail: jloncham@loria.fr

Q1

JrnlID 11412_ArtID 9068_Proof# 1 - 30/05/2009

Introduction

The research described in this article considers a specific usage scenario where a small group of learners, either colocated or remote, are simultaneously connected to a CSCL system which provides a shared workspace and synchronous communication tools. A wide range of learning activities and pedagogical methods can take place in this usage scenario. It is, therefore, important to provide teachers with a system that can match the needs of different learning situations and pedagogical approaches. Moreover, teachers' and learners' needs evolve over time. This means that the system should be able to adapt to different situations already existing, but should also be able to evolve to fit new needs and expectations. One way to provide this kind of flexibility is *end-user deep customization,* that is, a customization which impacts the most important aspects of how learners are supported by the system such as the kinds of artifacts they share, the way they can interact, and the process that is enforced. Making such deep customizations based on a priori preferences, rules of thumb, or surface analysis of how learners interact may produce counterproductive effects. Dillenbourg (2002) emphasizes in particular the danger of disturbing "natural" problem-solving processes and "natural" interactions, and the danger of artificially "didactising" interactions. This work stresses the importance of basing enhancement decisions on a socio-cognitive interpretation of how learners interact in a given learning situation. The central contribution of this article is a *methodological approach for conducting qualitative interaction analysis oriented toward the improvement of the supporting system*. This is in contrast to most other interaction analysis approaches which consider confined research questions or hypothesis. The proposal restricts its scope to the specific usage scenario described above. The methodological approach can be applied to any learning activity and pedagogical organization that take place in that scenario. The remainder of this introductory section details the context of the research, the problem, some related works, and outlines the proposed approach.

The context

Dual-interaction spaces (DIS) environments are CSCL systems which combine an action space, including one or several tools such as shared diagram editors, text editors and whiteboards, and a communication space, including generally a chat tool for quasi-synchronous textual communication among participants (Mühlpfordt and Stahl 2007). DIS environments support the synchronous construction and discussion of shared artifacts by distributed (same-time/different-places) or colocated (same-time/same-place) small groups of learners. Most existing DIS environments are devoted to a *single collaborative learning task* and *a single way to perform it* which drastically diminishes their reusability. DIS environments may be used for many different tasks such as collaboratively completing design activities (e.g., Baker and Lund 1996; Zumbach et al. 2002; Dimitracopoulou and Komis 2005), working together with simulations (e.g., Landsman and Alterman 2003; Jermann 2004; van Joolingen et al. 2005), collaboratively exploring a space of debate (e.g., Amelsvoort et al. 2008) or solving math problems (e.g., Cakir et al. 2007). For each task, a wide variety of pedagogical scenarios may be followed. This article considers as its central example the task of collaboratively building software engineering representations, such as use case diagrams, class diagrams, entity-relationship diagrams, or Petri nets. There are many ways to perform that task in a learning setting due to the subject, the context, and the pedagogical preferences of teachers. First, it is possible to divide the process into separate phases. For instance, in the case of a class diagram, one can find: (1) A three-step process including a phase for defining candidate classes from the problem description, a phase for structuring the class diagram, and a phase for verifying the completeness and correctness of the representation with regard to the problem definition (Alonso et al. 2008). (2) A "pyramid process" with several parallel design phases by subgroups of learners that join progressively into larger groups for merging their proposed class diagrams until reaching a common solution (Hernández-Leo et al. 2005). (3) A process reflecting the underlying UML philosophy, with a "problem analysis phase" where use cases are written, followed by a "design phase" where use cases are translated into collaboration diagrams and the classes included in these collaboration diagrams are integrated into the class diagram. Second, it is possible to customize the formalism manipulated by the learners. A frequent strategy is to begin with a simplified version of the formalism and to progressively introduce additional modeling concepts. For example, start with only class names and member names, then enrich the formalism with associations between classes, multiplicities, and roles, and finally add inheritance relationships. Third, there are many ways to structure and control interaction during each phase through floor control (e.g., Glassner and Schwarz 2005), distribution of roles among learners like distinguishing between "analyst" and "critic" roles (e.g., Weinberger et al. 2005), message openers (e.g., Baker and Lund 1996), or fully-fledged interaction protocols (e.g., Pfister and Mühlpfordt 2002). Until now, all existing DIS environments devoted to software engineering representation construction only support a single representation and a single process (e.g., Baghaei and Mitrovic 2006; Soller et al. 2002; Avouris et al. 2004; Constantino-González and Suthers 2001).

The research described in this article is part of the Omega+ project which aims at developing a generic and flexible CSCL infrastructure for creating DIS environments supporting *different collaborative learning tasks and different ways of performing these tasks* (Lonchamp 2006). A generic environment is a system which is *deeply customizable by its end users*. To achieve this goal, Omega+ is implemented as a reflective system, that is, a system which includes an explicit representation (model) of the supported activity. End users can customize the system for a specific learning task before the beginning of the learning session by providing a dedicated model. They can also evolve the system during learning sessions by modifying the model: The behavior of a reflective system depends on that (continuously queried) representation and changes when it is modified, thanks to the causal relationship which is implemented between the representation and the system behavior (Maes 1987). The solution explored in Omega+ associates four separate (sub-) models to the different facets of collaborative learning activities (Dillenbourg 1999): process model, interaction model, artifact (meta) model, and "effects model." The later model specifies how to monitor and measure the effects of collaborative learning (Lonchamp 2008). This multi-model approach makes possible to build the activity representation at different levels of abstraction, adapted to the skills and needs of different categories of end users by: (1) Just reusing existing models. (2) Building new combinations of existing sub-models (i.e., following a very high-level configuration process). (3) Defining or customizing simple sub-models through high-level visual languages. (4) Developing complex sub-models through low-level specification and programming languages (Lonchamp 2006). Omega+ reflexive kernel allows building client/server DIS environments tailored for particular synchronous collaborative learning situations and learning processes through these four (sub-) models. The kernel also provides a set of tools (chat, shared text editor, shared whiteboard, generic shared diagram editor) and several optional mechanisms for floor control (Lonchamp 2007a), explicit referencing (Lonchamp 2007b), monitoring (Lonchamp 2008), and collaborative session history browsing, that can be selected and parameterized. Omega+ can be compared with other generic synchronous CSCL systems, either *model based* by the means of artifact models (e.g., Pinkwart 2003; Fidas et al. 2002), process models (e.g., Farnham et al. 2000), protocol models (e.g., Gogoulou et al. 2005), or *component based* (e.g., Landsman and Alterman 2003). CoFFEE (www.coffee-soft.org) is an example of an open source deeply customizable synchronous CSCL environment which has been widely tested in schools, colleges, and universities (De Chiara et al. 2007).

The problem

When they use a generic infrastructure end users become *co-designers of an "under-designed system"* (Fischer 2003). This approach is assumed to be efficient because end users are experts in their own field. In the case of CSCL systems, these co-designers must also deal with the way learners will actually use the system according to their purposes. The "indirect design" concept (Jones et al. 2006) captures the idea that design must be thought of as a means to influence learner activity, and this activity must be taken into consideration *as it happens*, *and not as it was predicted by designers*. Therefore, as emphasized at the beginning of this article, the deep customization of a generic environment should be based on a socio-cognitive interpretation of how learners interact in a given learning situation. This article aims at providing end users with a methodological approach for conducting qualitative interaction analysis oriented toward the customization and improvement of the supporting environment. This methodological approach represents an important component of the global design of the generic environment and may be understood as *a counterpart to the fact that the system is initially under-designed*. It is worth noting that the term "end user" can designate educational technology providers or researchers as well as teachers. As previously emphasized, teachers can just reuse existing models, combine sub-models, or perform simple customizations at the interface level. In this case, the burden of analyzing learning sessions for properly customizing the environment for a given learning task will be on the shoulders of experienced designers and analysts such as researchers or technology providers. The proposed methodological approach must cope with the following two constraints: (1) The target system follows the DIS paradigm which adds a level of complexity in comparison with more simple text-based CSCL systems; (2) The analysis approach should be "generic," that is, independent of the learning task, of the process that is enforced, of the roles that learners play, of the artifacts that learners manipulate, of the message types that learners exchange, and of the protocols that are enforced.

The kind of interaction analysis that is required aims at explaining the "situated process by which participants accomplish learning" (Suthers et al. 2007). There exists obviously a great tension between the requirement of being "generic" and the requirement of deeply analyzing how learners behave in a given situation. For dealing with that issue, the proposed approach distinguishes between *different levels and dimensions of analysis*. What is fully "generic" (task independent) at the finest-grained level is the way learners interact with each other by exchanging messages and manipulating the shared artifacts. At a coarser-grained level, the collaborative knowledge-building process can be characterized to a certain extent in generic terms such as "argumentation," "negotiation," "clarification," and the like. However, the final interpretation of the overall process by which participants accomplish learning is, by essence, task dependent. As we will see later, these three dimensions and granularities are at the core of the proposed approach.

Related works

The characteristics emphasized in the previous subsection differentiate the present work from the main stream of quantitative interaction analysis works (De Wever et al. 2006; Strijbos et al. 2006). In the quantitative approach, the communication is coded, summarized, and frequencies are used for comparisons or statistical testing. Statistical comparisons require a precise hypothesis derived from theory formulated in advance. At the opposite extreme, the qualitative view, which is adopted in this work, requires less a priori expectations or even none (Strijbos et al. 2006).

The few existing qualitative interaction analysis approaches for CSCL environments cannot be directly reused. Most of them only consider asynchronous or synchronous textual interaction (forums and chat tools) and do not take into account the multiplicity of media resources (Suthers et al. 2007). Some others are founded on task-related concepts, such as "math proposal adjacency pairs" (Stahl 2007), "pivotal contributions" and "reasoning stages" (Wee and Looi 2007), or rely on the existence of specific mechanisms like explicit referencing mechanisms (Trausan-Matu et al. 2006).

However, several ideas in different works have strongly influenced the present proposal. One case, in particular, is Suthers' intersubjective meaning-making analysis (2006). This begins by identifying "uptake acts" in which one participant takes up another's contribution and does something further with it. It continues by representing the resulting collection of uptake relations as a directed acyclic graph that gathers distributed data together into a single analytic artifact and finally tries to recognize what the participants have accomplished through sequences or compositions of uptakes. The present proposal also follows the idea of having multiple levels and dimensions of analysis (e.g., Henri 1992; Schrire 2006; Strijbos and Stahl 2007) and the idea of associating a specific unit of analysis to each level of analysis (e.g., Schrire 2006).

