FLOOR CONTROL IN COMPLEX SYNCHRONOUS CSCL ENVIRONMENTS

Jacques Lonchamp
LORIA, Nancy University, Campus Scientifique, BP 239, 54506 Vandoeuvre-les-Nancy Cedex, France

Keywords: CSCL, collaborative learning, floor control, turn management.

Abstract: Complex synchronous CSCL environments are dual space environments providing a task space where users "do things" through a set of collaborative tools and a communication space where users "talk of what they do". The most recent systems provide several tools in each space. In such complex dual space environments, the definition and role of floor-control (FC) has to be clarified. FC can be associated to different granularity levels: environment, space, artefact, component or attribute. If FC is associated to the tool or space level, the coexistence of different FC policies has to be considered. This paper discusses FC in complex synchronous CSCL environments and describes the particular approach implemented in Omega+ generic framework.

1 INTRODUCTION

Complex synchronous CSCL environments are dual space environments providing a task space where users "do things" through a set of collaborative tools and a communication space where users "talk of what they do". This combination of communication and shared work artefacts is an important requirement for effective collaborative learning (Suthers and Xu, 2002).

Early dual space CSCL systems only provided a chat tool in the communication space and some artefact editor in the task space. It was the case, for instance, of C-Chene (Baker and Lund, 1996), EPSILON (Soller et al., 1999), Digalo (Glassner and Schwarz, 2004), Coler (Constantino-Gonzáles and Suthers, 2001) and Cosar (Jaspers et al., 2001). In most of these environments (C-Chene, Digalo, Coler, Cosar), a floor control (FC) mechanism was associated to the task editor. The rationale was either to ensure exclusive access to the shared artefact (concurrency control), or to disallow anarchic interaction (turn management). In a few other proposals, all users were allowed to use the editor at any moment ("free floor policy"), and social protocols were expected to avoid inconsistent usages through mutual awareness information (e.g., visual feedback indicating which objects are in use). The FC approach has often been criticized from a theoretical point of view, when compared to more flexible techniques for consistency control, such as serializability, optimistic locking, operational transformation (Yang and Li, 2005). At the opposite, a number of field studies comparing argumentative activities with and without FC have found that FC increases the efficiency with regard to the collaborative task, thanks to better turn management and more meaningful discussions (Glassner and Schwartz, 2004), (McKinlay et al., 1993). The most recent synchronous CSCL systems provide several tools in each space. This is the case of Modelling Spaces (Avouris et al., 2004), Algebra-Jam (Singley et al., 2000), Cool Modes (Pinkwart, 2003) and Co-Lab (van Joolingen et al., 2005). In such complex dual space environments, the definition and role of FC has to be clarified. FC can be associated to very different granularity levels: environment, space, tool/artefact, component of the artefact, attribute of a component. If FC is associated to the tool or space level, several FC policies can coexist. The consequences of combining different FC policies have to be carefully analyzed.

The problem of FC is discussed here in the scope of the Omega+ effort for building a generic synchronous CSCL framework. Omega+ applies "model-based genericity" to the four dimensions of collaborative learning: the situation, the interaction, the process, and the way of monitoring individual and group performance (Lonchamp, 2006). These four aspects are explicitly specified in four models (process, protocol, artefact, effect) that serve as parameters for the generic framework which is

Lonchamp J. (2007).
FLOOR CONTROL IN COMPLEX SYNCHRONOUS CSCL ENVIRONMENTS.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Society, e-Business and e-Government / e-Learning*, pages 397-402
DOI: 10.5220/0001270103970402

designed to model systems that are flexible and can be tailored to a wide range of users, communities, goals and contexts. Omega+ client looks like a classical dual space system, with a communication space and a task space. The chat in the communication space is either a regular chat or a protocol-driven chat. A protocol model defines typed messages, role assignment, and message sequencing. The process model describes the sequential and/or parallel ordering of phases ("rooms") within the synchronous session. Each phase is characterized by an interaction protocol type, a FC policy, and a set of tools available in the task space. Tools are either predefined editors (collaborative text editor, whiteboard) or shared editors for graph-based representations which are customized by artefact models. Individual and collective group performance representations can be generated on the basis of the effect model. Omega+ approach of FC has to be sufficiently generic for accommodating a large spectrum of learning situations and sufficiently simple for allowing non specialist model designers to well understand the consequences of the choices specified in the process models they create or reuse.

