Extending MoLIC for Collaborative Systems Design

Luiz Gustavo de Souza(✉) and Simone Diniz Junqueira Barbosa

Department of Informatics, PUC-Rio, Rio de Janeiro, Brazil
{lgsouza,simone}@inf.puc-rio.br

Abstract. Much interaction design research has been devoted to collaborative systems, resulting in diverse design methodologies. Despite these efforts, we still lack a widely adopted interaction model for collaborative systems design. In this paper, we present a study on model-based design approaches, focusing on their limitations with respect to the 3C Model of Collaboration. Based on the 3C Model, we propose an extension to MoLIC, an interaction design language grounded in semiotic engineering but with no support for collaboration. We then illustrate the expressiveness of the extended MoLIC in the interaction design representation of a collaborative document editor.

Keywords: Interaction design · Semiotic engineering · MoLIC

1 Introduction

Collaborative systems allow users to interact with each other, collaborating, coordinating and communicating, as defined by the 3C Model of Collaboration [1], [2]. This kind of system brought about new problems and challenges for different fields, especially for interaction design. The field of Computer Supported Collaborative Work (CSCW) studies the development of collaborative systems, in which the interactions between users and between user and the system occur in a distributed and synchronized way [3, 4]. Some examples of collaborative systems include e-mail services, chat, social networks, and collaborative document editing [4–6]. Different works propose ways to ease and improve the design process for collaboration [5, 7–10], as well as present studies about design practice [4, 6, 11, 12]. In this paper, we identify problems and gaps that motivate the improvement of MoLIC, a design language grounded in Semiotic Engineering, to represent collaborative interaction.

Semiotic Engineering (SemEng) [13] considers the human-computer interaction to be a conversation between designer and user, and as such all computational artifacts are viewed as a kind of computer-mediated communication. According to SemEng, the user interface is the designer's deputy, conveying a one-shot message from designer to user, called a metacommunication message, or simply metamessage, allowing the user to interact with it to achieve his goals. This metamessage represents the solution proposed by the designer about his understanding of the users' problems, needs and preferences, and how they may interact with the system to achieve a range of anticipated (or unanticipated) goals. When it comes to collaborative systems, users also want

© Springer International Publishing Switzerland 2015
M. Kurosu (Ed.): Human-Computer Interaction, Part I, HCII 2015, LNCS 9169, pp. 271–282, 2015.
DOI: 10.1007/978-3-319-20901-2_25

or need to cooperate, coordinate, and communicate with other users. According to SemEng, the content of the metamessage for collaborative systems includes support for the user to see who is talking, what they are saying, whether they are listening, and how they respond.

Based on SemEng, different researchers proposed modelling languages to represent user and system interaction, such as MoLIC, a design language that allows designers to conceive and represent the interaction between the user and the designer's deputy. On multi-user interaction, there are two proposed modelling languages, MetaCom-G*, a tool that provides a design language to create interaction models, based on the collaborative work of the users and their objectives in the communication process; and Manas, another design language to create interaction models, based on the collaborative section of the metamessage and on social aspects of the interaction.

The MoLIC language has improved over the years in various researches, and can be used in a design process with different artifacts, such as interaction scenarios, task models, and user interface mockups. The main research on MoLIC points to improvement opportunities to be able to design a broader set of systems, which is the case of collaborative systems. The state of the art of MoLIC regarding collaboration provides elements for a simplified representation of the contact between users, not considering synchronization issues or any other collaboration concepts defined by collaboration models (such as the 3C Model).

Based on the SemEng and the 3C Model of Collaboration, in this paper we address the following research question: *How can we extend MoLIC in order to represent the design of collaborative systems based on the 3C Model, grounded in Semiotic Engineering*?

To answer this question, we have investigated related interaction models and the 3C Model, which allow us to list a set of concepts expected for an extension and propose new elements for the language. The next section presents briefly the SemEng and the 3C Model, as well as the related model: Manas, ConcurTaskTrees (CTT) and a variation of CTT in a methodology called CIAM (Collaborative Interactive Application Methodology) [14], and the MoLIC language. Following, we present the concepts that an interactive model must represent in order to include the 3C Model, showing in the studied interactive models gaps related to these concepts. We conclude by proposing the extension and showing a design example, followed by some concluding remarks.

2 Background

In this section, we briefly present Semiotic Engineering concepts, in which MoLIC is rooted, and the 3C model of collaboration, followed by the background works based on interaction models for collaboration.

2.1 Semiotic Engineering

As we presented in the introduction section, SemEng considers that the designer, through signs codified in the interface of interactive systems, communicates with the

user, including both in the role of interlocutors in the communicative process. The user interface reflects the designer's deputy, who represents the designer at interaction time. Through the user interface, the designer must inform the user about the meaning of the constructed artifact, and the user is expected to understand and respond to it by interacting with the designer's deputy [13].

The metacommunication from designer to user can be as the following template:

"Here is my understanding of who you are, what I've learned you want or need to do, in which preferred ways, and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision" [13, p. 84].

Considering not only the user-system interaction, but also the collaborative interaction among users, the metacommunication artifact template is extended, including the following:

"You can communicate and interact with other users through the system. During communication, the system will help you check:

1. Who is speaking? To whom?
2. What is the speaker saying? Using which code and medium? Are code and medium appropriate for the situation? Are there alternatives?
3. Is (are) the listener(s) receiving the message? What for?
4. How can the listener(s) respond to the speaker?
5. Is there recourse if the speaker realizes the listener(s) misunderstood the message? What is it?" [13, p. 210].