The proposed approach

According to the theory of signs (Morris 1938), it is possible to analyze any kind of communication at three levels: syntax, semantics, and pragmatics. Syntax is about form and specifies what the components and the structure are, and how to decompose them. Semantics is about meaning and considers context-independent properties. Pragmatics is about use and considers practical aspects depending on the context and participants' objectives. The proposed generic approach for analyzing collaborative learning in DIS follows a *bottom-up strategy* organized into three levels that can be related in essence to these syntactic, semantic, and pragmatic levels: (1) The first level is called the *dialog level*. A task-independent *dialogical model* is proposed for analyzing communication/action traces produced by DIS environments as "generalized conversations." The model takes into account the specificities of DIS systems such as the fact that tool actions are generally accompanied by textual messages in which learners explain their initiatives. Learners' composite contributions (including both tool actions and messages) are identified and structured into *generalized adjacency pairs*. A graphical notation is provided for visualizing the syntactical characteristics of generalized conversations. Small details revealed at this fine-grained analysis level often play an important role for elaborating higher level interpretations. (2) The second level is called the *knowledge level*. By adopting a *collaborative knowledge-building perspective,* it is possible to define a typology of task-independent episode types that can occur during generalized conversations. Each episode type specifies what the learners jointly accomplish from the knowledge-building perspective, such as clarifying, negotiating, planning, and so forth, and can trigger basic learning mechanisms. Thanks to that classification scheme, *recurrent meaningful elements* (interaction patterns or "hot spots") that structure the low-level descriptions can be detected by the analysts and characterized from the knowledge-building perspective. These regularities help *to pass from local interpretations to a global interpretation* of the whole process at the next level. (3) The third level is called the *action level*. Task-dependent interpretations of *why* the collaborative learning process unfolds as observed are proposed on the basis of the recurrent elements detected at the knowledge level. Such global interpretations can constitute a firm basis for improving the customization of the DIS environment in order to support learners more efficiently. *Methodological guidelines* are suggested for elaborating the interpretation and defining the concrete improvements that can be derived from that interpretation.

All the analysis process is carried out manually. Building a meaningful interpretation requires going deeply into the details of the interaction process in an open-minded way. The feasibility of automating some parts of the approach is discussed in the final conclusion. The next three sections of the article describe each analysis level in turn and illustrate the whole approach with an example of collaborative construction of a software engineering representation by a small group of learners using an Omega+-based DIS environment.

The dialog level

The approach

Communication/action traces are analyzed as "generalized conversations." The analyst has first to segment the trace into contributions, that is, meaningful sets of elementary actions performed without interruption by the same participant. In a DIS context, contributions can be composite, that is, distributed over the task space and the

Table 1 The proposed classification scheme

Q2

The proposed classification scheme	Communicative functions in Arjvala et al (2007)
Suggestion	Suggestive
Precision or clarification	Justificational, clarificational, elaborative
Evaluation	Evaluative (feedback), judgmental (agreement), counter argumentative (disagreement)
Question	Interrogative
Answer	Responsive
Mostly performed by tool actions	Informative (giving information or example)
Explanation of a tool action	-

communication space. For instance, tool actions are generally accompanied by textual messages in which learners explain their initiatives. The segmentation procedure is based both on the delay between actions and their types. A message containing a question is generally a contribution by its own while a message explaining an action is a part of a composite contribution. So, as already emphasized by other researchers, segmentation and classification are difficult to dissociate (e.g., Beers et al. 2007). The proposed classification scheme of messages is derived from the analytical framework of language functions developed by Kumpulainen and Mutanen (1999) and adapted to collaborative knowledge construction by Arjvala et al. (2007). It has been further adapted *to the specific context of DIS systems* by taking into account tool actions, as shown in the last two lines of Table 1, and by merging some categories together. The existence of evaluative contributions is important for the subsequent knowledge level but their precise content (agreement, disagreement, or other kind of feedback) is not relevant.

Other less focused textual contribution types, corresponding to the social and personal communicative functions (e.g., salutation, jokes), are not considered in the proposed model, even if they are important for a successful collaboration (Rourke et al. 1999).

The structure of the generalized conversation becomes apparent when the analyst links contributions one to the other. Classical conversation analysis has defined descriptive units at different levels: turns, adjacency pairs, and complex sequences (Levinson 1983). A turn is a time during which a single participant speaks. An adjacency pair "consists of two ordered utterances, the first and second pair parts, produced by two different speakers. The two parts come in types that specify which is to come first and which second. The form and content of the second part depends on the type of the first part" (Clark and Schaefer 1989). For instance, an answer depends on a question or an acceptance depends on an offer. "Given a first pair part, a second pair part is conditionally relevant, that is, relevant and expectable, as the next utterance. Once A has asked a question, it is relevant and expectable for B to answer." Such paired turns can be components of larger sequences such as base pair completed with expansion sequences, stories, or topical trajectories (Schegloff 2006). The proposed dialogical model defines "*generalized adjacency pairs*" (GAPs) that are pairs of composite contributions. GAPs and classical adjacency pairs have some noticeable differences. First, there is a shift from temporal adjacency that underlies classical adjacency pairs, to the broader idea of *interactional and conceptual adjacency*. Second, the two contributions in a GAP are separated by a noticeable period of time necessary for producing the reaction. Third, the contributors in a GAP are not necessarily different learners like in classical adjacency pairs (inter-subjective reaction). Frequently, a GAP relates two contributions from the same participant (intra-subjective reaction)—for instance, when nobody else has reacted to the first part. However, both approaches have much in common and reflect "how mutual understanding is accomplished and displayed" (Hutchby and Wooffitt 1998). The next section goes into the details of the dialogical model.

The dialogical model

Tool contributions with diagram editors, text editors, or whiteboards are classified into three categories: (1) *Additive contributions*: Add a component (node, link, or property) to a diagram (which generally includes several elementary actions called "addVertex," "addEdge," "newName," and "newProperties" in Omega+), add a piece of text (which generally includes several characters' insertions and possibly some deletions), or add some drawing (which generally includes several insertions of figures, lines, and texts). (2) *Change contributions*: Modify a diagram component, a text, or a drawing. For diagram components, move actions are only relevant in some contexts, when the spatial positioning of components is meaningful. (3) *Destructive contributions*: Suppress a component, a piece of text, a drawing.

As explained in the previous section, textual contributions (chat contributions) are classified into five categories: (1) *Suggestion* (proposing components/ properties, actions, ideas). (2) *Evaluation* (agreement, disagreement, other kind of feedback) of a previous action, suggestion, or evaluation from another learner. (3) *Explanation* by a learner of his/ her own actions. (4) *Precision or clarification* by a learner of his/her own suggestions or evaluations. (5) *Question*. (6) *Answer*.

A GAP encompasses two parts. The first part, called the "initiation part," is characterized by a set of contribution types S1. The second part, called the "reaction part," is characterized by a set of possible contribution types S2 with two optional attributes. GAP = ((S1), (S2, REL, ADD)). S1 can be empty (in the case of a spontaneous initiative) or can contain one or several contribution types taken from the set: {additive contribution, change contribution, destructive contribution, suggestion, evaluation, question, answer}. S2 contains one or several contributions types taken into {additive contribution, change contribution, destructive contribution, suggestion,

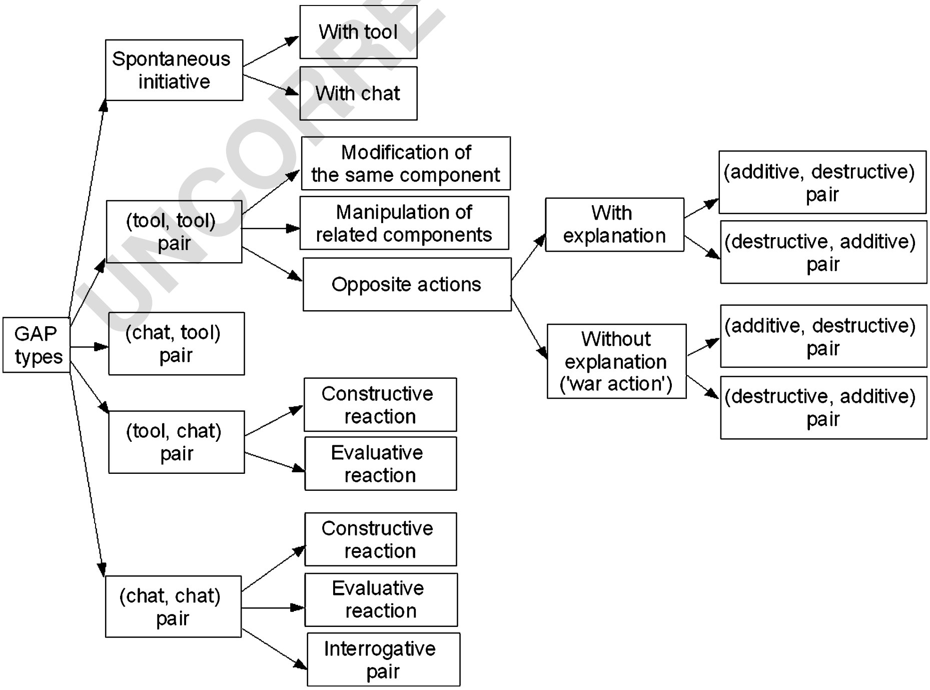


Fig. 1 GAP types structuring

Table 2 The taxonomy of GAP types

Q2

Type	Initiation part	Reaction part
1	S1 = ∅ (spontaneous initiative)	S2 = {additive/change/destructive contribution} ADD = explanation
2	S1 = ∅ (spontaneous initiative)	S2 = {suggestion, question} ADD = precision or rectification
3	S1 = {additive/change contribution}	S2 = {change contribution} REL = same tool & same component ADD = explanation
4	S1 = {additive/change/destructive contribution}	S2 = {additive/change/destructive contribution} REL = same tool & related components, different tools & related components ADD = explanation
5	S1 = {additive/change contribution}	S2 = {destructive contribution} REL = same tool & same component ADD = explanation
6	S1 = {destructive contribution}	S2 = {additive contribution} REL = same tool & same component ADD = explanation
7	S1 = {additive/change contribution}	S2 = {destructive contribution} REL = same tool & same component ADD = none
8	S1 = {destructive contribution }	CON = {additive contribution} REL = same tool & same component ADD = none
9	S1 = {suggestion, evaluation, answer}	S2 = {additive/change/destructive contribution} ADD = explanation
10	S1 = {additive/change/destructive contribution}	S2 = {suggestion, question} ADD = precision or rectification
11	S1 = {additive/change/destructive contribution}	S2 = {evaluation} ADD = precision or rectification
12	S1 = {suggestion, evaluation, answer}	S2 = {suggestion, question} ADD = precision or rectification
13	S1 = {suggestion, evaluation, answer}	S2 = {evaluation} ADD = precision or rectification
14	S1 = {question}	S2 = {answer, question, suggestion, evaluation}

evaluation, question, answer}. REL characterizes the type of relationship between the artifact component in the first part and the artifact component in the second part: Possible values are “same tool & same component,” “same tool & related components,” and “different tools & related components.” ADD defines the type of the accompanying textual contribution: Possible values are “explanation,” “precision,” “clarification.” The additional contribution can follow the main contribution S2 or it can come before announcing S2. The semantics is as follows: *Given a contribution whose type belongs to S1, a contribution whose type belongs to S2 is relevant and expectable. When the two contributions impact an artifact component, the type of relationship between these components is specified in the REL attribute. The second contribution can be accompanied by a textual contribution whose type is specified in the ADD attribute*. In a generalized conversation, some contributions are both the reaction part of a GAP and the initiation part of the following GAP(s).

