The 3C Collaboration Model

Hugo Fuks
Catholic University of Rio de Janeiro, Brazil

Alberto Raposo
Catholic University of Rio de Janeiro, Brazil

Marco A. Gerosa
University of Vila Velha, Brazil

Mariano Pimental
Federal University of the State of Rio de Janeiro, Brazil

Carlos J. P. Lucena
Catholic University of Rio de Janeiro, Brazil

INTRODUCTION

Computational support for collaboration may be realized through the interplay between communication, coordination, and cooperation tools. Communication is related to the exchange of messages and information among people; coordination is related to the management of people, their activities and resources; and cooperation is the production taking place on a shared workspace. This model, which we call the 3C model, was originally proposed by Ellis, Gibbs, and Rein (1991), with some terminological differences. Cooperation, which Ellis et al. denominates "collaboration," here characterizes the joint operation in a shared workspace.

The 3C model appears frequently in the literature as a means to classify collaborative systems, for example as done by Borghoff and Schlichter (2000). However, a few attempts have been made to use it in the context of groupware implementation. An example is the Clover design model, which defines three classes of functionalities, namely communication, coordination, and production (Laurillau & Nigay, 2002; Calvary, Coutaz, & Nigay, 1997). These three classes of services appear in each functional layer of the model and, during the system design phase, they "must be identified and their access harmoniously combined in the user interface." The Clover model shares the same usefulness of the 3C model in terms of groupware functional specification, because both deal with the three classes of functionalities that a groupware application may support.

Given its complex interactive nature, groupware testing has not yet achieved its maturity. The 3C model may also help evaluators focus their attention on the communication, coordination, and cooperation aspects, guiding the detection of usability problems. A groupware evaluation approach based on a model similar to the 3C one is presented in Neale, Carroll, and Rosson (2004). Differently from the approaches found in the literature, we explore the 3C model as a means to analyze and represent a groupware application domain and also to serve as a basis for groupware development.

The relationship among the 3Cs of the model may be used as a guidance to analyze a groupware application domain. Groupware such as chat, for example, which is a communication tool, requires communication (exchange of messages), coordination (access policies), and cooperation (registration and sharing). Despite their separation for analytic purposes, communication, coordination and cooperation should not be seen in an isolated fashion; there is a constant interplay between them (Pimentel, Fuks, & Lucena, 2004).

For the sake of development, we propose the use of 3C-based components as a means of developing extendable groupware whose assembly is determined by collaboration needs. By conceiving the problem from the viewpoint of the 3C model and using a component structure designed for this model, changes in the collaboration dynamics are mapped onto the computational support. This way, the developer has a workbench with a component-based infrastructure designed specifically for groupware, based on a collaboration model.

INSTANTIATING THE 3C MODEL

Below we present three different groupwork domains that illustrate that the iterative nature of collaboration may be represented as cycles connecting the 3Cs.

We start with the groupwork domain represented in Figure 1. According to this instantiation of the 3C model, while communicating, people negotiate and make decisions. While coordinating themselves, they deal with conflicts and organize their activities in a manner that prevents loss of communication and of cooperation efforts. Cooperation is the joint operation of members of the group in a shared space, seeking to execute tasks, and generate and manipulate cooperation objects. The need for renegotiating and for making decisions about unexpected situations that appear during cooperation may demand a new round of communication, which will require coordination to reorganize the tasks to be executed during cooperation.

Considering media spaces (Mackay, 1999), which are multimedia-enhanced spaces aimed at informal communication among people, the 3C model may be instantiated according to Figure 2a. The media space itself is the shared space. Since it is aimed at informal communication, its main goal is actually to create opportunities for informal meetings, which are coordinated by the standing social protocol, for example, by accessing the availability of remote colleagues. These meetings generate conversation, which may occur using the media provided by the system or any other available means, such as telephones.