Section 2 introduces the problem by defining the role of FC in complex synchronous CSCL systems, emphasizing its importance for user-oriented reasons and defining a set of global policies at the environment level. Section 3 discusses how these general ideas are implemented in Omega+.

2 THE PROBLEM

2.1 Understanding FC

Synchronous computer-mediated collaboration bears an inherent structure resembling a discourse model as it is known from linguistic pragmatics. A central concept is turn-taking, which is defined as the passing of speaker control among multiple participants. In pragmatic turn-taking models, at each turn a party assumes a social role such as speaker or listener, switching control in order to minimize pauses and maximize the conveyed information. The concept of transition relevance place (TRP) was introduced by Sacks et al. (1974). At each TRP, identified by syntactical constructs, control may eventually be switched. Duncan (1972) has proposed a model of turn-taking for face to face two-person conversations based on a description of the behaviour that accompanied speaking-role changes. Cues are actions which serve a signal's intent by being directly perceived and interpreted by the other participant as an expression of that intent. Many intuitive verbal cues (voice volume and rate) and non-verbal cues (gestures, eye contact, facial expressions) in face to face meetings are not valid in a computer-mediated environment. Floor control mechanisms are therefore introduced for facilitating turn management.

Synchronous collaborative environments are also concerned with how the learners' focus of attention relate, i.e., with mutual focus of attention. A person's focus of attention will correspond roughly to selections of one or more interface elements for immediate further processing, and such selections change from moment to moment. Other persons have to perceive and interpret these visual signals. FC is also concerned with helping to achieve such mutual focus of attention.

2.2 FC Policies

There are a surprisingly large number of different policies for FC. Policies are rules used for making decisions. The primary FC decisions concern how users acquire control, how users release control, and what happens to requests if control is not available (Myers et al., 1999). There are four options for control acquisition:

- *explicit request*: for instance by pushing a button,
- *implicit request*: by performing an input event, such as clicking the mouse or typing,
- *protocol-based*: for instance 'round robin', where each user gets a turn in a circular order,
- *designation*: a chair-person decides who gets control.

The three options for releasing control are:

- *explicit release*: the floor holder explicitly signals being finished,
- *idle pre-emption*: the system notices that the floor holder is not busy and releases the control,
- *explicit removal*: whether or not the user is finished, the control can be explicitly removed; for example, a moderator determines that the user has control for too long.

Finally, there are three options of what can happen to the requests:

- *immediate grant*: this only works with the explicit loss release policy,
- *queued*: usually in first-come, first-serve order,
- *ignored*: the request is thrown away if it cannot be satisfied.

By combining these options, most of the existing FC policies can be constructed: *free-floor* (implicit request + explicit removal + immediate grant), *pause*

detection (implicit request + idle pre-emption + ignored), *take the floor* (explicit request + explicit removal + immediate grant), *wait the floo*r (explicit request + explicit release + queued), *ask the floor* (explicit request + explicit release + ignored), *moderated* (designation + explicit removal + queued), etc.

2.3 Requirements

In a complex synchronous CSCL environment, FC is mandatory for user-oriented reasons. At the contrary of face-to-face conversations, a participant's attention might not be directed to a single other participant. If parallel activities may occur in different tools, mutual focus of attention can become very difficult to maintain. The danger of parallel activities from different learners is to obtain several threads of individual work, instead of a collaborative activity which is the fundamental objective of a CSCL system. However, a global floor at the environment level, ensuring the exclusive control of the whole system, is only one of the many possible solutions and in most cases not the optimal one. A complex (dual space) synchronous CSCL environment requires *a small set of global policies at the environment level, specifying who can talk and who can act*. For instance, a coarse grained form of parallelism, between a single active participant in the task space and several active commentators in the communication space, can be appropriate in many learning situations.