A GAP *is mainly characterized by the properties of its paired contributions*. The proposed model distinguishes at a first level between spontaneous initiatives (with an empty S1) and the four possible combinations of tool and chat media: tool-tool pairs, chat-tool pairs, tool-chat pairs, and chat-chat pairs. These last four categories are refined in subcategories by taking into account the contribution types, REL and ADD attributes, as shown in Fig. 1. Table 2 summarizes the complete taxonomy of GAP types.

Some additional rules are needed to deal with ambiguous cases in the pairing process. The following rules are used in the example which is discussed in the next subsection. When several contributions evaluate the same preceding contribution (tool action, suggestion, another evaluation…) they should all be paired with that initial contribution rather than linked in the chronological order. When these evaluations result in a tool action or suggestion, the resulting contribution should be paired with the last preceding evaluation message (although there is no absolute evidence that all the evaluation messages have been taken into account for producing the resulting contribution). Similarly, when a contribution reacts to a sequence of contributions, it is paired with the last preceding one.

As previously emphasized, Omega+ environment kernel provides several optional mechanisms. It is worth noting that these mechanisms do not require specific notations. Explicit referencing mechanisms, such as graphical pointers or sticky notes, aim at facilitating the designation of specific elements in graphical artifacts. The analyst can always go back to regular textual messages including spatial deictic expressions. Floor control mechanisms impact the production of actions and messages. Their consequences are fully reflected in the proposed descriptive model.

The example

The study involved 24 French students enrolled in a second-year university course in computer science. Small groups of three students, randomly assigned to the groups, received small case descriptions and were asked to build use case diagrams during 30 to 45

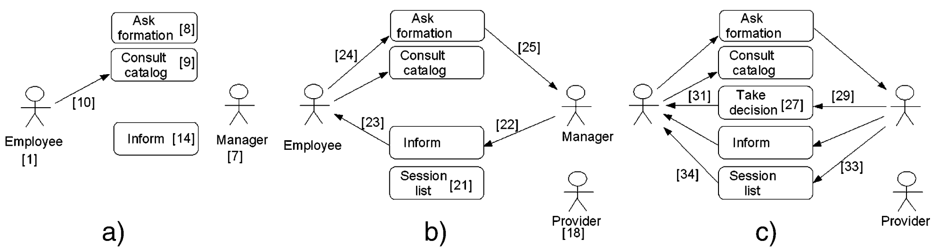


Fig. 2 The evolving artifact, annotated with GAP numbers

minutes long collaborative sessions with Omega+. Students were colocated (in the same classroom) but were not allowed to speak. Omega+ client was configured with a read-only text-board for the case description, a customized shared diagram editor, and a chat tool. Students had free access to the communication space and to the task space, and no specific process was enforced. The action/dialog trace example comes from the beginning of a collaborative use case diagram design session. Three learners, referred to as "toto," "titi," and "tata," are participating. This example is representative of average collaborative sessions that have been logged and was not selected because it contains contributions of a particular interest.

Use case diagrams overview the usage requirements for a system. Each use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse. An actor is a person, organization, or external system that plays a role in one or more interactions with the system. Actors are drawn as stick figures. Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicate the direction of the initial invocation of the relationship. Examples of use case diagrams are shown in Fig. 2. The following excerpt describes the part of the situation that the three students were trying to specify with a use case diagram: "The process starts when an employee asks for a training course. The employee can possibly consult the course catalog that contains the courses proposed by the providers selected by the training manager. The manager examines what the employee has asked. To take his/her decision, *i.e.* accept or reject the application, the manager consults the catalog. He/she describes the selected training course and the list of sessions that are planned. When the employee has selected a session, the manager registers the employee for that session at the course provider. If the employee cannot attend the session he/she must inform immediately the manager who will ask the provider to cancel the registration."

Omega+ produces XML-based log files that can be filtered and transformed into different presentation formats such as text, CSV, or SQL, through XSL technologies by the Omega+ LogFormatter tool. The log file excerpt below is shown as a manually annotated text. Actions of moving nodes and links have been automatically filtered from the original trace. A few off-task messages have been also manually removed. Utterances and component names have been translated from French and the original version is given in parenthesis. Each line starts with the date and time, the name of the session ("ex1" in the excerpt), the name of the learner, the action type ("says" for a chat contribution and "performs a diagram action" for a diagram editor contribution). For a chat contribution, the message follows (in bold). For a tool contribution, the line includes the tool type and number (e.g., "Diagrammer0") and a tool-dependent action description. In diagrams, a vertex has an internal identifier composed by the learner's name followed by a sequence number: "addVertex:actor:tata0:" means that tata has created an actor vertex identified by "tata0". Links have an identifier composed of the two vertex identifiers followed by the learner's name and a sequence number: "addEdge: interactionWithArrow:tata0:toto0:toto1:" means that toto has created an edge of type "interaction with an arrow" between tata0 vertex and toto0 vertex identified by "tata0:toto0:toto1". In the following excerpt, GAPs are numbered, annotated by the analyst (in italics), and separated by dotted lines.

Jan 30 15:25:09 in ex1 tata performs a diagram action: Diagrammer0 addVertex: actor:tata0:
Jan 30 15:25:20 in ex1 tata performs a diagram action: Diagrammer0 newName: employee (employé):tata0:|:
Jan 30 15:25:27 in ex1 tata says: **I have put the employee** (j'ai mis un employé)
[1] GAP of type 1: spontaneous actor addition with an explanation from tata

Jan 30 15:25:38 in ex1 toto says: **ok**
[2] GAP of type 11: evaluation of [1] by toto

Jan 30 15:26:31 in ex1 toto says: **he can ask for a formation or consult the catalog** (il pourra faire une demande de formation ou une demande de consulter le catalogue)
[3] GAP of type 10: suggestion from toto, intra-subjective reaction to [2]

Jan 30 15:26:43 in ex1 titi says: **add also a manager** (il faut mettre un responsable aussi)
[4] GAP of type 2: spontaneous suggestion from titi which opens a new topic

Jan 30 15:26:54 in ex1 toto says: **yes** (oui)
[5] GAP of type 13: evaluation of [4] by toto (ambiguous because it could also concern [3])

Jan 30 15:26:54 in ex1 tata says: **yes titi** (oui titi)
[6] GAP of type 13: evaluation of [4] by tata (disambiguated because it could also concern [3])

Jan 30 15:27:00 in ex1 tata performs a diagram action: Diagrammer0 addVertex: actor:tata1:
Jan 30 15:27:06 in ex1 tata performs a diagram action: Diagrammer0 newName: manager (responsable):tata1:|:
[7] GAP of type 9: actor addition by tata, reaction to [6] from titi

Jan 30 15:27:07 in ex1 titi performs a diagram action: Diagrammer0 addVertex: case:titi1:
[8] GAP of type 9: case addition by titi, reaction to [3] from toto

Jan 30 15:27:12 in ex1 toto says: **I put the two cases I mentioned** (je place les deux fonctions que j'ai cité)
Jan 30 15:27:14 in ex1 toto performs a diagram action: Diagrammer0 addVertex: case:toto0:
[9] GAP of type 9: case addition, intra-subjective reaction to [3] with an explanation before the action

Jan 30 15:27:19 in ex1 titi performs a diagram action: Diagrammer0 newName: Ask formation (Demande de formation):titi1:|:
[8í] Termination of [8]

Jan 30 15:27:27 in ex1 toto performs a diagram action: Diagrammer0 newName: Consult catalog (Demande de consultation):toto0:|:
[9í] Termination of [9]

Jan 30 15:27:52 in ex1 toto performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: tata0:toto0:toto0:
[10] GAP of type 4: edge addition, intra-subjective reaction to [9] (relates the actor to the case)

Jan 30 15:28:01 in ex1 titi says: **the manager must inform the employee and give him/her a list of available sessions** (le responsable doit informer l'employé et donner la liste des sessions)
[11] GAP of type 10: suggestion from titi, reaction to [7] from tata

Jan 30 15:28:13 in ex1 toto says: **yes** (oui)
[12] GAP of type 13: evaluation of [11] by toto

Jan 30 15:28:19 in ex1 tata says: **yes** (oui)
[13] GAP of type 13: evaluation of [11] by tata

Jan 30 15:28:25 in ex1 titi says: **I put them** (j'les met)
Jan 30 15:28:28 in ex1 titi performs a diagram action: Diagrammer0 addVertex: case:titi2:
[14] GAP of type 9: case addition by titi, reaction to [13] from tata with an explanation before the action

Jan 30 15:28:40 in ex1 toto says: **ok**
[15] GAP of type 12: evaluation of [14] by toto

Jan 30 15:29:05 in ex1 titi performs a diagram action: Diagrammer0 newName: Inform (Information sur la formation):titi2:|:
[14'] Termination of [14]
See the first intermediate state of the use case diagram in Fig2 a.