Another example is the family calendar (Figure 2b). The main reason for the family calendar is to schedule family activities. Modern family members have a variety of conflicting interests that can render last evening defined schedules ineffective next morning. In order to restore proper family coordination, negotiation among family members is needed. "This process involves seeing what has already been scheduled…and negotiating errand, ride, and other responsibilities are needed" (Elliot & Carpendale, 2005, p.4). The reconciliation obtained after the negotiation round is placed on the shared calendar. But as life never stops, next morning the cycle may start all over again.

These cycles show the iterative nature of collaboration. The participants obtain feedback from their actions and feedthrough from the actions of their companions by means of awareness information related to the interaction among participants (Gerosa, Fuks, & Lucena, 2003). This information mediates each of the 3Cs, which are detailed in the next sessions.

COMMUNICATION

The designer of a communication tool defines the communication elements that will set the communication channel between the interlocutors, taking into consideration the specific usage that is being planned for the tool (time, space, purpose, dynamics, and types of participants) and other factors such as privacy, development and execution restrictions, information overload, and so forth. Then, these elements are mapped onto software components that provide support to the specific needs.

The first communication element that must be considered is the choice of media. They can be textual, spoken, pictorial, or gestured—for example in a video

Figure 1. 3C collaboration model instantiated for group work

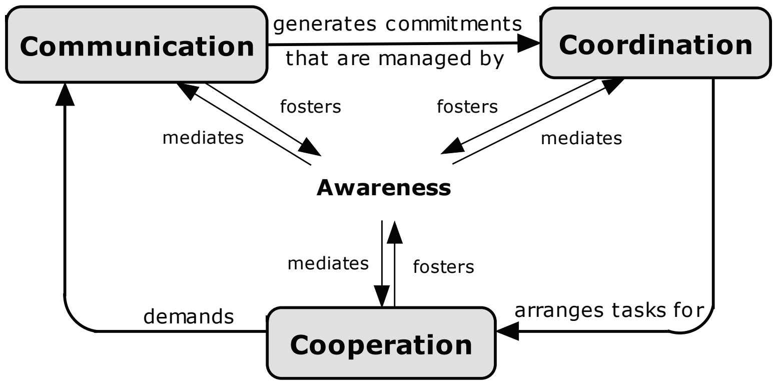
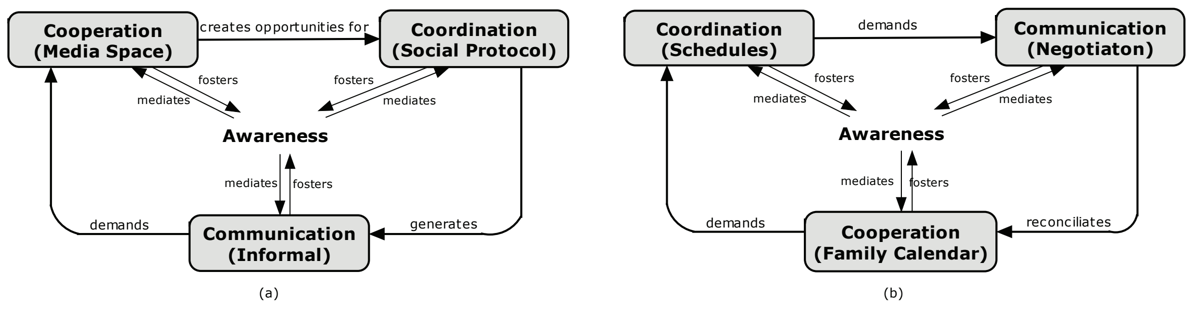


Figure 2. 3C collaboration model instantiated for the media space and for the family calendar domains



or avatar form. The media adopted restrict and influence the vocabulary used in the conversation, which is also influenced by the context.

