Modelling interaction during small-groups synchronous problem-solving activities: The Synergo approach

Nikolaos Avouris, Meletis Margaritis, Vassilis Komis

University of Patras, Patras, Greece
{ N.Avouris, Margaritis }@ee.upatras.gr, Komis@upatras.gr

Abstract

Monitoring and analysis of activities of small groups of students - collocated or at a distance- engaged in synchronous collaborative problem solving activity is the subject of this paper. This is discussed in the frame of Synergo, a new synchronous collaboration support environment that monitors the activity and permits visualization of various quantitative parameters, like density of interaction, symmetry of partners' activity, degree of collaboration etc, particularly useful for understanding the mechanics of collaboration. Synergo has been used for synchronous building of flow charts, concept maps, entity-relation diagrams and other semantic modeling activities by small groups of students and has been proposed as a testbed for micro-analysis of small scale interaction in order to gain an insight in collaborative learning.

Keywords: Synchronous collaborative problem solving, analysis of collaborative activity, collaboration factor

Introduction

Socially inspired theories, supported by the growing development of network and collaborative technology, have increased research on technology-based collaborative problem solving environments. These theories usually influence our considerations on effectiveness of the collaborative problem solving process, as well as the design of the collaboration-support tools involved. According to these perspectives, the methodological issues of collaboration analysis are of prime importance, given that they are directly related to the development of this research and technology area and the common understanding of the various disciplines involved.

In problem-solving collaborative learning activities in which the computer environment constitutes in itself a mediational resource, it contributes to the creation of a shared referent between the social partners (Rochelle et al, 1995). Typically the direct manipulation environments used are characterised by actions on objects representing entities or on concepts meaningful to the users. Usually operations on these objects have a reversible incremental effect on the 'environment' represented on a shared computer screen. Often more than one actor interact directly or indirectly with the objects in this world modifying their state, communicating between them and through the objects, as they advance problem solution. Various methods have been proposed for modelling and analysis of interaction during collaborative problem solving (e.g. Jermann et al. 2001, Muelenbrock and Hoppe, 1999, Martinez et al. 2003)

In this paper we outline an innovative framework for analysis of collaborative problem solving activities. This framework has been used for conceptualization of the situation of groups of individuals engaged in exploratory and design problem solving activities and for evaluation of the effectiveness of IT design supporting the process. This methodological framework is based on the "Object-oriented Collaboration Analysis Framework (OCAF)", originally proposed by Avouris et al. (2002, 2003). Recently, analysis tools have been built to support this framework, while OCAF has been used in a number of field studies investigating various aspects of collaborative problem solving (e.g. Komis et al. 2002, Margaritis et al. 2003, Avouris et al. 2004). In this paper we discuss the collaboration-support environment and the analysis method and tools that have been recently built to support the framework.

OCAF studies the activity through the objects of the solution, that is the objects that exist in the problem-solving context. These objects become the centre of attention and are studied as entities that carry their own history and are "acted upon" by their owners. This perspective produces a new view of the process, according to which the solution is made up of structural components that are "owned" by actors who have contributed in various degrees to their existence. This view of the world, can be useful, as it reveals the contribution of the various actors in parts of the solution, and the relevant focus shifts (Bertelsen and Bodker, 2003), identifies areas of intense collaboration in relation to the final solution and can relate to other analysis frameworks like interaction analysis. In this paper, we describe first the Synergo collaborative problem-solving environment. Subsequently, an outline of the model of interaction is included together with presentation of the functionality of the tools that have been proposed to support analysis of interaction. Through the Synergo analysis tools, the researcher can playback the activity off-line and annotate the activity and the produced solution using an annotation scheme which can be defined and adapted according to the specific objectives of the study. A brief example of use of the framework and the tools in collaborative problem-solving situations is also presented.

The Synergo Environment

Synergo is a new collaboration support environment based on the *Abstract Collaborative Applications Building Framework (ACABF)*, also used for building ModellingSpace (Margaritis et al. 2003) and ModelsCreator v3 (Fidas et al. 2002). Synergo architecture supports synchronous collaboration, as well as integration of collaboration analysis and visualization tools.