Each global policy can be characterized by the number of parallel floors available in the two spaces. Table 1 defines the five proposed global FC policies in terms of the corresponding space policies and illustrates each of them with a typical example. This proposal can deal with a large spectrum of learning situations. It is sufficiently simple for allowing non specialist process model designers to well understand the consequences of a given choice.

3 OMEGA+ IMPLEMENTATION

Some characteristics of Omega+ impact the implementation of these general ideas. As explained in the introduction, interaction models are used for customising the chat tool in the communication space. Interaction models allow describing complex policies for chat control based on application-related roles, typed messages and explicit sequencing rules. For example, some reviewing protocol defines two

Table 1: The five global FC policies.

Global policy	Task space policy	Comm. space policy	Typical example
Free floor	Free floor	Free floor	Free sketching + free commenting
Free talking-exclusive doing	Exclusive control	Free floor	Free commenting + exclusive diagramming
Free doing-exclusive talking	Free floor	Exclusive control	Free sketching + round robin talking
Parallel floors	Exclusive control	Exclusive control	Exclusive diagramming + exclusive voice channel
Common floor	Exclusive control		Round robin talking and diagramming

roles (writer and reviewer) and three message types (correction, supplement, comment). The rules specify that each reviewer contributes in turn with a correction, a supplement, or a comment. If a correction or supplement is provided, it is the writer's turn, who can accept or reject the proposed contribution. If a comment is provided, it is the next reviewer's turn (Pfister and Mühlpfordt, 2002). This kind of "protocol model-driven policy" may concern not only the communication space but also the task space, by using the communication floor as a *common floor* for the whole environment.

Omega+ provides an application-independent 'room operator' role. A participant playing this role has extended rights for dynamically changing most of the constraints that apply in the room (e.g., change the current interaction protocol, kick off or skip a participant, modify the ongoing process model) In Omega+ implementation, a default FC policy is proposed which achieves exclusive control at the space level. The proposed default policy is what was called above *wait the floor*, characterized by *explicit request*, *explicit release* and *FIFO queuing* options. This default policy may be dynamically customised by room operators. The definition of maximum idle time duration turns the default policy into a pre-emptive one. The capability to pass the floor to a specific participant turns the default policy into a moderated one. By this way, many different policies can be derived from the default policy. Apart the *implicit request* option (problematic when there are several tools), all the other options for control acquisition (*explicit request*, *protocol-based*, *designation*) and control

release (*explicit release*, *idle pre-emption*, *explicit removal*) become available as derivatives of the default policy or protocol model-driven policies. Table 2 summarizes Omega+ implementation of the global policies proposed in the previous section.

Table 2: Omega+ implementation.

Global policy	**Task space policy**	**Communication space policy**
Free floor	Free floor	Free floor
Free talking-exclusive doing	Wait the floor (customisable)	Free floor
Free doing-exclusive talking	Free floor	Wait the floor (customisable) or protocol-driven
Parallel floors	Wait the floor (customisable)	Wait the floor (customisable) or protocol-driven
Common floor	Wait the floor (customisable) or protocol-driven	

3.1 The User Interface

The background colour of all interactive areas shows if the user has the floor: a white area is ready to accept contributions from the user (such as typing, creating or modifying graphical elements) whereas a coloured (light blue) background means that interaction is impossible. The only exception concerns interacting through the annotation mechanism provided by Omega+: graphical pointers, "sticky notes", and "sticky annotated snapshot" are independent of the FC mechanism providing unconstrained means of interaction to the users (Lonchamp, 2007).

In the case of the default *wait the floor* policy, explicit control request and release are performed through dedicated buttons. When the "ask floor" button is pressed, it becomes greyed and its label changes to "waiting…" until the floor is received. At this time, the "release floor" button stops to be greyed and the background of the tool changes to white. With the "queue?" button, it is possible to know the FIFO queue state during the waiting state. Menu options allow a room operator to change the maximum idle time duration and to give the floor to a specific participant.

3.2 A Process Model Example

A process model, within a "structured room", defines a sequence of phase types. We differentiate between regular and split phases. In a regular phase the whole group of participants works in the same room. A split phase is a structured phase comprising a small set of sub phases running in parallel. The group of participants is divided into sub groups working in different sub rooms. Room Operators participate to all sub rooms. All sub phases of a split phase start and terminate simultaneously.