- -

Jan 30 15:29:11 in ex1 tata says: **the course provider is also relevant, don't you think so?** (il faut aussi l'organisme de formation non?)
[16] GAP of type 2: spontaneous suggestion from tata which opens a new topic

- -

Jan 30 15:29:34 in ex1 toto says: **yes** (oui)
[17] GAP of type 13: evaluation of [16] by toto

- -

Jan 30 15:29:41 in ex1 tata says: **I put it then** (je le met alors)
Jan 30 15:29:49 in ex1 tata performs a diagram action: Diagrammer0 addVertex: actor:tata2:
[18] GAP of type 9: actor addition by tata, reaction to [17] from toto with an explanation before the action

- -

Jan 30 15:29:54 in ex1 titi says: **probably yes and the inscription by the manager** (oui sans doute et l'inscription par le responsable)
[19] GAP of type 13: evaluation of [16] by titi
[20] GAP of type 12: an additional suggestion is included in the same message, uptake of [18] from tata

- -

Jan 30 15:30:02 in ex1 titi performs a diagram action: Diagrammer0 addVertex: case:titi3:
[21] GAP of type 4: case addition by titi, intra-subjective reaction to [14]

- -

Jan 30 15:30:06 in ex1 tata performs a diagram action: Diagrammer0 newName: Formation provider (organisme de formation):tata2:|:
[18í] Termination of [18]

- -

Jan 30 15:30:10 in ex1 titi performs a diagram action: Diagrammer0 newName: Session list (Liste des sessions):titi3:|:
[21í] Termination of [21]

- -

Jan 30 15:30:50 in ex1 titi performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: tata1:titi2:titi4:
[22] GAP of type 4: edge addition by titi, intra-subjective reaction to [14] (relates the actor to the case)

- -

Jan 30 15:31:00 in ex1 titi performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: titi2:tata0:titi6:
[23] GAP of type 4: edge addition by titi, intra-subjective reaction to [22] (relates the case to the actor)

- -

Jan 30 15:31:11 in ex1 titi performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: tata0:titi1:titi7:
[24] GAP of type 4: edge addition by titi, intra-subjective reaction to [8] (relates the actor and the case)

- -

Jan 30 15:31:15 in ex1 titi performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: titi1:tata1:titi8:
[25] GAP of type 4: edge addition by titi, intra-subjective reaction to [24] (relate the case and the actor)
See the second intermediate state of the use case diagram in Fig. 2b.

- -

Jan 30 15:32:18 in ex1 toto says: **there is also the decision of the manager** (y'a aussi la décision du responsable)
Jan 30 15:32:25 in ex1 toto says: **refusal or acceptance** (le refus ou pas)
[26] GAP of type 12: suggestion from toto, uptake of [21] from titi with a precision

- -

Jan 30 15:32:34 in ex1 toto performs a diagram action: Diagrammer0 addVertex: case:toto1:
[27] GAP of type 9: case addition from toto, intra-subjective uptake of [26] (in the absence of reaction, interpreted as an approval, the suggestion is concretized into an action)

Jan 30 15:32:37 in ex1 titi says: **there is also inscription cancelling from the employee to the manager and from the manager to the formation provider** (il y a l'annulation de l'inscription, de l'employé vers le responsable et du responsable vers l'organisme)
[28] GAP of type 12: suggestion from titi, reaction to [27] from toto

---

Jan 30 15:32:47 in ex1 toto performs a diagram action: Diagrammer0 newName: Take decision (Decision responsable):toto1:|:
[27'] Termination of [27]

Jan 30 15:33:03 in ex1 toto performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: tata1:toto1:toto7:
[29] GAP of type 4: edge addition by toto, intra-subjective reaction to [27] (relates the actor and the case)

Jan 30 15:33:05 in ex1 tata says: **is the catalog relevant?** (le catalogue des formations il faut le mettre?)
[30] GAP of type 2: spontaneous question from tata which opens a new topic

---

Jan 30 15:33:11 in ex1 toto performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: toto1:tata0:toto8:
[31] GAP of type 4: edge addition by toto, intra-subjective reaction to [29] (relate the case and the actor)

Jan 30 15:33:21 in ex1 toto says: **no, because it is an object** (non parce que c'est un objet)
[32] GAP of type 14: answer from toto to the question [30] from tata

---

Jan 30 15:33:22 in ex1 titi performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: tata1:titi3:titi9:
[33] GAP of type 4: edge addition by titi, inter-subjective reaction to [21] (relates the actor and the case)

Jan 30 15:33:28 in ex1 titi performs a diagram action: Diagrammer0 addEdge: interactionWithArrow: titi3:tata0:titi10:
[34] GAP of type 4: edge addition by titi, inter-subjective reaction to [33] (relates the case and the actor)

Jan 30 15:33:35 in ex1 titi says: **I agree** (je suis d'accord)
[35] GAP of type 14: answer from titi to the question [30] from tata
See the final state of the use case diagram in Fig. 2c.

This excerpt exemplifies two particular cases. First, a contribution including several elementary actions like "addVertex" followed by "newName" can be interrupted by another contribution that occurs in parallel. The model uses the concept of "contribution termination" in this specific case ([8′], [9′], [14′]…). Second, a chat contribution can serve two different functions, such as evaluating and suggesting. In this case, the chat contribution is divided into two distinct GAPs (e.g., [19], [20]). This excerpt also shows that in a few cases the pairing process is not fully independent of application-related knowledge. In GAP [11] the suggestion "The manager must inform the employee and give him/her a list of available sessions" is paired with the creation of the manager actor while the suggestion also mentions the employee. The reason is that use cases are, by essence, associated to their initiators (a use case is a sequence of actions performed by the system for producing some useful result for its initiating actor). It is the reason why the GAP definition refers to the broad idea of "conceptual adjacency."

The graphical notation

A graphical representation is well adapted to reveal the syntactical structure of a trace (Wee and Looi 2007). In the proposed notation, time flows from the top to the bottom. Nodes are associated to contributions. Arrows are associated to intra-subjective and inter-subjective reactions (with dotted lines for relating a contribution and its termination). Therefore, a GAP is a pair of nodes linked by an arrow, except for the first two GAP types (spontaneous initiative) where there is no origin node. Each node has a distinctive shape which specifies who has performed the contribution: in the following example, a triangle for tata, an oval for toto, and a rectangle for titi. Each node also has a distinctive colour for showing the tool that has been used: light grey for the chat tool and white for the diagram editor. A small circle before or after a node indicates the existence of an accompanying message before or after the contribution. The GAP number is written near the reaction (second part) node. Finally, the label of each node gives the precise nature of the contribution: "s" for a suggestion, "e" for an evaluation, "q" for a question, "a" for an answer, "A" for an additive tool contribution, "C" for a change contribution, and "D" for a destructive contribution. A "=" or "≠" character can follow "C" for specifying if the target component is the same or different in both parts of a GAP and disambiguating between GAP types 3 and 4. Similarly, a "!" character can follow "A" or "D" for specifying a "war action" and disambiguating between 5 and 7 or 6 and 8 GAP types. Figure 3 gives the graphical representation of the previous example.

At a glance, it is possible to analyze some characteristics of the collaborative learning process, such as its topical structure and the use of the different media. A distinct tree in the graph is called a "topic." Most topics start with an opening phase where a learner takes a spontaneous initiative through an action, suggestion, or question. During the following construction phase, the group collaboratively elaborates on what was proposed. In the closing phase, one can often observe a predominance of tool contributions without evaluations. The graphical representation shows a predominance of chat-based initiatives for starting new topics, in three of the four cases. The participation of the different learners and their individual trajectories (Suthers et al. 2007) can also be analyzed. For instance, the graphical representation reveals that titi often terminates the work in the closing phases of the topics and rarely evaluates other participants' contributions.

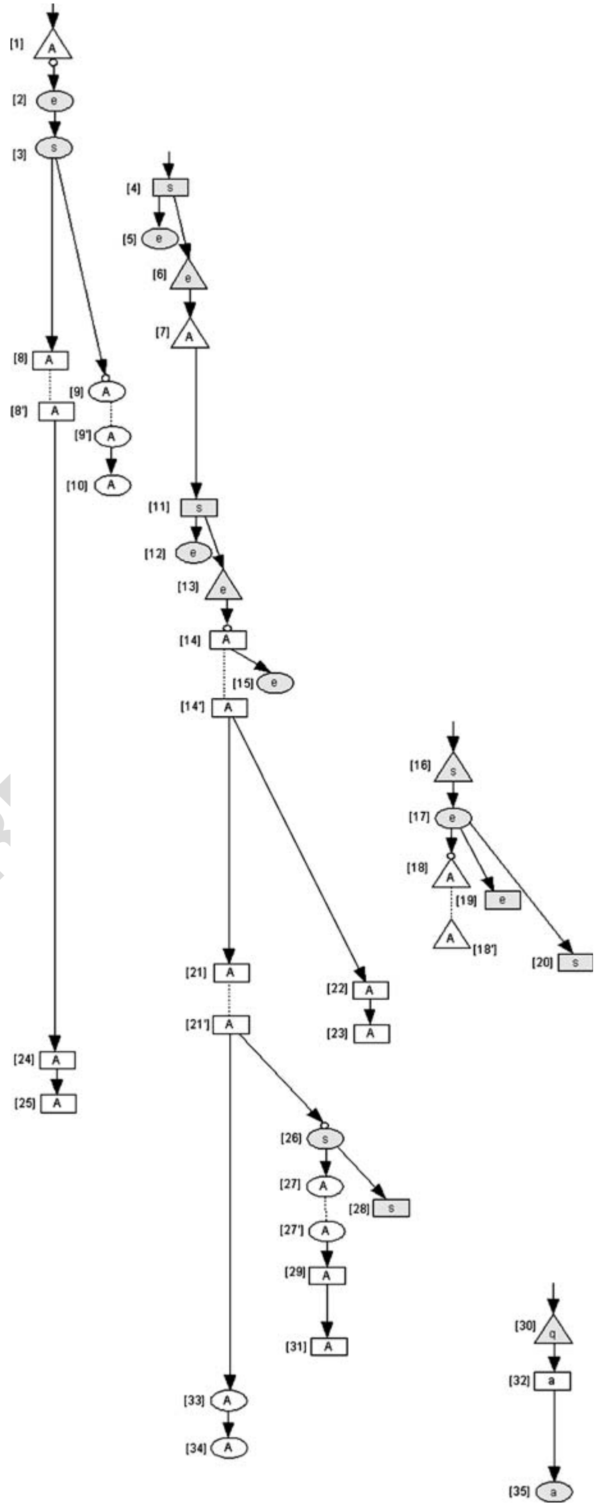
However, it would be hazardous to make decisions on the way the learning situation or the supporting environment has to be changed on the sole basis of such a *surface analysis*. If a graph shows that learners interact less than expected for a given activity, before constraining the interaction flow by creating specialized roles or by enforcing specific protocols, it would probably be better first to seek to understand how learners have coped with the proposed learning situation and how they actually behaved at the socio-cognitive level. The next two sections consider how to conduct such deeper analysis.

The knowledge level

The approach

In many recent theories, learning is understood through a *knowledge-building metaphor* and address the same kinds of questions concerning how new knowledge is created by innovative communities (Paavola et al. 2002). The goal of this intermediate level is to explain *how* the events take place from this collaborative knowledge-building perspective. When compared with the dialogue level, a larger grained unit of analysis is required. Similarly, Beers et al. (2007) emphasizes that "the sentence and the turn level are too fine-grained to identify negotiation episodes" (Beers et al. 2007). Schrire (2006) also distinguishes three levels of analysis with their own analysis units: discourse (move types) at the finest grain size, cognitive content at the medium level and interaction (threads) at the coarsest grain size. The unit of analysis at the knowledge level, called "*episode,*" is a sequence of GAPs that addresses the same primitive goal in collaborative knowledge building.

