Protocols of Coordination and Structured Communication in Synchronous CSCL Environments

Crescencio Bravo

Computer Engineering School, Department of Information Technologies and Systems, University of Castilla – La Mancha
Crescencio.Bravo@uclm.es

1 Motivation

Software Engineering studies the principles and methodologies for the development and maintenance of software systems. The Collaborative Systems (CS) are a kind of software system that are particularly complex to develop, especially when they support distributed synchronous collaboration. For example, it is difficult to carry out an exhaustive test because of the multiple interactions of different types that occur between the users. In this situation, the use of tested components facilitates development and provides quality. On the other hand, the software patterns facilitate the collection and application of experience. A pattern is a valid *solution* to a common *problem* in a specific context. Components and patterns follow the reuse principle and allow the improvement of software development.

The CSCL (Computer-Supported Collaborative Learning) and CSCW (Computer-Supported Cooperative Work) are the two scientific areas that make up the CS. In [4] Collis identifies the main contributions of the investigations in CSCL and CSCW. For instance, some essential characteristics of CSCL that differentiate it from CSCW are the influence of the teacher, the importance of structuring the group activities and of establishing shared aims, and the utilization of theoretic frames and computer support based on constructivist and cognitive learning theories.

Some of the teaching-learning methods that are usually used in CSCL are Problem-Based Learning, Project-Based Learning and Learning by Design. Taking these methods as a frame of reference, a CSCL environment allows the users to carry out an activity such as solving a problem, developing a project or designing an artefact. In the collaborative learning underlying these activities, two kinds of interactions take place: negotiation and argumentation [5]. Communication, coordination and decision-making are tasks which are necessary for implementing the negotiation and argumentation processes.

The contribution of this article consists of the proposal of a set of collaboration protocols for coordination and decision-making processes and for structured communication. These protocols make up a collection of design patterns to be used in the development of synchronous CSCL environments. Examples of these processes in different systems will be shown, in order to exemplify the use of such protocols. These examples represent proposals of user interfaces, and therefore constitute patterns for human-computer interaction (HCI).

Different authors have proposed pattern systems for HCI [6], although these patterns do not approach collaborative interaction. Although patterns for Groupware do exist, they do not consider CSCL aspects. In [8] the authors formalize collaborative learning patterns by means of IMS Learning Design. Nevertheless, the general patterns these authors propose do not approach specific coordination processes and they do not deal with user interface aspects. Therefore, there are no specific collections of design and interaction patterns for CSCL environments.

Support structured methods in CSCL are presented in the next section. In Section 3, different protocols for structuring coordination processes are shown, including decision-making and communication, and some implications regarding the user interface are discussed. Finally, the article concludes with an analysis of the protocols proposed and outlines the future lines of work.

2 Structuring the Support for Collaborative Learning

In Collaborative Learning, two support methods are distinguished [9]:

- Global methods that structure the collaboration at general level.
- Structured or guided methods of collaborative learning that provide protocols to structure the learners' dialogue and actions.

In [4] Collis also highlights the importance of structuring learning activities. The Learning Protocols (or Collaboration Protocols) are an integrated learning support that describes how to carry out and apply constraints, rules and methods of structuring of the processes in a CSCL environment [11]. The collaboration protocols, understood as guided methods of collaborative learning, are based on the theories of the psychological scripts. In cognitive and social psychology, the general knowledge regarding a routine sequence of related events is commonly known as a script. According to this approach, the collaboration protocols are a set of useful scripts for collaborative learning, expressed as executable methods with roles, events and explicit actions.

The collaboration protocols can be represented by means of State Diagrams. Each state represents the execution of a task in a shared workspace. In each workspace the users carry out actions such as communicating or manipulating artefacts. The transition to other workspaces is triggered by means of the users' actions or when a specific situation is reached, such as the occurrence of a time-out or when the artefact reaches a certain condition.