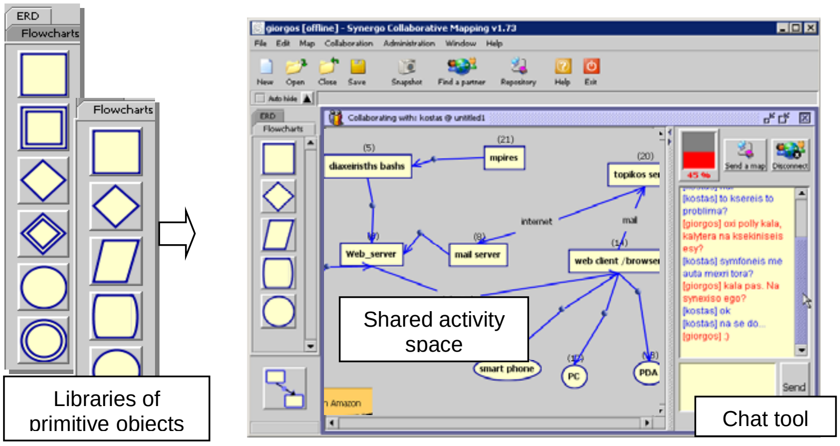

Figure 1. The Synergo environment: client user interface

The Synergo environment (http://www.ee.upatras.gr/hci/synergo) is a client-server distributed application, which comprises a suite of interconnected tools to support collaborative drawing activities. The main functionality of the Synergo environment is described through fig. 1, which shows a typical problem-solving activity. Synergo supports building of different kinds of diagrams. It contains libraries for building flowcharts, entity-relationship diagrams, concept maps, data flow diagrams etc. On the left-hand side column of figure 1, libraries of primitive objects are shown. The activity is monitored and logfiles are generated and made available for inspection by the users or researchers.

Synchronous collaboration for problem solving is a case of computer-supported collaboration based on the concept of shared artefact/work surface (Dix et al, 1998). The related notion of feed-through the artefact implies that one participant's manipulation of shared objects can be observed by the other participants. This communication through the artefact can be as important as direct communication between participants. Considering that the collaborative activity is done mainly between partners at a distance, the direct communication mechanism has to be defined. A text communication has been used in this case. One additional decision is related to the design of the shared activity space. In Synergo a mixture of alternatives is provided. A strict WYSIWIS (what you see is what I see) is allowed in the shared problem-solving window. This is because most of communication and reasoning is based on this shared viewpoint, which becomes the main grounding mechanism of dialogue and through which eventually common understanding can occur. However all additional operations outside this shared workspace, e.g. relating to browsing of activity sheets and other auxiliary material, saving of the flow chart or using private activity windows, should be performed independently by partners involved, a model-level coupling approach according to Suthers (2001). This approach, also known as *relaxed*

WISIWYS, guarantees only that users will see the same semantic state of a shared model, but the views may be entirely different.

In Synergo a floor control coordination mechanism is included. This mechanism involves the notion of the Action Enabling Key, which is owned by one of the participants at any given time. This key owner can then act in the shared workspace, while the rest just observe this activity and make comments through the chat tool. This mechanism is supported by key request, key accept, key pass, key reject functions, which can be found in the Coordination Panel (see fig.1). Experiments with this floor control mechanism, see also (Fidas et al. 2001) and (Komis et al. 2002), demonstrate that it supports reasoning about action, as partners need to reason and negotiate during key requests. Synergo users may opt for this mechanism or may decide to act in the shared activity space with no specific floor control, in which case locking is effected at the level of the single entity.
In the frame of the collaborative use of Synergo, a *dialogue tool* has been integrated, shown at the right panel of fig.1, which is based on an instant messaging protocol, using the same point-to-point connection and protocol of the shared activity space. Through this, text messages are exchanged during collaborative problem solving. The chat tool is activated from the collaboration panel. The possibility of definition of dialogue openers is included in this tool, as shown in figure 2, however due to concerns related to the usability of such approaches in the case studies discussed here, such dialogue openers have not been used.