Fig. 3 The graphical representation of the example



In the global bottom-up approach that is proposed in this article, the knowledge level should remain independent of the learning task. Task-related explanations of *why* learners behave as observed will take place at the next level. The classification scheme of collaborative knowledge-building episodes is targeted to qualitative analysis. The objective is not to classify precisely all the episodes that take place for counting them and testing some predefined hypothesis but to *characterize the most important recurrent episodes* that will help in passing from local interpretations to more global ones. The search of recurrent elements is central to other proposals in the field. Wee and Looi (2007) define the notion of “pivotal contribution” as a “contribution that plays a significant role in changing the direction of the discourse.” These pivotal contributions are the basis for analyzing math reasoning in terms of meaning-making paths. Similarly, Trausan-Matu et al. (2006) emphasize “strong chat utterances” that influence the future of a conversation. The strength is evaluated by the number and importance of utterances that explicitly or implicitly refer to them. Stahl (2007) proposes the concept of “math proposal adjacency pair” which is a recurrent pattern starting with a proposal bid made by someone. This bid is taken up by someone else and the proposal is elaborated by the learners, possibly by means of similar secondary patterns. This pattern is the basis for analyzing at the task-dependent level math problem solving in terms of making math proposals.

The classification scheme of collaborative knowledge-building episodes described in the next section is based on previous proposals in the CSCL literature, but also takes into account the specificities of DIS systems and has been adjusted during case studies.

The classification scheme

For many researchers (Garrison et al. 2001; Veerman and Veldhuis-Diermanse 2001; Beers et al. 2007), most collaborative knowledge-building activities share, at a very abstract level, a common structure: They start with an externalization event which triggers more or less complex “exploration phase” and “resolution phase.”

The triggering event corresponds to the externalization by one learner of some private tacit knowledge which is made explicit (by words) or tangible (by tool actions) to the others (Polanyi 1962; Nonaka and Takeuchi 1995) in order to be analyzed and evaluated. In general, this triggering event corresponds to a single contribution either spontaneous or complementing a previous contribution. For dealing with more complex externalization processes, it is possible to define an “*initiation episode*” similar to the “initiation phase” of Garrison et al. (2001).

The exploration phase can include different subprocesses. When learners internalize the initial contribution, they can detect differences of understanding which can be resolved through verification-clarification exchanges and differences of opinions which can be debated by exchanging arguments (Beers et al. 2007). The first case corresponds to what is called a “*clarification episode*” in the proposed classification and the second case is called an “*argumentation episode.*” Some classifications do not distinguish between the two cases, under the umbrella terms of “exploration” (Garrison et al. 2001) or “explicitation” (Veerman and Veldhuis-Diermanse 2001), while others include more detailed categories such as “elementary clarification” and “in-depth clarification” for Henri (1992) or “elicitation” as a specific case of clarification for Fischer et al. (2003).

During the resolution phase, a consensus regarding the acceptance/rejection of what was suggested or produced is negotiated and personal agreement/ disagreement about the result can be expressed (Garrison et al. 2001; Fischer et al. 2003; Beers et al. 2007). Accepted suggestions can be implemented by tool actions. The term "*resolution episode*" is used in the proposed classification.

During "*planning episodes,*" learners explicitly address the issues of how to organize their work such as task division, time planning, and so forth. This first class is called "task coordination" in Strijbos et al. (2006).

Finally, the concept of "*war episod*e" is specific to DIS systems. During such an episode, two or more learners solve a cognitive conflict by acting directly on a shared artifact without any accompanying explanation. It is the case, for instance, when a learner "silently" suppresses a component or property just created by another learner or recreates something just suppressed by someone else. It is symptomatic of underlying problems that need to be considered.

The next two sections deepen the concept of collaborative knowledge-building episodes by analyzing the relationships that exist between episode types and GAP types and by discussing the generic (task-independent) learning mechanisms that can take place in each episode type.

From GAP types to episode types

A spontaneous initiation episode starts with a GAP of type 1 (spontaneous tool action) or a GAP of type 2 (spontaneous suggestion or question). When the initiation episode complements a previous contribution, it starts with a suggestion or a question (GAP of type 10 or 12). The episode can also include GAPs of type 12 for rectifying the initial suggestion or GAPs of type 3 for modifying the initial tool action. In the excerpt, all initiation episodes contain a single contribution ([1], [3], [4], [11], [16], [20], [26], [28], and [30]).

Clarification episodes mainly include GAPs of type 14 (question-answer) and GAPs of type 10 or 12 which produce suggestions and questions with a clarification objective. Argumentation episodes mainly include GAPs of type 11 and 13 when the evaluative message in the second part corresponds to positive or negative arguments. It is also possible to find GAPs of type 14 (question-answer) for deepening the argumentation process. The excerpt does not contain examples of clarification and argumentation episodes.

Resolution episodes mainly include GAPs of type 11 and 13 when the evaluative message in the second part corresponds to acceptance/rejection or agreement/ disagreement types. It can also contain a sequence of related tool actions (GAP of type 9 and types 3 to 6) which directly and immediately implement what was suggested and agreed upon by the learners. In the excerpt, resolution episodes mainly contain acceptance messages and tool actions implementing what was suggested before: [2], [8, 9], [5, 6, 7], [12, 13, 14, 15, 21], [17, 18. 19], [27, 29, 31], [32, 35].

Finally, by definition, a war episode only contains GAPs of type 7 (destructive action without explanation) and 8 (additive action without explanation).

The associated learning mechanisms

For a long time, many researchers have tried to characterize *generic learning mechanisms*. This section tries to establish a link between the proposed episode types and these theoretical mechanisms. (1) The action of externalizing personal knowledge can lead to individual learning through deepening and clarification of that personal knowledge (*learning by articulating tacit knowledge*; Webb 1982). This process is sometimes made visible through accompanying explanations or questions to other learners. (2) The action of putting personal knowledge in the social arena always launches an implicit interrogation about its correctness. The answer is either an explicit evaluation by others or a lack of response which is generally interpreted as an implicit agreement. This can lead to *learning by reinforcing or weakening personal knowledge schemas through their evaluation by peers* (Topping 1998). (3) A participant who observes the externalization of some piece of knowledge can *learn by imitation* at the tacit level (Bandura 1977). It is very effective, in particular for procedural aspects. During a process for learning UML class schema design, it is possible to define a specific phase where all inheritance relationships are created among a set of candidate classes. Learners can repetitively observe and assimilate the concept through the procedure which includes the creation of the superclass, of the inheritance relationships between the superclass and the subclasses and the transfer of common members from the subclasses to the superclass. (4) When a learner reacts to another learner's contribution, he/she must first decode and interpret the contribution. If the learner has already sufficient cognitive structures in place, an interindividual knowledge transfer process, called *learning by assimilation*, can take place (Piaget 1977; Cress and Kiemmerle 2008). (5) This assimilated knowledge can then interact with other existing knowledge elements in the second learner's knowledge space and modify these elements or trigger the inference of additional knowledge. The subsequent contributions can reflect these evolutions, called *learning by accommodation* (Piaget 1977; Cress and Kiemmerle 2008). (6) Synergistic knowledge building can also take place during argumentative episodes where personal perspectives interact and can be merged into a group perspective (*learning through argumentation*; Lipman 1991).

This enumeration shows the variety and complexity of the learning processes that take place during the collaborative learning sessions under consideration. Furthermore, it seems difficult to establish a link between these theoretical mechanisms and the concrete issue of better customizing the supporting environment that we want to address. Only a global task-dependent interpretation of why learners behave as observed in a specific context may help to reach this goal.

The example

The analysis at the knowledge level aims at detecting and characterizing recurrent elements which structure the generalized conversation description. In the previous example, the most obvious recurrent elements are related initiation and resolution episodes. Table 3 shows the different topics defined at the dialog level with the associated sequence of GAPs (characterized by their numbers and types) and episodes. Clarification and argumentation episodes are missing in the excerpt. Most importantly, many tool actions which are not direct consequences of the resolution episode appear at the end of each topic (cells shaded in grey in Table 3). They correspond to indirect consequences of the resolution which are not evaluated by the learners. Therefore, the recurrent three-step pattern observed in the example is slightly different from the theoretical "externalization-exploration-resolution" structure (Garrison et al. 2001; Veerman and Veldhuis-Diermanse 2001; Beers et al. 2007). Q3 Its initial *externalization step* corresponds to the creation of an element of the representation, or a suggestion for creating such an element, or a question about the possibility of creating such an element, taking the form of a tool action with an accompanying message or a chat contribution. The central *resolution step* includes positive or negative evaluations about the proposed or suggested element. In the case of a suggestion, the proposed action is implemented by tool actions if it is agreed to. Often, this agreement triggers a final *follow-up step,* including a sequence of tool actions that t3.1 **Table 3** The intermediate level analysis

Topic 1								
GAP#	1	2	3	8	9	10	24	25
GAP types	1	11	10	9	9	4	4	4
Episodes	I_1	R_1	I_2	R_2				
3-step pattern & sub-patterns	↔	↔	←					→
			↔	←	→			

I_i: initiation episode
R_i: resolution episode

Topic 2																			
GAP#	4	5	6	7	11	12	13	14	15	21	22	23	26	27	28	29	31	33	34
GAP types	2	13	13	9	10	13	13	9	12	4	4	4	12	9	12	4	4	4	4
Episodes	I_3	R_3			I_4	R_4							I_5	R_5	I_6	R_5			
3-step pattern & sub-patterns	↔	←		→	←														→
					↔	←				→			↔	←			→		
															↔				

Topic 3						Topic 4		
GAP#	16	17	18	19	20	30	32	35
GAP types	1	11	10	9	9	2	14	14
Episodes	I_7	R_7			I_8	I_9	R_9	
3-step pattern & sub-patterns	↔	←		→	↔	↔	←	→
					↔			

implement all the consequences of the creation of the proposed element which do not require additional evaluation. The follow-up step can also include fully-fledged sub-patterns for the most complex cases that require to be discussed. Table 3 shows all the three-step patterns and sub-patterns of the excerpt. One can notice two incomplete patterns corresponding to initiation episodes which are not followed by resolutions at this point of time ([20], [28]).

This description is not sufficient for explaining *why* the collaborative learning session unfolds in this way. The use case diagram at the end of the excerpt (see Fig. 2) contains a large number of components (about 20 actors, cases, and relationships between actors and cases) compared to the small number of patterns that were found. So, many questions remain unanswered like: Which types of component are fully discussed? In which order? Why? What is the influence of the software engineering formalism? The next section addresses this kind of question on the basis of a task-dependent interpretation.