The *transmission mode* defines whether the information is transmitted in blocks or continuously. In an audio or videoconference and in some chat tools the information is transmitted continuously as it is generated. In asynchronous and in other chat and messenger tools the information is transmitted in blocks: The author edits the message and it is only sent with an explicit command. Asynchronous communication tools are used when one wants to enhance reflection by the interlocutors, since they will have more time before they act, while synchronous tools are used for communication bursts. Restrictions policy is another communication element. For example, it is possible to restrict the text's size, characters allowed, bit rate (for video and audio), allowed vocabulary, and so on. These restrictions are used in some tools to reduce information overload and to save bandwidth.

Meta-information complements the information being transmitted in the body of the message. Common meta-information available in communication tools are the subject of the message, its date, priority, and category.

Conversation structure defines how messages are structured, which can be in a linear, hierarchical, or network form. The conversation structure makes the relations between messages, which are usually implicit within the text, visually explicit. The linear form is used when there are not so many message interconnections, the chronological order of the messages is relevant and fluency in the conversation is desired. Hierarchical structuring is appropriate when the relationships between messages, such as questions and answers, need to be quickly identified. However, as there is no way to link messages from two different branches, the tree can only grow wide and, thus, the discussion takes place in diverging lines (Stahl, 2001). Network structuring can be used to seek convergence in the discussion.

Conversation paths can be used to restrict the possible directions the conversation can take. Conversation paths formalize the conversation and it is not recommended when fluency is desired.

In order to provide proper support to communication, the designer should also take into account coordination and cooperation elements. Coordination elements deal with access policies to the communication channel, while cooperation elements deal with information rendering and registration.

COORDINATION

Coordination may be viewed as the link connecting the other two Cs in order to enforce the success of collaboration. This is more clearly observed when we analyze the elements that need to be coordinated, namely people, resources, and tasks. The coordination of people, for example, is deeply related to communication and context. The coordination of resources, on the other hand, is related to the shared space (i.e., cooperation). For this reason, coordination aspects also appear in the discussion about the other Cs of the model. In this section, we focus on the coordination of tasks. The management of the tasks being carried out consists in managing interdependencies between tasks that are carried out to achieve a goal (Malone & Crowston, 1990).

Some computer-supported collaborative activities, the so-called *loosely integrated* collaborative activities are deeply associated with social relations and are generally satisfactorily coordinated by the standing social protocol, which is characterized by the absence of any computer-supported coordination mechanism—"a specialized software device, which interacts with a specific software application so as to support articulation work" (Schmidt & Simone, 1996, p. 184)—among the tasks, trusting the users' abilities to mediate interactions. Coordination, in these situations, is contextually established and strongly dependent on mutual awareness. Through awareness information, the participants detect changes in plans and understand how the work of their colleagues is getting along (Dourish & Belloti, 1992). However, there are also the so-called *tightly integrated* collaborative activities, whose tasks are highly interdependent, as the name suggests. They require sophisticated coordination mechanisms in order to be supported by computer systems.

The great challenge of the designer in designing coordination mechanisms in groupware is to achieve flexibility without losing the regulation, which is necessary in some situations in which the social protocol is not enough. The system should not impose rigid work or communication patterns, but rather offer the user the possibility to use, alter, or simply ignore them. Thus, coordination flexibility and accessibility should be pursued by groupware designers. Flexibility is related to the possibility of dynamically allowing redefinition and temporary modifications in the coordination scheme. Accessibility is related to exposing the coordination mechanisms to system users rather than having them deeply embedded in the system's implementation.

Coordination may take place on the temporal and on the object levels (Ellis & Wainer, 1994). On the temporal level, coordination defines the sequence of tasks that makes up an activity. On the object level, coordination describes how to handle the sequential or simultaneous access of multiple participants through the same set of cooperation objects (Raposo & Fuks, 2002).

Communication and coordination, although crucial, are not enough. Given that coordination is required to manage the tasks; according to the 3C model it is also necessary to provide a shared workspace where cooperation will take place (Raposo, Gerosa, & Fuks, 2004).