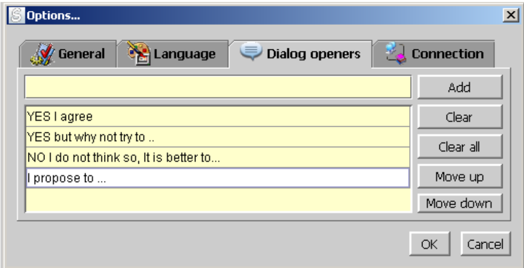

Figure 2. Examples of dialogue openers of the chat tool

Other means for exchange of text messages are the *sticky notes* as text containers positioned in the activity space, associated to existing objects, through which, gestures to them can be simulated.

An innovative feature of Synergo relates to analysis of collaboration activities. So a number of Analysis and Visualization tools are included in the environment. These are mainly used by the teachers and researchers, while limited versions of the tools may be used in some cases by students as meta-cognitive aids, as is the case of the level of collaboration monitoring display. The main functionality of the Analysis tool is the presentation and processing of logfiles, which are created during Synergo use. These logfiles contain actions and text messages of all partners, in sequential order. The logfiles are based on the format of the coordination and communication protocol and are stored in XML. These files can be viewed, commended and annotated by the researchers, using an adequate analysis framework, as discussed by Avouris et al. (2003a). A related functionality is the capability of the analysis environment of posterior reproduction of the modelling activity, using this logfile, in a step-by-step or continuous way. This is complementary to the logfile inspection and annotation functionality.

Modelling Collaboration

In this section we describe the key parameters through which we can model collaborative problem solving activity in Synergo. We suppose that the activity involves a small group of subjects (actors) who are engaged in collaborative problem solving (2 to 5 actors). Problem solving activity is usually considered as a process of refinement of abstract ideas ("abstract objects") and externalisation of these ideas in the form of parts of the solution to the given problem. Collaborative activity is based on communication, which takes the form of either direct communication acts or operations in the shared activity space. The activity is defined according to the following four dimensions:

– *The time dimension*
– *The actors' dimension*: actors, $A = \{A_1, A_2, ..., A_k\}$.
– *The objects' dimension*: $O = \{O_1, O_2, ..., O_\ell\}$. In the frame of the Synergo tool, a solution is considered as made of components (objects that compose the final solution), rejected components and abstract components
– *The typology of event dimension*: This is a dimension through which interpretation of the activity can take

place. We assume that there is an existing analytical framework, which defines this typology. If r is the finite number of expected event types, then we define a set $T = \{T_1, T_2, ..., T_r\}$ as the analytical framework of the study. While in the original OCAF proposal such a closed set T was included, (Avouris et al. 2003), in Synergo, we consider the method as independent of a specific analytical framework, so set T can be defined by the framework user.

Using the above four dimensions we can describe any given activity as a set of discrete non-trivial events produced by the actors. These define an ordered set of m events $E = \{E_1, E_2, ..., E_m\}$. Each one of these events is related to meaningful operations of the actors who interact with objects of set O. Each event is defined as a tuple $E_{i,AOT} = (t_t, A_A, [O_O], [T_T])_i$ where $i \in [1, m]$, t the event timestamp, A the actor who performed the operation of the specific event, O an optional parameter referring to the object of the specific operation and T an optional parameter which interprets the event according to the analysis framework T.

This is a useful model for ethnographic studies. Every time an event is produced by the actors, this is recorded and a history of such events, i.e. an ordered list of *Es* can be produced, as a result of such an activity. No mental or cognitive operators are defined, as these can be generated later as interpretations of the recorded activity. This model permits further analysis and interpretation of the activity, while quantitative indices of the activity can be easily produced or visualizations can be generated (Margaritis et al. 2004), as discussed in the next section.

Synergo adheres to a typology of generated events, thus automating the task of categorization of observed events (insertion, modification, deletion of primitive objects in the shared space and exchange of text messages), every time such an operation is recorded, this is automatically categorized according to the scheme of analysis defined by the user. OCAF suggests interpretation of exchanged messages (written dialogues during collaboration by distance), or recorded oral utterances (during face to face collaboration), in relation to operations towards "objects" of the activity space, using a language for action approach (Winograd 1987), defining a unifying framework for analysis of dialogue and action.