The action level

The approach

The recurrent meaningful elements characterized at the knowledge level constitute the starting point for building a global interpretation of why the collaborative learning process unfolds as observed. At the action level, analysts must focus on all task-dependent aspects and characteristics that can explain the collaborative process and give sense to what has been observed at the lower levels. The main guidelines that can be given are that the proposed interpretation should rely on precise *domain knowledge* and, when possible, on *domain-related theories*.

Then, analysts must determine the concrete changes to the learning situation and environment configuration to be carried out for better supporting learners in the way they work. Analysts can first select some *general improvement directions*. “Facilitating the categorization process of actors” and “enforcing an earlier structuring process of use cases” constitute two possible improvement directions for the example. Finally, analysts may follow different strategies for finding the *concrete changes* in each improvement direction. They can consider successively the four modeling dimensions in Omega+, that is, the process, artifact, interaction, and monitoring dimensions. They can also rely on several scaffolding frameworks proposed in the CSCL literature (e.g., Reiser 2004; Quintana et al. 2004) which define and classify many generic solutions like “decomposing the learning task,” “forcing learners to address important aspects or processes that they might otherwise overlook,” “providing access to expert knowledge,” “using representations and language that bridge learners’ understanding,” and so forth.

The example—a candidate interpretation

In the first part of the example, it is argued that learners do not negotiate modelling elements (which are always actors at the beginning of use case diagram construction) but negotiate in highly implicit terms *modeling rules and their immediate and possibly repetitive application by the group*. The modelling rule is defined by a *case* (example) rather than by an explicit definition. When an element is proposed, it is interpreted as a prototypical example of a category of modelling elements and learners have to infer the corresponding modelling rule. It is in line with the *modern theories of categorization in cognitive science*. The classical Aristotelian view is that categories are mentally represented as sets of necessary and sufficient conditions. In contrast, according to the prototype theory (Rosch 1975), a category’s mental representation is based on prototypical exemplars or prototypes. According to the exemplar theory (Nosofsky 1988), a category’s mental representation encodes the exemplars that compose the category. To decide whether an entity is a member of a category, this entity is compared either to the category’s prototype or to the category’s exemplars. In most models, exemplars are mentally represented in a psychological space whose dimensions correspond to perceptual dimensions along which the category’s exemplars vary. In the collaborative session, all learners share a common knowledge ground including both definitional elements and a set of exemplars resulting from the introduction course and from the exercises they have participated in previously. In the excerpt of section 2, tata starts the session by creating a first actor [1]. The accompanying comment does not include any questioning (“I have put the employee”). For tata, this case implicitly satisfies the modelling rule stating that “someone external to the system who acts on the system is modelled as an actor.” So, there is no need for any additional comment. The same reasoning holds for the second proposal from titi [4] (“add also a manager”). The “also” adverb can be interpreted as “in the same category (of active actors),” creating a semantic relationship between the two cases that has not been perceived at the dialogical level. By contrast, in [16] (“the course provider is also relevant, don’t you think so?”), tata explicitly asks for a confirmation probably because the course provider case does not share all the properties of the first two cases. The differentiating property is passiveness: The course provider does not take any initiative in the process but just passively receives information, that is, is acted on by the system. The question implicitly raised by tata is about the relevance of such passive participants as actors in a use case diagram. When the uncertainty exceeds a certain level, the initial externalization part of the pattern can take the form of an explicit question, like in [30] ("is the catalog relevant?" that can be interpreted as "should a concept such as the training course catalog be modelled as an actor?"). The short answer from toto [32] is meaningful in the context of the common knowledge ground. "No, because it is an object" does not mean that an actor cannot be an object but means that the catalog is an informational resource—that is, in UML object-oriented modelling terms, an instance of a class which should be modelled into object and class diagrams and not into use case diagrams.

The application of the rule by the group, corresponding to the follow-up part of the pattern, can implement two strategies: "horizontal search," where the rule is applied repetitively for searching as much as possible exemplars of the category (possibly defining subcategories), and "vertical search," where all the properties of each case are considered through multiple sub-patterns. Both strategies are used and interweaved. With that interpretation, the excerpt of section 2 appears now as a single unified process whose structure is depicted in Fig. 4. What was called "topic" at the dialog level is now interpreted as a step during the course of the horizontal search strategy during which three subcategories are recognized: active actor, passive actor, and "not an actor."

The proposed interpretation in terms of categorization rule negotiation is consistent with the way the concept of actor is defined in the use case literature, that is, in terms of categories and prototypical examples. One can find generally multidimensional classifications in which the main axes are the nature of the actor (person, group, device, system, subsystem, organization, etc.) and its behavioral profile (active/passive, primary actor/supporting actors, etc.). During the categorization process, all the participants have to explore their personal memories for comparing the proposed actor with the more or less prototypical exemplars they know. Participants can benefit from that collaborative evaluation for evolving their personal memories. In the excerpt, the rule about the special case of passive actors was not clearly owned by tata (who has asked the question about formation providers [16]) nor by titi (who has answered "probably yes" [19]) but only by toto whose answer was affirmative. Tata and titi can add this exemplar to the actor category (and the passive actor subcategory) that will help them form a correct categorization of passive actors in the future. It is important to emphasize that these rules are not stated explicitly and passively received but actively constructed by each learner. In the less

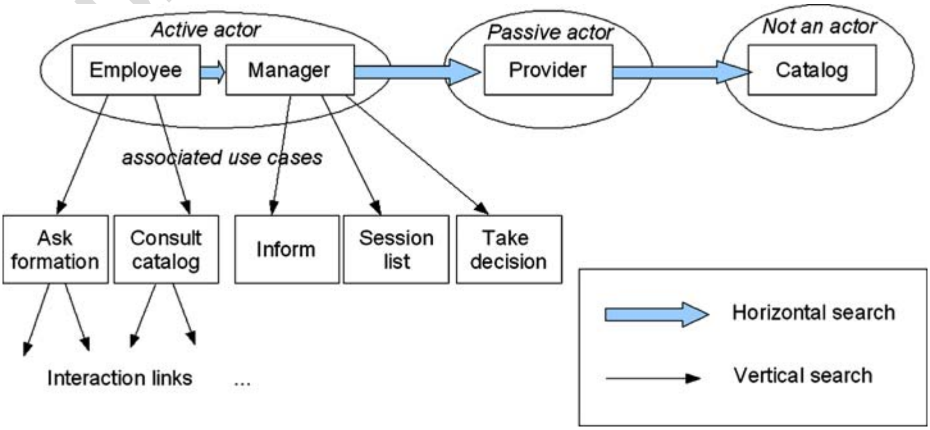


Fig. 4 The high-level task-dependent process interpretation

frequent case of explicit rule externalization, participants can benefit from original points of view and wordings from peers. "Negative rules," such as the rule specifying that an object is not an actor (proposed by toto [32]), can greatly help in the categorization process. They are easier to internalize than artificial counter-examples because they are embedded into a meaningful context.

Use cases are externalized during similar three-step patterns: instances then actors. This happens during what was called the "vertical search" strategy, that is, the search of all the elements associated to a given actor. The underlying process is no more a categorization process but should be a *structuring process*. In the literature, the concept of use case is never described in terms of categories and prototypical examples but with an analytic definition and a set of properties such as: "A use case is a sequence of actions performed by the system for producing some useful result for its initiating actor; it is characterized by its goal, preconditions, and postconditions, the basic course of its actions, alternative paths, its initiator, and its supporting actors." All actions in a use case share the same initiator, participate to the same overall goal, and have consistent preconditions. In the excerpt, the use cases that are proposed match exactly the elementary actions mentioned in the problem definition and are accepted without analysis and discussion. During this early part of the session, there is no example of a structuring process. It can be hypothesized that these structuring processes are delayed later in the collaborative session when the learners can grasp the whole system. In consequence, the diagram in Fig. 2 is not fully satisfying at this stage. For instance, "Inform" and "Session list" use cases are probably two components of a larger "Process the application" use case at the initiative of the manager: Either the application is refused and the employee is informed or it is accepted and the selected training course with the corresponding session list is transmitted to the employee.

Finally, interaction links between actors and use cases are created at the end of the vertical search process without additional evaluation as the "vertical relationship" between the actor and the use case has already been accepted.

The example—improving the supporting system

In this section, some possible improvements are suggested which impact all Omega+ modeling dimensions and follow the two previously mentioned improvement directions ("facilitating the categorization process of actors" and "enforcing an earlier structuring process of use cases"). These proposals are not necessarily original from the research point of view but are deeply rooted in the proposed task-dependent interpretation.

The first proposal takes into account the fact that the underlying cognitive processes are different for identifying actors and use cases. The idea is to focus exclusively on the "horizontal search" of actors during an initial dedicated phase. The second part of the process will include either a single "vertical" use case search phase or one specific phase for each actor.

During the initial phase, a second idea is to make categorization rules more explicit. It is in line with Klausmeier's theory of Concept Learning and Development which describes how a given concept is attained at four successively higher levels of understanding: (1) Concrete, that is, recognizing an object which has been encountered previously. (2) Identity, that is, recognizing a known object when it appears in a different spatial, time, or sensory perspective. (3) Classificatory, that is, generalizing that two items are alike in some way. (4) Formal, that is, naming and defining the concept, listing its attributes, and judging the presence of such attributes in an object (Klausmeier 1992). For reaching the higher formal level of understanding of the categorization process, a learner could play the role of "scribe" with the unique goal of writing down the categorization rules applied during actor identification. The scribe would question other learners when necessary for making the rules explicit and reaching a consensus on their definitions. The corresponding artifact, which could take a tabular form with a first column for rule definitions and a second column for categories' exemplars, would be reused during all the case studies carried out by the same students.

A third idea for the initial phase is to facilitate direct access in the session history to a specific categorization episode by the name of its prototypical exemplar. It requires a session history browser providing textual search capabilities applied to *both spaces* of the DIS environment. Omega+ has been extended with such a mechanism which generalizes the single artifact history browser proposed by Mühlpfordt and Stahl (2007). When a text is searched the different occurrences can be accessed thanks to the "next," "previous," "first," and "last" buttons of the text search panel (see Fig. 5). Participants can also browse the session history step- by-step with directional buttons or with a slider located in the bottom panel of the browser. In addition, when a learner presses the "sync" button on the left of the bottom panel, a browser is automatically launched in each client environment if it is not already started, and all browsers are synchronized for enforcing a shared focus on a given point of the collaborative process history.