3 Coordination, Decision-making and Communication Protocols

In order to describe collaboration protocols, several formalisms can be used: the extended Activity Diagrams of AMENITIES [7], State Diagrams, etc. A Conversational Graph [1] allows the contribution types issued by the users and the relationships between these contributions to be shown. The graph nodes represent the contributions and the arcs, which are directed according to which contributions reply to a given contribution, thus determining the order of emission. In a conversation there are initial, final and intermediate contributions. In addition, contributions issued by the users or automatically issued by the system can be distinguished. The former are represented with a rectangle, and the latter with a rectangle with rounded corners (see Fig. 1). However, conversational graphs only show the contribution types and their relationships, so that more detailed techniques, such as State Diagrams, are required to express the working of the protocol.

Protocols for coordination, decision-making and communication are presented in subsequent subsections. The final subsection discusses user interfaces to be used to support these protocols.

3.1 Processes Based on Proposals and Agreements

In this type of processes, a leading student makes a proposal of a value for a variable, of the execution of an action, etc. The remaining students in the group have to reply to this proposal with an agreement, disagreement or abstention. This procedure is described by the conversational graph in Fig. 1.

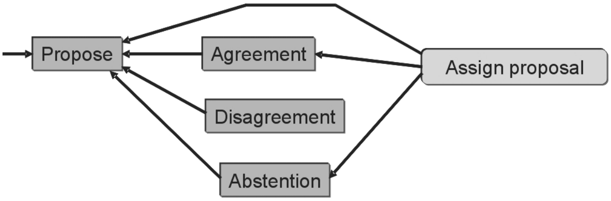


Fig. 1. Conversational graph of a process based on proposals and agreements

This scheme represents a process for a single proposal. However, it is common for users to carry out many proposals in parallel during a task. In this case, in order to reply to a proposal this has to be identified previously. A more detailed conversational graph would allow, for instance, a proposal to be replied to with another proposal, that is, a counterproposal. However, these more complex structures are more typical of asynchronous collaboration processes, in which the users have more time to think and to select the most appropriate conversational act at any moment.

3.2 Request and Release of Resources

In this scheme, the users request resources, and take control of them if they are available. However a resource can only be owned by one user, and cannot pass to another user until it has been released by the user who had control over it. Frequently, the assignment of resources is made by order of request. A typical situation in which this process is applied is when there is a floor control which is not managed by the system and that must be negotiated among the users. The conversational graph of this protocol (Fig. 2) contains two user contributions (request and release) and one system contribution (assign).

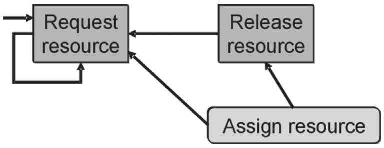


Fig. 2. Conversational graph of a process to request and release a resource

3.3 Democratic Processes

A democratic process avoids replying to contributions. In this protocol the system has to calculate the value to be assigned to a variable as the users make proposals. The system should have a configurable method which defines how to obtain the variable value from the values proposed and depending on the variable type. In the DomoSim-TPC environment [3], the Parameter Definition tool (see Fig. 4) uses a democratic process. This tool manipulates two kinds of variables: numeric and alphanumeric. Numeric variables are calculated as the arithmetic mean of all the values proposed, whereas for alphanumeric ones the most voted value is taken.

3.4 Decision-making Processes

The three coordination processes presented above are characterized by generating a system action as a result of adopting a specific proposal or negotiating the control of a resource. On the contrary, a structured decision-making process is a more guided process that is implemented as a voting process. In a voting process, a user makes a voting definition, then this user and his/her partners vote (the users can issue an abstention vote), and finally the system counts the voting results and shows them to the group. In so doing the users are responsible for using the results to carry out some action, and therefore a system action is not generated automatically.

A voting process consists of formulating a question for the users. There are three types of voting procedures that correspond to three types of questions: those which have a yes-no vote as an answer, those which have a numeric value as an answer, and those which have a value from a list of possible values as an answer.