Quantitative indices of collaboration

Using the model of activity described above, a number of indices have been defined and accordingly presented in a visual form. Some of these indices relate to the density of occurrence of a type of event per time interval t_q, e.g. number of exchanged text messages per t_q, number of new objects in the shared space per t_q, etc.

One other kind of index is related to the degree of symmetry of activity in the group members. This index describes the relative contribution of the group members in a specific type of events. An example of an empirical index, called Collaboration Factor is described here. For instance, if we assume that N events of Actor A concern object O, then the contribution of Actor A to object O is measured as $AC_{AO} = W(A) \cdot \sum_{i=1}^{N} W(T_i)$, where *W(A)* is the relative weight of actor A και *W(T_i)* is the weight of type T_i of event i, that contributed to O history. The history factor *HF* of *O*, is defined as $HF_O = 1 - \frac{stdev(AC)}{\overline{M}\sqrt{k}}$, where $HF \in [0,1]$ and M is the mean value of the AC for object O. HF takes value around 1 when there is symmetrical contribution of all actors in the history of object O and around 0 when the object has been discussed and used by small section of the group. The collaboration factor of object O is defined subsequently, as $CF_O = HF_O \cdot W_O \cdot \frac{L(OE_O)}{m}$, $CF_O \in [0,1]$, where W_o the relative weight of object O in the model, $L(OE_O)$ is the length of action events of object O and m the total number of action events in E. Finally the collaboration factor of the modeling activity CF is defined as the mean value of all components' collaboration factors, including the abstract objects, or objects that were discussed and later rejected: $CF = \frac{\sum_{i=1}^{\ell} CF_{O_i}}{\ell}$, $CF \in [0,1]$.

This parameter, in addition to other indices like the density of activity of specific type of action events per time unit, can produce views of the activity that can lead to understanding of the collaboration dynamics, as discussed in the following section.

A case study of analysis of collaboration with Synergo

In this section we describe an example of a study that involved analysis of collaborative activity using the Synergo tool. The activity involved building of a concept map of an Internet service (an electronic bookshop was chosen as the example of the service to be model by the participants in this case) by small groups of students of an undergraduate University course, in the frame of one lab session (45’). We focus on one of these groups made of 4 students in this section. The logfile of the activity of this specific group was studied using the Synergo. More details of this study can be found in Avouris et al. (2004). First the relative weights of the activity types and the actors were defined, as seen in figure 3(a). In our case events related to creation and modification of sticky notes are assigned lower weight (0.3), as they are used for administration issues.

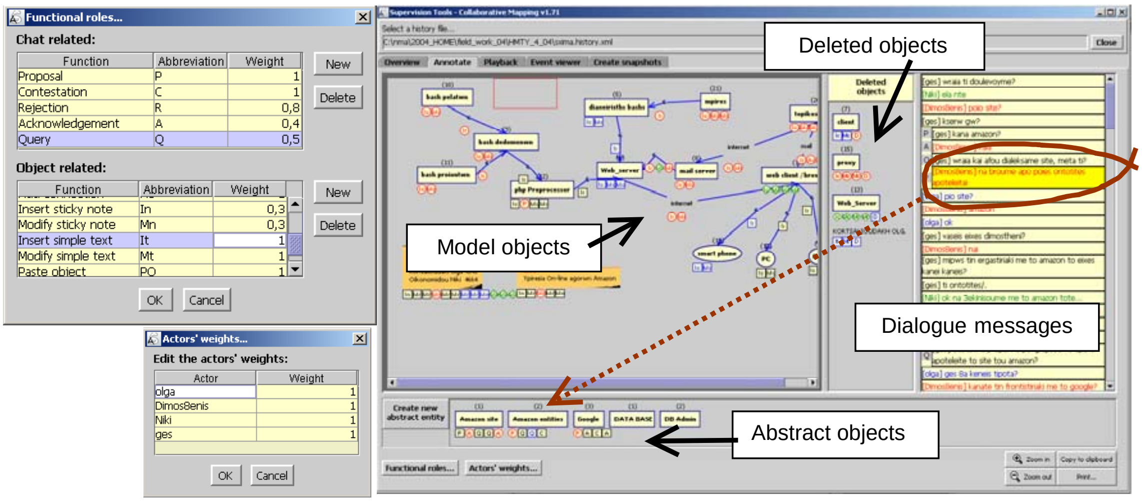

Figure 3. (a) Definition of activity type *W(T)* and actors weights *W(A)* and (b) annotation of dialogue events