During the "vertical search" phase, the main idea is to strengthen the focus on the properties of the proposed use cases for better structuring them. Thanks to the artifact meta-model, it is easy to add properties to the use case concept. For instance, three attributes can give a chance to discuss how cases should be structured: "Who" (the initiator), "When" (the cause and conditions), and "How" (the course of actions). It is also possible to enforce the rule that this use case search phase cannot terminate if a case attribute has not received a value.

Moreover, high-level interpretations can also help improve some generic mechanisms of Omega+ kernel, like the monitoring device proposed in Lonchamp (2008). For quantifying

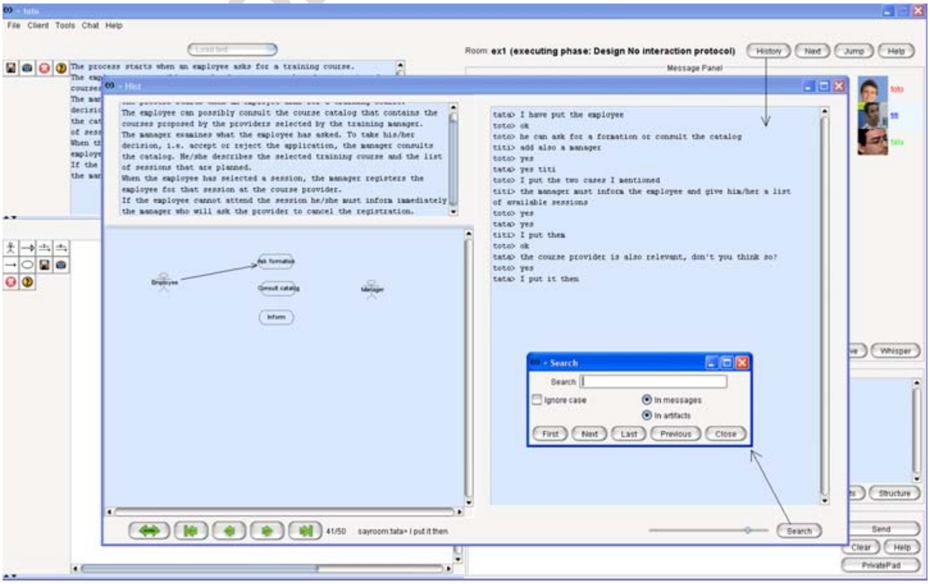


Fig. 5 Omega+ session history browser launched from Omega+ client

learners' participation, a high-level explicative model allows weighing the contributions on the basis of their relative importance. Table 4 contrasts the classical evaluation, where all contributions are equally weighted, with an evaluation based exclusively on the contributions that advance the two fundamental search processes emphasized in Fig. 4, that is, actor proposals and use case proposals. This model-informed participation evaluation applied to the excerpt demonstrates that tata, with fewer contributions than toto, has, nevertheless, a wider impact on the collaborative learning process.

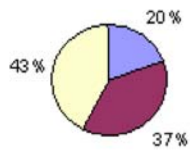
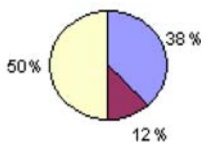
Discussion and conclusions

With a generic CSCL system, teachers, educational technology providers, or researchers can customize the learning situation and the way that technology provides scaffolding support for learners by acting on many aspects such as the process that is enforced, the roles that learners must play, the artifact types that learners can manipulate, the message types that learners can exchange, the protocols that are enforced, and the monitoring tools that are provided. It is not satisfactory to base customization only on a priori preferences, rules of thumb, or surface analysis of how learners interact. The goal of this article was to provide a methodological approach for deeply analyzing interaction traces in order to decide how to better customize the system in a specific context. The generic CSCL system under consideration follows the DIS paradigm which adds a level of complexity in comparison with simple text-based systems.

The proposed three-level "generic approach" can be applied to any learning task and system configuration. At the dialog level, a task-independent dialogical model is provided for analyzing action/communication traces as "generalized conversations" and a graphical notation enables the visual analysis of the "syntactical" characteristics of collaborative sessions. At the knowledge level, a typology of task-independent collaborative knowledge-building episode types that can occur during such generalized conversations is proposed. Thanks to that "semantical" interpretation grid, recurrent meaningful elements that structure low-level descriptions can be detected and characterized. These regularities help in passing from local interpretations to a global interpretation of the whole process. At the "pragmatical" action level, task-dependent socio-cognitive interpretations of why the collaborative learning process unfolds as observed are elaborated. They constitute a firm basis for improving the customization of the system in order to support learners more efficiently.

This methodological proposal is far from being perfect and is still a work in progress. First, its *complexity* and the *extensive manual coding work* that it requires can be criticized, in particular at the dialog level. There are two ways for simplifying the task of analysts (Mühlenbrock 2001). The first solution is based on *automated coding and automated*

t4.1 **Table 4** Participation monitoring Q2

Learner	Classical evaluation		Model-based evaluation	
tata	7		3	
toto	13		1	
titi	15		4	

adjacency pairs recognition. Some preliminary experiments in the context of Omega+ have already been conducted (Lonchamp 2008). They show that it is possible, in a generic context, to distinguish between "off-task" and "on-task" messages with a naive Bayes classifier. It is also possible to automatically recognize some patterns, explicitly specified with a simple pattern definition language, such as a tool action followed immediately by an on-task message from the same learner that (probably) explains the previous action. Moreover, Erkens and Janssen (2008) demonstrate that the main communicative functions of messages in online discussions can be automatically recognized thanks to a rule system for segmentation and a rule system for dialog act coding (with respectively 300 and 1,250 rules for the Dutch language). However, recognizing generalized adjacency pairs at a generic level is much more complex because some aspects of threading and referencing can depend on domain-specific knowledge and subtle semantic interpretations. In the second solution, *learners are asked to categorize their contributions and to make explicit the dependencies they perceive* during specific (analysis-oriented) sessions. It is easier in classical text-based systems (forum or chat tools) than in DIS systems where it is necessary to reference and also relate tool actions. There is a risk that this active participation of learners could impact the process that is under study. However, both approaches should be considered for further investigation.

A second potentially controversial point concerns the interpretations at the action level. They can appear quite hypothetical like the one proposed in the collaborative use case design example. More generally, the issues of *reliability* and *validity* of the approach must be considered. In quantitative approaches, reliability indices measure precisely the level of agreement between independent coders. In qualitative approaches, the objective in terms of reliability can be to keep analysts' understanding of category definitions calibrated by making the rules as explicit as possible, reviewing, and training (Schrire 2006). What is important and what is enforced by the proposed approach is to go deep into the details at different levels of granularity and by taking different perspectives. Interpretations are based on the recognition into the action/communication traces of *"hidden indications"* such as meaningful regularities, highly critical events, or specific problems encountered by learners. The multilevel and multidimensional analysis process strengthens the reliability (credibility) of the approach. *Internal validity* relates to whether the findings reflect what is really there. Philosophically, qualitative approaches are based on the idea that there are multiple realities (Schrire 2006). So, there are also multiple possible interpretations. However, an important criterion that was already suggested is the triangulation of what is proposed with domain-related theories. Finally, the generalizability of the study, that is, its *external validity*, can be improved by multiple case studies (Schrire 2006). In the near future, we plan to conduct both cross-application comparisons in search of differences, commonalities, and generalisations at the macro level, and cross-configuration comparisons for the same application in search of the best supporting strategies and mechanisms at the micro level, leading to the further iterative improvement of both the technical infrastructure and the methodological approach.

References

Alonso, M., Py, D., & Lemeunier, T. (2008). A learning environment for object-oriented modeling, supporting metacognitive regulations. In *Proceedings 8th IEEE International Conference on Advanced Learning Technologies* (pp. 69–73). IEEE Computer Society.

Amelsvoort, M. A. A., Andriessen, J. E. B., & Kanselaar, G. (2008). How students structure and relate argumentative knowledge. *Computers in Human Behavior, 24*, 1293–1313.

Avouris N., Margaritis M., & Komis V. (2004). Modelling interaction during small-group synchronous problem-solving activities: The Synergo approach. *2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction*, 7th Conference on Intelligent Tutoring Systems (pp. 13–18). Maceio, Brazil.

Baghaei, N., & Mitrovic, A. (2006). A constraint-based collaborative environment for learning UML class diagrams. In M. Ikeda, K. Ashley & T.-W. Chan (Eds.), *Proceedings 8th International Conference on Intelligent Tutoring Systems 2006, LNCS 4053* (pp. 176–186). Berlin, Heidelberg, New York: Springer-Verlag.

Baker, M., & Lund, K. (1996). Flexibly structuring the interaction in a CSCL environment. In P. Brna, A. Paiva & J. Self (Eds.), *Proceedings of the European Conference on Artificial Intelligence in Education* (pp. 401–407). Lisbon, Portugal: Edicôes Colibri.

Bandura, A. (1977). *Social learning theory.* New York: General Learning.

Beers, P. J., Boshuizen, H. P. A., Kirschner, P. A., & Gijselaers, W. H. (2007). The analysis of negotiation of common ground in CSCL. *Learning and Instruction, 17*, 427–435.

Çakir, M. P., Zemel, A., & Stahl, G. (2007). The organization of collaborative math problem solving activities across dual interaction spaces. In C. Hmelo-Silver & A. O'Donnell (Eds.), *Proceedings of the 7th International Conference on Computer-Supported Collaborative Learning* (pp. 107–109). Mahwah, NJ: Lawrence Erlbaum Associates.

Clark, H., & Schaefer, E. (1989). Contributing to discourse. *Cognitive Science, 13*, 259–294.

Constantino-Gonzales, M. A., & Suthers, D. (2001). Coaching collaboration by comparing solutions and tracking participation. In P. Dillenbourg, A. Eurelings & K. Hakkarainen (Eds.), *European perspectives on computer-supported collaborative learning, proceedings of the first European conference on computer-supported collaborative learning* (pp. 173–180). Maastricht, NL: Universiteit Maastricht.

Cress, U., & Kimmerle, J. (2008). A systemic and cognitive view on collaborative knowledge building with wikis. *International Journal of Computer-Supported Collaborative Learning, 3*(2), 105–122.

De Chiara, R., Di Matteo, A., Manno, I., & Scarano, V. (2007). CoFFEE: Cooperative Face2Face Educational Environment. In *Proceedings of the 3 rd International Conference on Collaborative Computing: Networking, Applications and Worksharing* (CollaborateCom 2007). New York, NY.

De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education, 46*, 6–28.

Dillenbourg, P. (1999). *What do you mean by collaborative learning? Collaborative-learning: Cognitive and computational approaches* (pp. 1–19). Oxford, U.K: Elsevier.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL?* (pp. 61–91). Heerlen: Open Universiteit Nederland.