3.5 Structured Communication

Many synchronous CSCL environments incorporate communication functionalities in the supported tasks. Communication is another means for negotiation and argumentation. In this situation, a structured communication interface can be an excellent complement, presenting three potential advantages:

- It provides the explicit representation of certain communication acts (for instance, the act *why…?*) that encourage learners' participation.
- It can reduce the writing load and facilitate coordination, allowing the learners to focus more on the task and on the reflexive interaction.
- It avoids problems of understanding the natural language.

The language underlying structured communication takes the form shown in the conversational graph in Fig. 3. There are initial messages (*i*, *k*) and messages that reply to others (*j*, *l*, *m*). There are also messages that can have more than one reply (*i*), and communication cycles are possible (*i*, *l*, *m*, *i*, …).

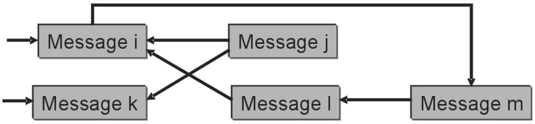
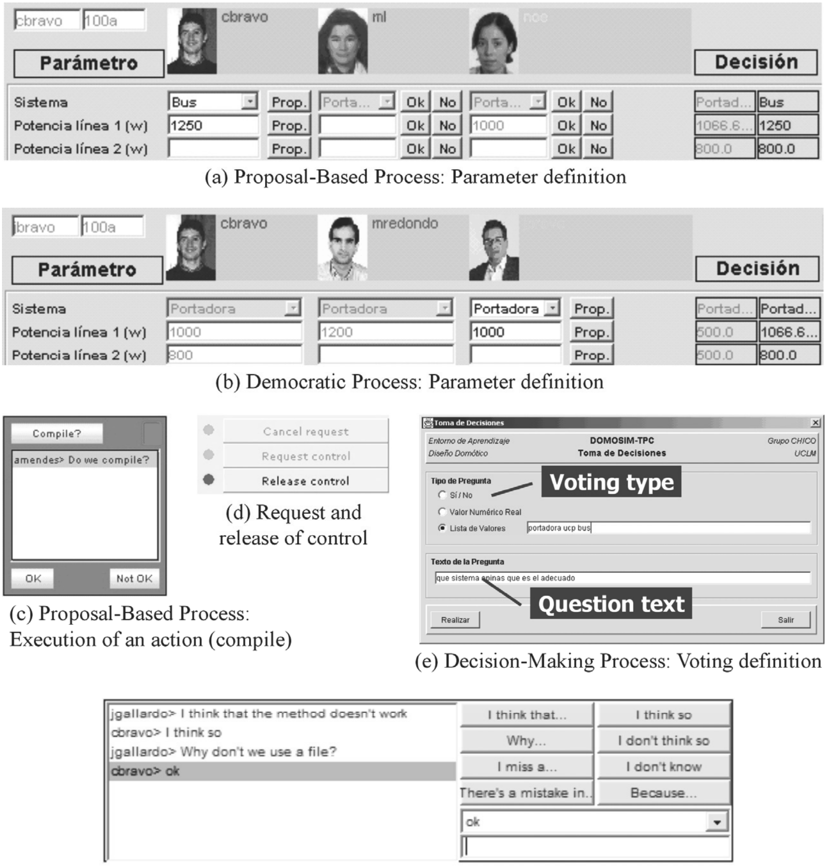


Fig. 3. Conversational graph of a generic structured communication

3.6 User Interfaces for Coordination, Decision-making and Structured Communication

In this section, some example user interfaces to support the protocols presented in the previous subsections are shown. Fig. 4 (a) shows the Parameter Definition tool of DomoSim-TPC [3], which allows the learners to give values to different variables by means of proposals, agreements and disagreements. There is a row for each parameter. The user interface supports multiple proposals, from several users and for several parameters simultaneously. The panel in Fig. 4 (c) allows the students to decide when to compile. This supports one proposal from each user. The arrangement of this user interface is different to the previous one: all the contributions are inserted in a list. Thus, the contribution to which a reply is to be made has to be selected from a list.