COOPERATION

Cooperation is the joint operation during a session within a shared workspace. Group members cooperate by producing, manipulating, and organizing information, and by building and refining cooperation objects, such as documents, spreadsheets, artwork, and so forth. The shared workspace provides a number of tools for managing these artifacts, such as the recording and the recovery of previous versions, access control and permission. By recording the information exchanged, the group is able to count on collective memory, which can be consulted whenever necessary to recover the history of a discussion or the context in which a decision was made.

Production is dependent on how the shared workspace is structured to present the cooperation objects and the interaction that is taking place there. In a face-to-face situation, a large part of how we maintain a sense of who is around and what is going on is related to being able to see and hear events or actions with little conscious effort. On the other hand, in a computer-supported workspace, awareness support is less effective since the means for making information available to sensory organs are limited; however, irrelevant information can be filtered in a way that reduces distractions that usually affect face-to-face collaboration.

Individuals seek the awareness information necessary to create a shared context and to anticipate actions and requirements related to their collaboration goals. Thus, it becomes possible to interpret the intentions of the members of the group in such a way that one can provide assistance in terms of their work whenever it is convenient and needed (Baker, Greenberg, & Gutwin, 2001).

The designer of a digital environment must identify what awareness information is relevant, how it will be obtained, where it is needed and how to display it. Excessive information can cause overload and disrupt the collaboration flow. To avoid disruption, it is necessary to balance the need to supply information with care to avoid distracting the attention required to work. The supply of information in an asynchronous, structured, filtered and summarized form can accomplish this balance (Kraut & Attewell, 1997). The big picture should be supplied and individuals could select which parts of the information they want to work with, leaving further details to be obtained when required. There must also be some form of privacy protection. The shared space must be conceived in a way that group members could seamlessly move from awareness to work.

The register of group interactions is filed, catalogued, categorized and structured within cooperation objects. Ideas, facts, questions, points of view, conversations, discussions, decisions, and so on, are retrievable, providing a history of the collaboration and the context in which learning took place.

DESIGN AND IMPLEMENTATION ISSUES

The 3C collaboration model has been used as a basis for the development of the AulaNet Learning Management System. AulaNet is a freeware web-based environment for teaching and learning. It has been under development since June 1997 by the Software Engineering Laboratory of the Catholic University of Rio de Janeiro (PUC-Rio) (Fuks, Raposo, Gerosa, & Fuks, 2005).

The AulaNet environment's services are subdivided into communication, coordination and cooperation services, as can be seen in Figure 3. The communication services provide tools for forum-style asynchronous text discussion (*conferences*), chat-style synchronous text discussion (*debate*) (Fuks, Pimentel, & Lucena, 2006), instant message exchange between simultaneously connected learners (*instant messaging*), and individual electronic mail with the mediators (*message to participants*) and with the whole class, in a list-server style (*message to the class*).

Coordination services support the management and the enforcement of group activities. In AulaNet, coordination services include tools for notification (*notices*), evaluation (*tasks* and *exams*) as well as a tool that allows monitoring group participation (*follow-up reports*). Cooperation services in AulaNet include *Lessons* and *Documentation*, a list of course references (*bibliography* and Webliography) and course co-authoring support, both for teachers (teacher co-authoring) and for learners (learner co-authoring).

In developing groupware, the requirements are rarely clear enough to allow for a precise specification of the system's behavior in advance. Groupware development is evolutionary in the sense that it is difficult to predict how a particular group will collaborate and each group has highly distinct characteristics and objectives (Gutwin & Greenberg, 2000). By involving a group, the possibilities of interactions multiply and the demand for synchronism and solving deadlocks increases, posing problems in the construction of suitable interaction mechanisms and conducting tests.

This scenario is suitable for the application of component-based development techniques, which provide the flexibility needed in projects with changing requirements. Groupware services can be seen as groupware components that are plugged and unplugged from the system. The system's architecture comprises component frameworks that define overall invariants and protocols for plugging components.