Subsequently the dialogue events were annotated according to the defined typology. This phase involved definition of abstract entities that appeared in the dialogue. The dialogue annotation window is shown in figure 3(b). Three types of objects are shown in this window: the components of the final solution in the main panel (model objects), the deleted components in the vertical panel and the abstract components at the bottom panel. In the example of fig.3(b) a dialogue event is associated to the abstract object “Amazon model”: Actor *Ges* said: ”*what to assign to the Amazon site?*”, This dialogue message was categorized as a Q (Query) and was associated to the abstract object “*Amazon model*”, by a simple drag operation.

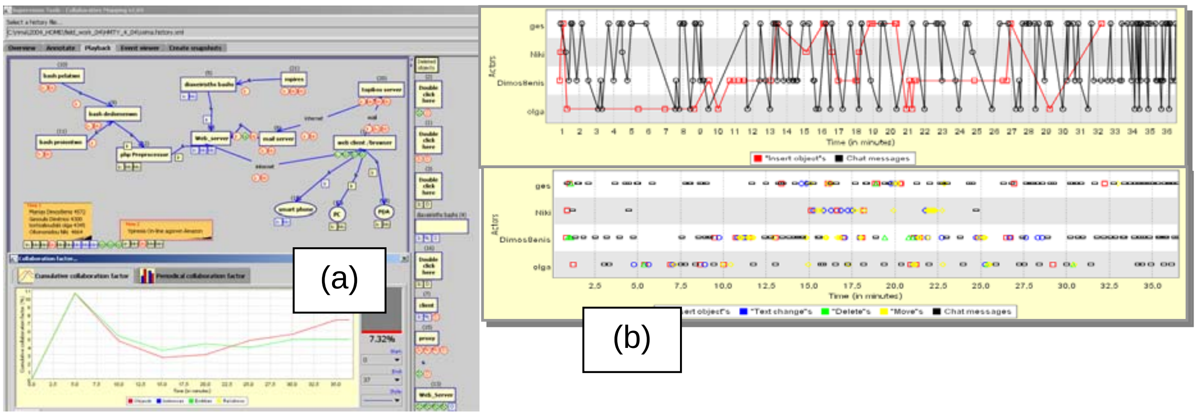

Figure 4. Visualization of collaboration indices (a) Collaboration Factor, (b) Evolution of Actor activity

After annotating dialogue events, we are able to playback the activity and produce in numeric and visual form the evolution of the Collaboration Factor. This is shown in figure 4(a). Some other indices, like the density of actors activity of various types in the shared activity space, can be produced automatically, from the Synergo logfile. Also the contribution of each actor in the activity can be visualized. In figure 4(b) the actor contribution of “insert object” events and chat messages is shown. Each line of these diagrams represents one of the four group members. From this picture, it is deduced that the second actor shows relatively low activity. More complex indices like the Collaboration Factor discussed here, are produced as a result of interpretation of actions and dialogue events. An example is the visual representation shown in fig.4(a). This provides an indication of the degree of collaboration of the group of the four students as they are building the e-shop concept map. From this graph it seems that while for the first period of the activity the degree of collaboration was high, subsequently the partners became more individualistic, working on parts of the solution, as also shown in the annotated concept map of fig 4(a). Later on towards the end of the session, there is more interaction, at the level of specific concepts and entities, the final value was CF=7,32%.