Each phase type (regular phase or sub phase) can be characterized by a name, a type (regular or split), an informal description, an interaction protocol type (either predefined – moderated, round robin, single contribution, unique contributor – or application-specific), a global FC policy (see Table 2), and a set of available shared tools (at most three). Each tool can be characterized by a name, a type (text editor, whiteboard, artefact editor), a read-only boolean, the path of the input file automatically loaded when the phase starts and the path of the output file automatically created when the phase terminates.

Omega+ favours visual modelling and model reuse. Collaborative model editing sessions use Omega+ generic editor customized for editing process models.

The OODesign process model has been created for an object-oriented design course. Small groups of three or four students receive the wording of a situation (see the read-only textboard in the middle part of the task space on the left of Figure 1) and have to build an UML class diagram. In the first phase, they are asked to specify some use cases with short textual descriptions (with the textboard in the upper part of the task space) and to draw the overall use case diagram (with the diagrammer in the lower part of the task space). This phase requires free talking like during a brainstorming. However, when producing the artefacts some coordination is required. Therefore, the *free talking and exclusive doing* FC policy has been selected. Figure 1 shows Jack's client who has the floor for doing (see the 'release floor' button on the top left and white backgrounds for all tools excepted the read-only textboard). All users (including Jack) can communicate with the regular chat tool (we present here simplified dialogues rewritten in English because the original course was given in French).

The second phase is the core of the design process. Students can see their use case descriptions in a read-only textboard (in the upper part of the task space in Figure 2). They translate them into collaboration diagrams (with the diagrammer in the middle part) while introducing new classes in the class diagram (with the diagrammer in the lower part). For ensuring both disciplined work and equality of participation among the students, the

'CircularWork' protocol is used as a common floor at the environment level. Each student in turn takes control of the whole environment. The predefined 'round robin' protocol has not been chosen because it implies that the floor moves to the next learner after each utterance. A specific protocol has been designed which allows sending several messages before passing explicitly the floor to the next learner.

Figure 3 shows Omega+ generic editor when the user has selected the 'Protocol Model' type and the 'CircularWork' protocol model. Such a protocol model includes a set of roles, a set of typed messages (utterances), and a set adjacency pairs (Clark and Schaefer, 1989) saying that if a user playing role A produces a message of type X then any user (or the next one, the same one, etc.) playing

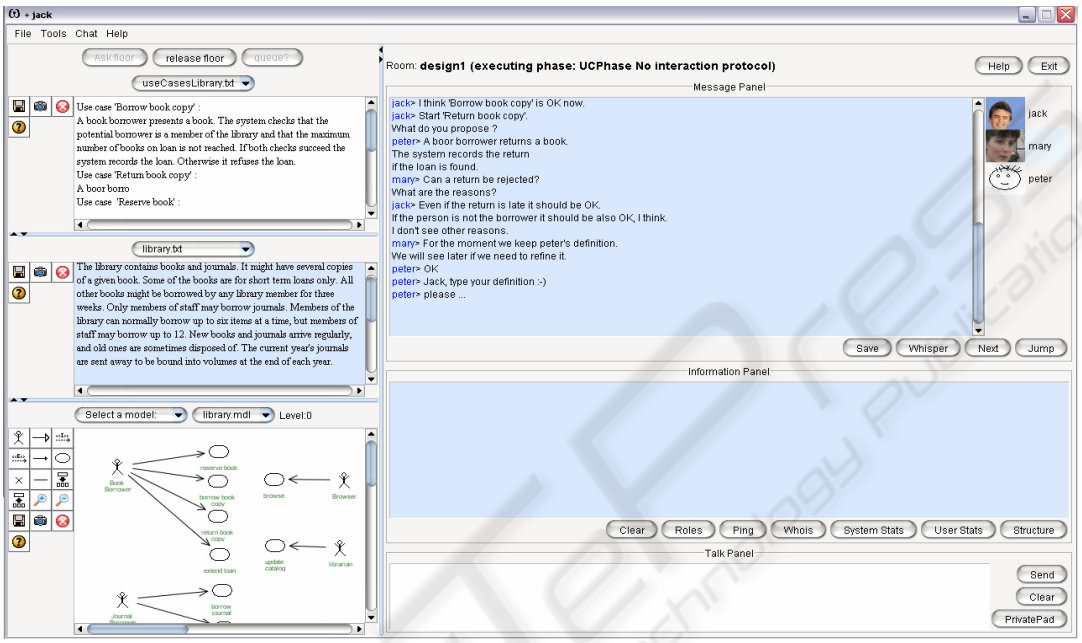

Figure 1: Jack's client during the first phase.