AulaNet services were developed using a component-framework-based architecture, as can be seen in Figure 4. There is a common structure implemented by the collaboration framework, which defines the skeleton

Figure 3. Classification of AulaNet services based on the 3C model. The 3C triangle appears in Borghoff and Schlichter (2000).

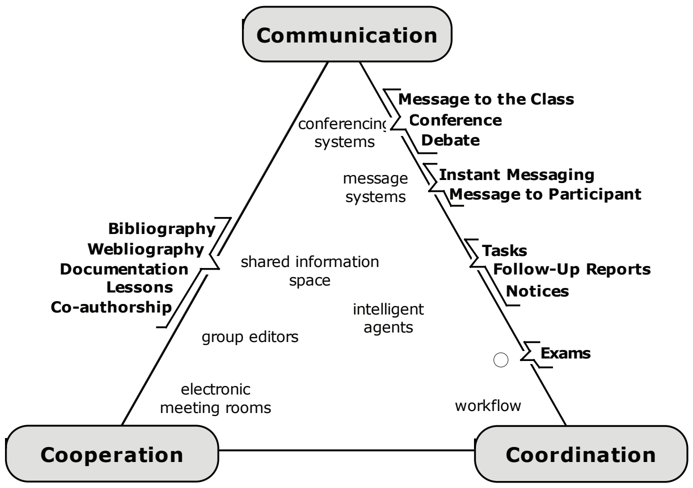
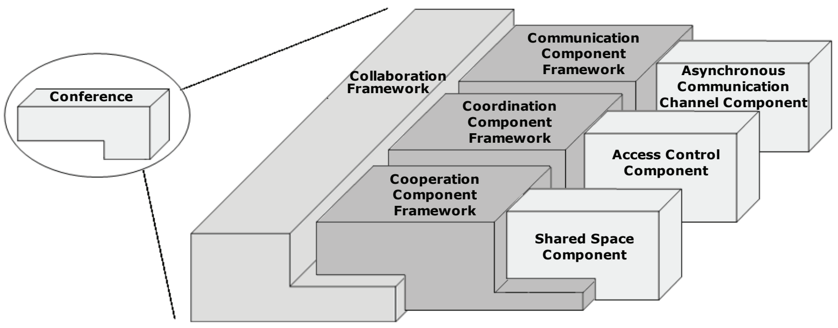


Figure 4. Architecture of a collaboration service


Conference
Collaboration Framework
Communication Component Framework
Asynchronous Communication Channel Component
Coordination Component Framework
Access Control Component
Cooperation Component Framework
Shared Space Component

of the services, and plugged to this framework there are the communication, the coordination and the cooperation component frameworks, which support each C of the 3C model. Class frameworks are used to implement components, which are plugged to the corresponding C-framework and implement the specific functionalities of the service (Gerosa et al., 2005).

For example, using the communication class framework, the developer implements components for synchronous and asynchronous communication, message transmission etc. Using coordination class framework, components for task management, participation follow-up, workflow, and so on are implemented. Using cooperation class framework, components for managing the shared space and its awareness elements, version management, among others are implemented.

CONCLUSION

Groupware systems are evolutionary because the composition and the characteristics of workgroups change with time, as well as the tasks that need to be executed. For this reason, even if a groupware designer is able to develop an “optimal” application for a group, it will eventually become inadequate due to new situations and problems that certainly will appear.

Ideally, groupware should be prototyped because collaborative systems are especially prone to failure (Grudin, 1989), hence demanding iterative evaluation during their development. However, given the excessive cost of throwing code away, as demanded by “pure” prototyping (Brooks, 1975), an incremental model can be considered more adequate, leading to the development of more advanced prototypes in the subsequent cycles.

In face of construction and maintenance difficulties, the groupware developer spent more time dealing with technical difficulties than moderating and providing support to the interaction among users. Such problems led to the need to create a quicker and more effective way to develop groupware in which low-level complexities resulting from distributed and multi-user systems were encapsulated into infrastructure components of the architecture. Besides, the concepts of the domain’s modeling should permeate all other activities and artifacts of the application’s development. This way, the modeling done during the domain analysis could be mapped to implementation, thus increasing productivity in groupware development and maintenance and making the applications more adequate to the users’ collaboration needs.