In Fig. 4 (b), the Parameter Definition tool of DomoSim-TPC is shown again, but in this case the tool is configured to support a democratic process, so that agreement and disagreement contributions are not needed. Fig. 4 (d) shows the user interface to manage the floor control in Co-Lab [10]. There are buttons to request and release control as well as to cancel a request. The decision-making process is implemented in the Voting Tool of DomoSim-TPC [3]. Fig. 4 (e) is an example of definition of a *list-of-values* voting type. The COLLEGE system [2] incorporates a structured chat (Fig. 4 (f)) implementing the structured communication protocol. The tool controls the correct use of the messages according to the communication structure. For example, the messages *I think so* or *I don't think so* can only reply to a message *I think that…*



Sistema	Bus	Porta...	Porta...	Portad...	Bus
Potencia línea 1 (w)	1250		1000	1066.6...	1250
Potencia línea 2 (w)				800.0	800.0

(a) Proposal-Based Process: Parameter definition

Sistema	Portadora	Portadora	Portadora	Portad...	Portad...
Potencia línea 1 (w)	1000	1200	1000	500.0	1066.6...
Potencia línea 2 (w)	800			500.0	800.0

(b) Democratic Process: Parameter definition

amendes> Do we compile?

(c) Proposal-Based Process: Execution of an action (compile)

(d) Request and release of control

Lista de Valores: portadora ucp bus

Texto de la Pregunta: que sistema opinas que es el adecuado

(e) Decision-Making Process: Voting definition

jgallardo> I think that the method doesn't work
cbravo> I think so
jgallardo> Why don't we use a file?
cbravo> ok

(f) Structured Comunication: Structured chat

Fig. 4. User interface examples for some collaboration protocols

The example user interfaces in Fig. 4 constitute a collection of interaction patterns, since they involve a solution for the problem of designing a user interface for a coordination, decision-making or communication process. However, several solutions to designing a user interface for such processes exist. The following are some ideas which are put forward for analyzing which user interface elements (controls) are more suitable to support a specific collaboration protocol and how to issue their corresponding contributions.

- Proposals: Usually, buttons are used to make proposals. The data set that characterizes a proposal (variables and values) must be defined using other user interface controls, such as check boxes, lists or text fields.

- Agreements, disagreements and abstentions: A button is the most intuitive and suitable control. However, other controls or strategies are required to identify which specific proposal to reply to. For instance, in Fig. 4 (a, b), the reply refers to the proposal of the corresponding row, and in Fig. 4 (c) the proposal is identified by selecting it from a list.
- Voting: To define voting it is necessary to use controls to depict their characteristics.
- Votes: The value to be voted can be selected by means of radio buttons or buttons, or be written in a text field. An abstention can be issued using a specific button.
- Structured communication: Buttons are typically used to send messages. An option that utilizes less space consists of using a pulldown list with the possible messages instead of buttons. In order to complete the messages that require additional text, text fields can be used.

Awareness facilitates the understanding of the other users' activities and provides a context for your own activity, reducing the effort required to coordinate tasks and resources, and allowing users to anticipate the other users' actions. Therefore, the perception of the contributions of the different protocols is crucial for the progress of argumentation and negotiation processes. The main awareness information included in the user interfaces in Fig. 4 is: information about the contributions (occurrence, type, data, issuing user, etc.), beeps and visual signals (see Fig. 4 (c)). Other elements external to these processes, such as panels showing the list of participants in the sessions, including their photos, help the users to reinforce their awareness.

4 Conclusions and Future Work

In this article a collection of collaboration protocols that allow the modelling of coordination, decision-making and communication processes in CSCL environments has been presented. In addition, examples of user interfaces to enable the necessary human-computer-human interaction to support these processes have been shown. These protocols are, in fact, patterns, since they represent a problem (to coordinate a task or collaborative action) and an effective solution (a coordination model) in a given context. More specifically, patterns for processes based on proposals and agreements, processes for request and release of resources, democratic processes, decision-making processes, and structured communication processes have been proposed. From the example user interfaces, guidelines from which patterns for HCI derive can be extracted. Each protocol and its user interface can be implemented by means of software components, so that by making a component-based development, CSCL environments could be constructed by composing building blocks (components).