Conclusions

The innovative nature of Synergo is related to its capacity of monitoring and visualizing activity both of action and dialogue events using a unified framework, implementing the OCAF analysis framework. Dialogue events are assumed to be related to abstract of concrete objects of the solution. Thus the notion of history of objects creation is defined. The researcher using Synergo can define the analysis scheme in terms of types of events, and their relative weight. Also weights are associated to specific group members, so for instance the tutors as members of a group can be assigned with different weights than the students. A number of quantitative indices are calculated by the tool and can be visualized during playback of the activity. Also an intuitive environment for annotation of dialogue events is included which permits categorization of the exchanged messages according to the defined typology and association of them to objects of the solution. The proposed model of interaction in Synergo is used for visualization of indices and support of actors and analysis. No attempt has been made to relate this model to automatic supporting and scaffolding of interaction, as these approaches usually move the locus of control of activity from the user to the system, reducing usability and acceptability of the environment. The Synergo tool has already been effectively used for analysis of interaction of collocated small groups of students (Voyiatzaki et al. 2004) and of distant groups in the context of a course of distant learning (Xenos et al. 2004). It is believed that this kind of environment can facilitate and advance our understanding of the mechanics of collaboration of small groups of students, as micro-scale patterns of interaction and solution building can emerge. This understanding can facilitate support of the activity by tutors or by the environment itself at run time.

References

Avouris N, Komis V., Margaritis M., Fidas K., (2004a), ModellingSpace: A tool for synchronous collaborative problem solving, Proc. AACE ED-Media, pp. 381–386, Lugano, June 2004.

Avouris N., M. Margaritis, V. Komis, (2004b). The effect of group size in synchronous collaborative problem solving activities, Proc. ED Media AACE Conf., pp. 4303-4306, Lugano, June 2004.

Avouris N., V. Komis, M. Margaritis, G. Fiotakis, (2004c) An environment for studying collaborative learning activities, Journal of Technology & Society, 7 (2), pp. 34-41, April 2004 .

Avouris N.M., Dimitracopoulou A., Komis V., (2003), On analysis of collaborative problem solving: An object-oriented approach, Computers in Human Behavior, 19, (2), March 2003, pp. 147-167.

Bertelsen O.W., Bodker S., (2003), Activity Theory, in J. M Carroll (ed.), HCI Models, Theories and Frameworks, Morgan Kaufmann, 2003.

Dix A., Finlay J., Abowd G, Beale R., (1998), Human-Computer Interaction, Prentice Hall.

Fidas C., Komis V., Tzanavaris S., Avouris N., (2004), Heterogeneity of learning material in synchronous computer-supported collaborative modeling, Computers & Education, (in press).

Jermann, P., Soller A. & Muhlenbrock M. (2001) "From mirroring to guiding: a review of the state of the art technology or supporting collaborative learning". In Proceedings EuroCSCL´2001, Maastrich pp. 324-331.

Komis V., Avouris N., Fidas C., (2002), Computer-Supported Collaborative Concept Mapping: Study of Synchronous Peer Interaction, Education and Information Technologies, 7:2, 169–188.

Margaritis M., Avouris N., Komis V., (2004), Methods and Tools for representation of Collaborative Learning activities. Proc. ETPE 2004, September 2004, Athens.

Martinez A., Dimitriadis Y., Gomez E., Rubia B., De la Fuente P., (2003), Combining qualitative and social network analysis for the study of classroom social interactions, Computers and Education, 41, (4), pp. 353-368

Muelenbrock, M. & Hoppe, U. (1999), Computer Supported Interaction Analysis of Group Problem Solving. C.Hoadley & J.Roschelle (Eds). In Proc. CSCL 1999; Dec 12-15; Stanford University, Palo Alto, California. Mahwah, NJ: Lawrence Erlbaum Associates; pp. 398-405.

Suthers D. 2001, Architectures for Computer Supported Collaborative Learning. In proceedings of the IEEE International Conf. on Advanced Learning Technologies (ICALT2001), 6-8- Aug. 2001. Madison, Wisconsin.

Voyiatzaki E., Christakoudis C., Margaritis M., Avouris N., (2004), Algorithms Teaching in Secondary Education: A collaborative Approach, Proc. ED- Media 2004, pp. 2781-2789, Lugano, June 2004.

Winograd T., (1987). A Language/Action Perspective on the Design of Cooperative Work, Human-Computer Interaction 3:1 (1987-88), 3-30.

Xenos M., Avouris N., Komis V., Stavrinoudis D., Margaritis M., (2004), Synchronous Collaboration in Distance Education: A Case Study on a CS Course, *Proc. IEEE ICALT 2004*, Joensuu, FI.