The 3C collaboration model defines three types of services that a groupware may support. The concepts and representation models described in this paper can be used to provide a common language for representing and describing the collaboration aspects of a workgroup and to guide the functional specification and the implementation of computational support for group work.

ACKNOWLEDGMENT

The AulaNet project is partially financed by Fundação Padre Leonel Franca and by the Ministry of Science and Technology through its Program Multi-Agent Systems for Software Engineering Project (ESSMA) grant nº 552068/2002-0. It is also financed by individual grants awarded by the National Research Council to: Carlos José Pereira de Lucena nº 300031/92-0, Hugo

Fuks nº 301917/2005-1, and Marco Aurélio Gerosa nº 140103/02-3. Thanks to Prof. Marcelo Gattass, Head of Tecgraf/PUC-Rio, a group mainly funded by Petrobras, Brazilian Oil & Gas Company.

REFERENCES

Baker, K., Greenberg, S., & Gutwin, C. (2001). Heuristic evaluation of groupware based on the mechanics of collaboration. In M. Little & L. Nigay (Eds.), *Proceedings of 8th IFIP International Conference (EHCI 2001)* (pp. 123-139), Lecture Notes in Computer Science Vol. 2254. Berlin: Springer-Verlag.

Borghoff, U. M., & Schlichter, J. H. (2000). *Computer-supported cooperative work: Introduction to distributed applications*. Berlin: Springer-Verlag.

Brooks, F. P. Jr. (1975). Plan to throw one away. In F. P. Brooks, Jr. (Ed.), *The mythical man-month: Essays on software engineering* (pp. 115-123). Reading, MA: Addison-Wesley.

Calvary, G., Coutaz, J., & Nigay, L. (1997). From single-user architectural design to PAC*: A generic software architectural model for CSCW. In S. Pemberton (Ed.), *Proceedings of the Conference on Human Factors in Computing Systems (CHI'97)* (pp. 242-249). New York: ACM Press.

Dourish, P., & Belloti, V. (1992). Awareness and coordination in shared workspaces. In J. Turner & R. Kraut (Eds.), *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW)* (pp. 107-114). New York: ACM Press.

Elliot, K., & Carpendale, S. (2005). *Awareness and coordination: A calendar for families* (Technical Report 2005-791-22). Calgary, Canada: University of Calgary, Department of Computer Science.

Ellis, C. A., Gibbs, S.J., & Rein, G. L. (1991). Groupware: Some issues and experiences. *Communications of the ACM, 34*(1), 38-58.

Ellis, C. A., & Wainer, J. (1994). A conceptual model of groupware. In T. Malone (Ed.), *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW)* (pp. 79-88). New York: ACM Press.

Fuks H., Pimentel, M., & Lucena, C. J. P. (2006). R-U-Typing-2-Me? Evolving a chat tool to increase understanding in learning activities. *International Journal of Computer-Supported Collaborative Learning, 1*(1), 117-142.

Fuks, H., Raposo, A. B., Gerosa, M. A., & Lucena, C. J. P. (2005). Applying the 3C model to groupware development. *International Journal of Cooperative Information Systems (IJCIS), 14*(2-3), 299-238.

Gerosa, M. A., Fuks, H., & Lucena, C. J. P. (2003). Analysis and design of awareness elements in collaboration digital environments: A case study in the AulaNet learning environment. *The Journal of Interactive Learning Research, 14*(3), 315-332.

Gerosa, M. A., Pimentel, M., Fuks, H., & Lucena, C. J. P. (2005). No need to read messages right now: Helping mediators to steer educational forums using statistical and visual information. In T. Koschmann, T. -W. Cha, & D. D. Suthers (Eds.), *Proceedings of Computer Supported Collaborative Learning* (pp. 160-169). Mahwah, NJ: Lawrence Erlbaum Associates.