The protocols proposed develop the principle of structuring. This structuring facilitates the storage of the students' tasks and actions, and of their interactions on the user interface, which allows the later analysis and study of the work carried out, which is a characteristic issue in CSCL.

With respect to the limitations of the protocols presented, they do not provide mechanisms to avoid both the lack of participation and the conflicts of intentions. From the point of view of software development, the need exists for methods for the modelling of collaboration processes, so that direct engineering is facilitated. To reach this objective, it is necessary to use formal description techniques to represent the protocols and the patterns without ambiguity, in order to facilitate their computational processing and to represent and apply previous experiences. In this respect, it will be necessary to link the elements of the user interface with the types of contributions and the working of the protocol. This is our current line of work.

Acknowledgments

This work is supported by the Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha (Spain), in the PCI-05-006 project.

References

1. Barros B (1999) Aprendizaje Colaborativo en Enseñanza a Distancia: Entorno Genérico para Configurar, Realizar y Analizar Actividades en Grupo. Ph.D. thesis, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid
2. Bravo C, Redondo MA, Mendes AJ, Ortega M (2006a) Group Learning of Programming by means of Real Time Distributed Collaboration Techniques. In: Lorés J, Navarro R (eds) HCI related papers of Interacción 2004. Springer (The Netherlands), pp 289–302
3. Bravo C, Redondo MA, Ortega M, Verdejo MF (2006b) Collaborative Environments for the Learning of Design: A Model and a Case Study in Domotics. Computers and Education 46 (2): 152–173
4. Collis B (1993) Cooperative Learning and CSCW: Research Perspectives for Internetworked Educational Environments. In: IFIP Working Group 3.3. Working Conference, Lessons from Learning, Theme B. Archamps, Francia
5. Dillenbourg P, Baker M, Blaye A, O'Malley C (1996) The Evolution of Research on Collaborative Learning. In: Espada E, Reiman P (eds) Learning

in Humans and Machine: Towards an Interdisciplinary Learning Science. Elsevier, Oxford, pp 189–211

6. Fincher S (2000) Patterns for HCI. Pattern gallery. http://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html
7. Garrido JL, Gea M, Rodríguez ML (2005) Requirements Engineering in Cooperative Systems. In: Requirements Engineering for Socio-Technical Systems. IDEA GROUP INC, USA, pp 226–244
8. Hernández D, Asensio JI, Dimitriadis Y (2004) IMS Learning Design Support for the Formalization of Collaborative Learning Patterns. In: Proceedings of 4th International Conference on Advanced Learning Technologies. Joensuu, Finland, pp 350–354
9. Mancini BM, Hall RH, Hall MA, Stewart B (1998) The Individual in the Dyad: a Qualitative Analysis of Scripted Cooperative Learning. Journal of Classroom Interaction 33 (1): 14–22
10. van Joolingen WR, de Jong T, Lazonder AW, Savelsbergh E, Manlove S (2005) Co-Lab: Research and Development of an On-line Learning Environment for Collaborative Scientific Discovery Learning. Computers in Human Behavior 21: 671–688
11. Wessner M, Hans-Rüdiger P, Miao Y (1999) Using Learning Protocols to Structure Computer-Supported Cooperative Learning. In: Proceedings of the ED-MEDIA'99, World Conference on Educational Multimedia, Hypermedia & Telecommunications. Seattle, Washington, pp 471–476

Springer

http://www.springer.com/978-1-84628-928-6

Computers and Education: Towards Educational
Change and Innovation
Mendes, A.J.; Pereira, I.; Costa, R. (Eds.)
2008, XVIII, 284 p., Hardcover
ISBN: 978-1-84628-928-6