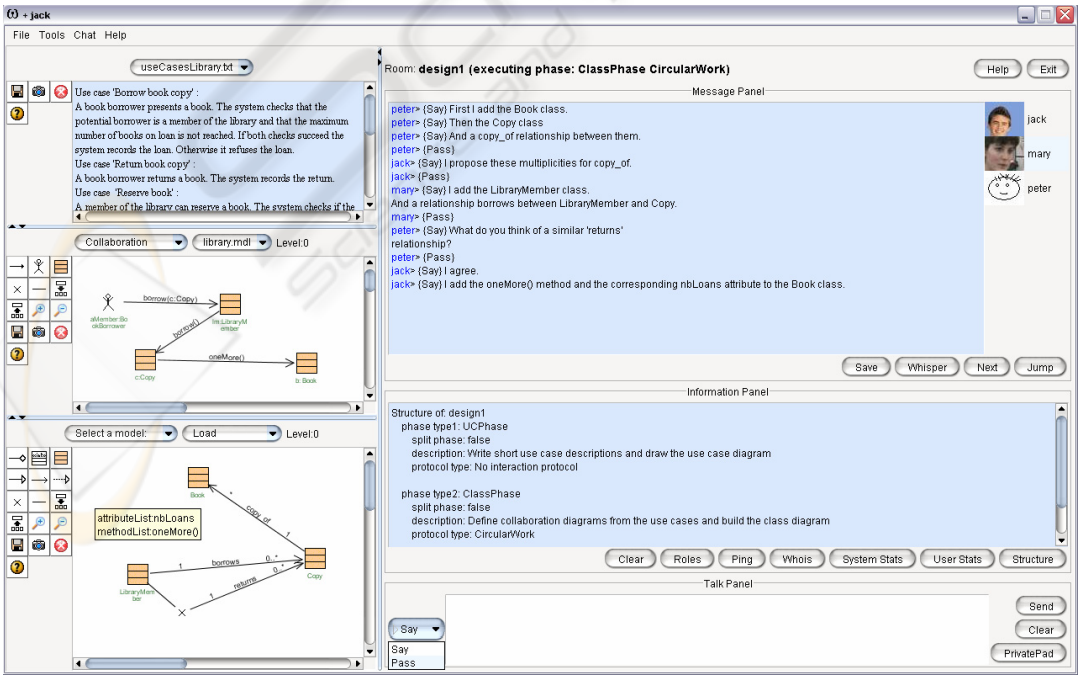

Figure 2: The protocol model-driven second phase.

role B can continue with a message of type Y. At each moment, a participant using the protocol-driven chat can only select through a combo box a type of message in accordance with his(her) role and the protocol rules ('Say' or 'Pass' for the floor holder, no message for the other students – see Fig. 2). The chat history also reflects the use of this protocol. At every moment, the room operator can switch to another FC policy like *wait the floor* or *free floor*.

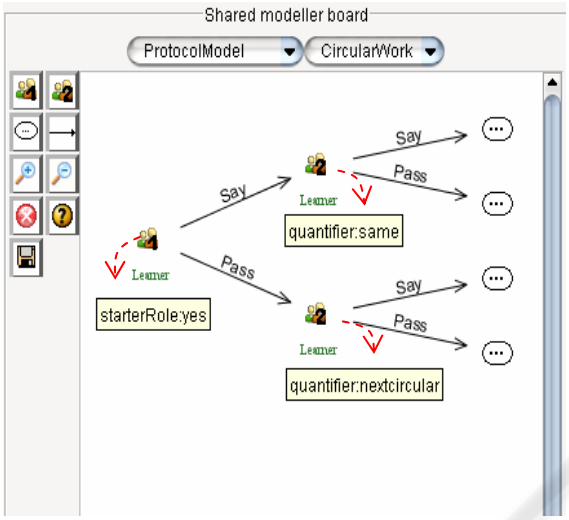

Figure 3: The 'CircularWork' protocol model.