Grudin, J. (1989). Why groupware applications fail: Problems in design and evaluation. *Office: Technology and People, 4*(3), 245-264.

Gutwin, C., & Greenberg, S. (2000). The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In R. D. Sriram & N. Shahmehri (Eds.), *Proceedings of the IEEE 9th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE* (pp. 98-103). Washington, DC: IEEE Computer Society Press.

Kraut, R. E., & Attewell, P. (1997). *Media use in global corporation: Electronic mail and organisational knowledge, research milestone on the information highway*. Mahwah, NJ: Lawrence Erlbaum Associates.

Laurillau, Y., & Nigay, L. (2002). Clover architecture for groupware. In E. F. Churchill, J. McCarthy, C. Neuwirth, & T. Rodden (Eds.), *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW)* (pp. 236-245). New York: ACM Press.

Mackay, W. E. (1999). Media spaces: Environments for informal multimedia interaction. In M. Beaudouin-Lafon (Ed.), *Computer supported co-operative work: Trends in software* (Vol. 7, pp. 55-82). Chichester, England: John Wiley & Sons.

Malone, T. W., & Crowston, K. (1990). What is coordination theory and how can it help design cooperative work systems? In F. Halasz (Ed.), *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW)* (pp. 357-370). New York: ACM Press.

Neale, D. C., Carroll, J. M., & Rosson, M. B. (2004). Evaluating computer-supported cooperative work: Models and frameworks. In J. Herbsleb & G. Olson (Eds.), *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW)* (pp. 112-121). New York: ACM Press.

Pimentel, M., Fuks, H., & Lucena, C. J. P. (2004). Mediated chat 2.0: Embedding coordination into chat tools. In F. Darses, R. Dieng, C. Simone, & M. Zacklad (Eds.), *Conference Supplement of the 6th International Conference on the Design of Cooperative Systems* (pp. 99-103). Amsterdam: IOS Press.

Raposo, A. B., & Fuks, H. (2002). Defining task interdependencies and coordination mechanisms for collaborative systems. In M. Blay-Fornarino, A. M. Pinna-Dery, K. Schmidt, & P. Zaraté (Eds.), *Cooperative systems design*: *Frontiers in artificial intelligence and applications* (Vol. 74, pp. 88-103). Amsterdam: IOS Press.

Raposo, A. B., Gerosa, M. A., & Fuks, H. (2004). Combining communication and coordination toward articulation of collaborative activities. In G. J. Vreede, L. A. Guerrero, & G. M. Raventós (Eds.), *Proceedings of the 10th International Workshop on Groupware, CRIWG.* Lecture Notes on Computer Science Vol. 3198 (pp. 121-136). Berlin: Springer-Verlag.

Schmidt, K., & Simone, C. (1996). Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work, 5*(2-3), 155-200.

Stahl, G. (2001). WebGuide: Guiding collaborative learning on the Web with perspectives. *Journal of Interactive Media in Education*. Retrieved July 3, 2006, from http://www-jime.open.ac.uk/

KEY TERMS

3C Collaboration Model: A model for the analysis, representation, and development of groupware by means of the interplay between the 3Cs, namely, communication, coordination, and cooperation.

Awareness: The human beings' capability of perceiving the activities of the others and their own activities in the context of collaboration. A groupware generally provides elements and information to enable awareness.

Collaboration: The interplay between communication, coordination, and cooperation.

Communication: Conversation to negotiate and make decisions through an augmentation process.

Component-Based Development Techniques: Techniques that seek to develop modular systems composed of software components that may be adapted and combined as needed, always having reuse and maintenance in mind.

Component Framework: Defines overall invariants and protocols for plugging components.

Cooperation: Joint operation in the shared workspace.

Coordination: The management of people, their activities and resources, in the context of collaboration. In a narrower definition, it consists in managing interdependencies between tasks that are carried out to achieve a goal.