Dimitracopoulou, A., & Komis, V. (2005). Design principles for the support of modelling and collaboration in a technology-based learning environment. *International Journal of Continuing Engineering Education and Lifelong Learning, 15*(1/2), 30–55.

Erkens, G., & Janssen, J. (2008). Automatic coding of communication in collaboration protocols. *International Journal of Computer-Supported Collaborative Learning, 3*(4), 447–470.

Farnham, S., Chesley, H. R., McGhee, D. E., Kawal, R., & Landau, J. (2000). Structured online interactions: Improving the decision-making of small discussion groups. In *Proceedings of ACM Computer Supported Cooperative Work 2000* (pp. 299–308). New York, NY: ACM Press.

Fidas, C., Komis, V., Avouris, N., & Dimitracopoulou, A. (2002). Collaborative problem solving using an open modeling environment. In G. Stahl (Ed.), *Computer support for collaborative learning: Foundations for a CSCL community, Proceedings of International Conference on Computer-Supported Collaborative Learning CSCL 2002* (pp. 654–655). Hillsdale, NJ: Erlbaum.

Fischer, G. (2003). Meta-design—beyond user-centered and participatory design. In *Proceedings of the 10th International Conference on Human-Computer Interaction* (HCI 2003), Crete, Greece, 88–92.

Glassner, A., & Schwarz, B. (2005). The role of floor control and of ontology in argumentative activities with discussion-based tools. In T. Koschmann, D. Suthers & T. Chan (Eds.), *Computer supported collaborative learning 2005: The next 10 years* (pp. 170–179). Mahwah, NJ: Erlbaum.

Gogoulou, A., Gouli, E., Grigoriadou, M., & Samarakou, M. (2005). ACT: A web-based adaptive communication tool. In T. Koschmann, D. Suthers & T. W. Chan (Eds.), *Computer supported collaborative learning 2005: The next 10 years* (pp. 180–189). Mahwah, NJ: Erlbaum.

Henri, F. (1992). Computer conferencing and content analysis. In A. Kaye (Ed.), *Collaborative learning through computer conferencing: The Najaden papers* (pp. 117–136). London: Springer Verlag.

Hernández-Leo, D., Asensio-Pérez, J. I., Dimitriadis, Y., Bote-Lorenzo, M. L., Jorrín-Abellán, I. M., & Villasclaras-Fernández, E. D. (2005). Reusing IMS-LD formalized best practices in collaborative learning structuring. *Advanced Technology for Learning, 2*(3), 223–232.

Hutchby, I., & Wooffitt, R. (1998). *Conversation analysis*. Cambridge, UK: Polity.

Jermann, P. (2004). *Computer support for interaction regulation in collaborative problem-solving*. Geneva: Dissertation, University of Geneva.

Jones, C., Dirckinck-Holmfeld, L., & Lindström, B. (2006). A relational, indirect, meso-level approach to CSCL design in the next decade. *International Journal of Computer-Supported Collaborative Learning, 1*(1), 35–56.

Klausmeier, H. J. (1992). Concept learning and concept teaching. *Educational Psychologist, 27*(3), 267–286.

Landsman, S., & Alterman, R. (2003). *Building Groupware on THYME. Technical Report CS-03-234*. Waltham, MA: Brandeis University.

Levinson, S. (1983). *Pragmatics*. Cambridge, UK: Cambridge University.

Lipman, M. (1991). *Thinking in education*. New York: Cambridge Education.

Lonchamp, J. (2006). Supporting synchronous collaborative learning: a generic, multi-dimensional model. *International Journal of Computer-Supported Collaborative Learning, 1*(2), 247–276.

Lonchamp, J. (2007a). Floor control in complex CSCL synchronous environments, In *Proceedings of the Third International Conference on Web Information Systems and Technology* (pp. 397–402). Barcelona, Spain.

Lonchamp, J. (2007b). Linking conversation and task objects in complex synchronous CSCL environments. In *Proceedings of the Third International Conference on Web Information Systems and Technology* (pp. 281–288). Barcelona, Spain.

Lonchamp, J. (2008). Interaction analysis supporting participants' self-regulation in a generic CSCL system. In P. Dillenbourg & M. Specht (Eds.), *Times of convergence, Proceedings of the Third European Conference on Technology Enhanced Learning, LNCS 5192* (pp. 262–273). Berlin, Heidelberg, New York: Springer Verlag.

Maes, P. (1987). Concepts and experiments in computational reflection. *Proceedings of the 2^{nd} ACM International Conference on Object-oriented Programming Systems, Languages and Applications* (pp. 147–155). Orlando, Florida.

Morris, C. (1938). *Foundations of the theory of signs*. Chicago University Press.

Mühlenbrock, M. (2001). *Action-based collaboration analysis for group learning*. Amsterdam: Dissertations in Artificial Intelligence, IOS.

Mühlpfordt, M., & Stahl, G. (2007). The integration of synchronous communication across dual interaction spaces. In C. Hmelo-Silver & A. O'Donnell (Eds.), *Proceedings of the 7th International Conference on Computer Supported Collaborative Learning* (pp. 525–534). Mahwah, NJ: Lawrence Erlbaum Associates.

Nonaka, I., & Takeushi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation* (pp. 61–85). New York: Oxford University.

Nosofsky, R. (1988). Exemplar-based accounts of relations between classification, recognition, and typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 700–708.

Paavola, S., Lipponen, L., & Hakkarainen, K. (2002). Epistemological foundations for CSCL: A comparison of three models of innovative knowledge communities. In G. Stahl (Ed.), *Computer Support for Collaborative Learning: Foundations for a CSCL community, Proceedings of International Conference on Computer-Supported Collaborative Learning CSCL 2002* (pp. 24–32). Hillsdale, NJ: Lawrence Erlbaum Associates.

Pfister, H.-R., & Mühlpfordt, M. (2002). Supporting discourse in a synchronous learning environment: The learning protocol approach. In G. Stahl (Ed.), *Computer Support for Collaborative Learning: Foundations for a CSCL community, Proceedings of International Conference on Computer-Supported Collaborative Learning CSCL 2002* (pp. 581–589). Hillsdale, NJ: Lawrence Erlbaum Associates.

Piaget, J. (1977). *The development of thought: Equilibration of cognitive structures*. New York: The Viking.

Pinkwart, N. (2003). A plug-in architecture for graph based collaborative modeling systems. In U. Hoppe, F. Verdejo & J. Kay (Eds.), *Shaping the future of learning through intelligent technologies: Proceedings of the 11th Conference on Artificial Intelligence in Education* (pp. 535–536). Amsterdam, NL: IOS.

Polanyi, M. (1962). *Personal knowledge: towards a post critical philosophy*. London: Routledge.

Quintana, C., Reiser, B. J., Davis, E. A., Krajcik, J., Fretz, E., Duncan, R., et al. (2004). A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences, 13*(3), 337–386.

Reiser, B. J. (2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of the Learning Sciences, 13*(3), 273–304.

Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology: General, 104*(3), 192–233.

Rourke, L., Anderson, T., Garrison, D. R., & Archer, W. (1999). Assessing social presence in asynchronous text-based computer conferencing. *Journal of Distance Education, 14*, 51–70.

Schegloff, E. A. (2006). *Sequence organization in interaction: A primer in conversation analysis*. Cambridge: Cambridge University.

Schrire, S. (2006). Knowledge building in asynchronous discussion groups: Going beyond quantitative analysis. *Computers & Education, 46*, 49–70.

Soller, A., Wiebe, J., & Lesgold, A. (2002). A machine learning approach to assessing knowledge sharing during collaborative learning activities. In G. Stahl (Ed.), *Computer Support for Collaborative Learning: Foundations for a CSCL community, Proceedings of International Conference on Computer-Supported Collaborative Learning CSCL 2002* (pp. 128–137). Hillsdale, NJ: Lawrence Erlbaum Associates.

Stahl, G. (2006). *Group cognition: Computer support for building collaborative knowledge*. Cambridge, MA: MIT.

Stahl, G. (2007). Social practices of group cognition in virtual math teams. In S. Ludvigsen, A. Lund, & R. Säljö (Eds.), *Learning in social practices. ICT and new artifacts—transformation of social and cultural practices*: Pergamon.

Strijbos, J.-W., & Stahl, G. (2007). Methodological issues in developing a multi-dimensional coding procedure for small-group chat communication. *Learning and Instruction, 17*, 394–404.

Strijbos, J.-W., Martens, R. L., Prins, F. J., & Jochems, W. M. G. (2006). Content analysis: What are they talking about? *Computers & Education, 46*, 29–48.

Suthers, D. D. (2006). A qualitative analysis of collaborative knowledge construction through shared representations. *Research and Practice in Technology Enhanced Learning, 1*(2), 1–28.

Suthers, D. D., Dwyer, N., Vatrapu, R., & Medina, R. (2007). An abstract transcript notation for analyzing interactional construction of meaning in online learning. In *Proceedings of the 40th Hawaii International Conference on the Systems Science* (CD-ROM). IEEE Computer Society Press.

Topping, K. (1998). Peer-assessment between students in colleges and universities. *Review of Educational Research, 68*(3), 249–276.

Trausan-Matu, S., Stahl, G., & Sarmiento, J. (2006). Polyphonic Support for Collaborative Learning. In Y. Dimitriadis, I. Zigurs, & E. Gómez-Sánchez (Eds.) *Groupware: Design, Implementation, and Use, 12th International Workshop CRIWG 2006* (pp.132–139), LNCS 4154, Berlin, Heidelberg, New York: Springer-Verlag.

van Joolingen, W., de Jong, T., Lazonder, A., Savelsbergh, E., & Manlove, S. (2005). Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*, 671–688.

Webb, N. M. (1982). Student interaction and learning in small groups. *Review of Educational Research, 52*(3), 421–445.

Wee, J. D., & Looi, C. K (2007). Meaning-making paths as a unit of analysis in synchronous chat environments: Application of the collaboration interaction model. In *Proceedings of the Redesigning Pedagogy: Culture, Knowledge and Understanding Conference*, Singapore, May 2007.

Weinberger, A., Ertl, B., Fischer, F., & Mandl, H. (2005). Epistemic and social scripts in computer-supported collaborative learning. *Instructional Science, 33*(1), 1–30.

Zumbach, J., Muehlenbrock, M., Jansen, M., Reimann, P., & Hoppe, H.-U. (2002). Multidimensional tracking in virtual learning teams. In G. Stahl (Ed.), *Computer Support for Collaborative Learning: Foundations for a CSCL community, Proceedings of International Conference on Computer-Supported Collaborative Learning CSCL 2002* (pp. 650–651). Hillsdale, NJ: Lawrence Erlbaum Associates.