4 CONCLUSION

Complex synchronous CSCL environments require a set of well defined FC policies at the environment level, specifying who can talk and who can act.

This paper proposes five global FC policies which should satisfy a large spectrum of learning situations. Omega+ implementation takes into account the existence of 'protocol model-based' policies for controlling the chat tool which have to be extended to the task space and the strong requirement for dynamic policy evolution by users playing the room operator role.

REFERENCES

Avouris, N., Komis, V., Margaritis, M., Fidas, C., 2004. Modelling Space: A tool for synchronous collaborative problem solving. In *Proc. ED-MEDIA' 04*, AACE Press, 381-386.

Baker, M., Lund, K., 1996. Flexibly structuring the interaction in a CSCL environment. In Proc. *European Conf. on Artificial Intelligence and Education*, Colibri Ed., 401-407.

Constantino-González, M., Suthers, D., 2002. Coaching Collaboration in a Computer-Mediated Learning Environment. In *Proc.CSCL'05*, Lawrence Erlbaum Associates, 583-586.

Duncan, S., 1972. Some Signals and Rules for Taking Speaking Turns in Conversations. *Journal of Personality and Social Psychology* 23, 283-292.

Glassner, A., Schwarz, B, 2005. The Role of Floor Control and of Ontology in Argumentative Activities with Discussion-Based Tools. In *Proc. CSCL'05*, Lawrence Erlbaum Associates, 170-179.

Jaspers, J., Erkens, G., Kanselaar, G., 2001. COSAR: Collaborative writing of argumentative texts. In *Proc. ICALT'01*, IEEE Computer Society Press, 269-272.

Lonchamp, J., 2006. Supporting synchronous collaborative learning: a generic, multi-dimensional model. *Int. Journal of CSCL*, 1 (2), 247-276.

Lonchamp, J., 2007. Linking Conversations and Task Objects in Synchronous CSCL Environments, *WEBIST 2007*.

McKinlay, A., Arnott, J., Procter, R., Masting, O., Woodburn, R., 1993. A Study of Turn-Taking in a Computer-Supported Group Task. In *Proc. HCI '93*, Cambridge University Press, 383-394.

Myers, B.A., Chuang, Y.S.A., Tjandra, M., Chen, M.C., Lee, C.K., 2000. Floor Control in a Highly Collaborative Co-Located Task. http://www.cs.cmu.edu/~pebbles/papers/pebblesfloorcontrol.pdf

Pfister, H.-R., Mühlpfordt, M., 2002. Supporting discourse in a synchronous learning environment: The learning protocol approach. In *Proc.CSCL'02*, Lawrence Erlbaum Associates, 581-589.

Pinkwart, N., 2003. A Plug-In Architecture for Graph Based Collaborative Modelling Systems. In *Proc. AIED'03*, IOS Press, 535-536.

Sacks, H., Schegloff, E. A., Jefferson, G., 1974. A Simplest Semantics for the Organization of Turn Taking for Conversation. *Language*, 50 (4).

Singley, M., Singh, M., Fairweather, P., Farrell, R., Swerling, S., 2000. Algebra jam: supporting teamwork and managing roles in a collaborative learning environment. In *Proc.CSCW'00*, ACM Press,145-154.

Soller, A., Linton, F., Goodman, B., Lesgold, A., 1999. Toward Intelligent Analysis and Support of Collaborative Learning Interaction. In *Proc. AIED'99, IOS Press*, 75-82.

Suthers, D. Xu, J., 2002. Kukakuka: An Online Environment for Artifact-Centered Discourse. In *Proc. 11th WWW Conference*, ACM Press, 472-480.

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.

Yang, Y.,Li, D., 2005. Supporting Adaptable Consistency Control in Structured Collaborative Workspaces. *Computer Supported Cooperative Work*, 14, 469–503.