This way, the user can understand whether he can collaborate with other users, and how he can do it, based on the tools provided.

The MoLIC language, as we present in Sect. 2.6, describes the metamessage, supporting the designer to reflect on it. Our study presents the intention to represent the extended metamessage, allowing MoLIC to support collaborative interaction design.

2.2 3C Model of Collaboration

The 3C Model envisioned by Fuks [1] and based on the work of Ellis, Gibs and Reins [2] defined that the collaboration process can be described by three elements: communication, coordination, and cooperation. During collaboration it is necessary to provide ways for *communication* among users, synchronized or not, allowing for sharing of ideas, information and data. The *coordination* is achieved as a task organizer, improving the collaboration process as a whole while preventing conflicts and task rework. The *cooperation* is the process of joint work, sharing one task with members aiming at the overall goal.

These three elements work in a circular dynamic, where the communication is intended for preparing the coordination. While coordinating, the group acts and cooperates, creating demands again for communication in diverse situations like decision making and task renegotiation, restarting the cycle.

As a representation of the collaboration, we use the 3C Model as basis to understand what an interaction model should provide in order to represent collaborative

interaction. Along with the study of other design models, we present the concepts based on 3C to be represented in the MoLIC language on Sect. 3.

2.3 Design Scenario

The following sections present background works for interaction models, illustrating their main concepts through an example: the design of a collaborative text editor. The design intention is as follows:

> A user can create a document and edit it. At any moment, he can share this document with other users, so they can all edit it at the same time, keeping the synchronization and the awareness of each user's actions. It is also possible to chat with the group.

2.4 SemEng Collaborative Model: Manas

Barbosa [5] proposed Manas, a tool that considers social factors and focuses on the metacommunication artifact of SemEng, instead of considering the users' work, understanding that users might want to interact with no work-related, predetermined tasks.

Manas aims to support the designer's reflection on the users' problems, providing qualitative information about the communication. It provides the L-ComUSU language, which creates a m-ComUSU model, to be interpreted by a design logic interpreter.

The interpreter returns two types of information: the interpretation of the model exposing decisions both made and disregarded in the design process; and a set of the possible social impacts on the communication process and on the experience of the users.

Example. The design process of Manas involves defining the user roles and, for each possible user task, a speech is created, indicating who is talking, the purpose, topic and who the listeners are. When there is interaction among users, all the speeches involved are encapsulated as a conversation process. This way the designer can structure when there is user collaboration, and the role of each user in the collaboration.

For the design using Manas, we achieve the complete application design, specifying the sharing, editing and communication processes. After creating the model, the tool generates a report, indicating all the decisions made during design, also pointing to situations not considered, giving feedback about the possible problems and how they can be avoided.

Different from other models, Manas does not present a visual schema, and does not support a more tactical level of interaction, that is, Manas does not support the definition of how the interaction can occur, as it allows the definition of the users, what each one can do and how what they do will affect others.

Manas also does not indicate explicitly the cooperation objects, nor the expression elements to interact with them. The tool also does not support reactions based on another user action, that is, something that occurs to a user when another user performs

some other action. For example, when some user A removes a user B from a shared document while B is editing it, B not only needs to be notified, but also prevented from reading and editing the new document, by closing it automatically or disabling the interface. For Manas, in this situation, the user would only be notified as a listener of the "remove share" speech, but what happens beyond that is not considered.

2.5 Collaborative Task Model: ConcurTaskTrees

ConcurTaskTrees (CTT) was proposed by Paternò [15], and reflects the interaction design aiming to support the users and their tasks. CTT leads the designer to reflect on the users' tasks, decomposing them to the desired level of abstraction, modeling their representation in a temporal sequence.

The model takes the form of a tree, where the root represents the general task, and each level represents a more specific detail of the task. Each level of the tree uses temporal associations among tasks from left to right depending on their relationships. This means that each task can be executed at the same time, in a given order or based on some condition.

To support the design of cooperation, Paternò created the idea of multiple user roles, where one task to be performed by more than one user can be described in a cooperative CTT model, and then that task is shared among each user role, having an indication that the task belongs to the cooperative model.

Example. CTT gives support only for cooperation, with the concepts that the act of cooperation is to allow members to perform tasks sequentially and synchronized among each other. From the 3C Model's point of view, the cooperation aims to support the task performed in a shared environment, that is, there is task synchronization, but this synchronization is made on demand, not necessarily needing a logical sequence, allowing members to perform their tasks at any time, not just when another user reaches a desired state.

This way, it is possible to affirm that the support for cooperation given by CTT is limited, considering the full specification of the collaboration as defined by the 3C Model.

The figures above present a possible design solution for the collaborative text editor. The model in Fig. 1 shows the tasks to be performed by the user interacting with the system, including the emission of an asynchronous message. In the model presented in Fig. 2 it is possible to see that there is not a description of the shared environment, and also there is no description of the cooperation objects that the users are manipulating, which include the message log and the document itself. Besides, The document editing and the message exchange by other users do not create notifications, that is, there is no awareness about what the other users are doing and, as such, there is no coordination. The use of the CTT's cooperation was not the right solution for this case, giving a model that can be read as single or multi user. These issues and ambiguities show that the model:

- Does not represent the synchronous message exchange
- Does not define cooperation objects

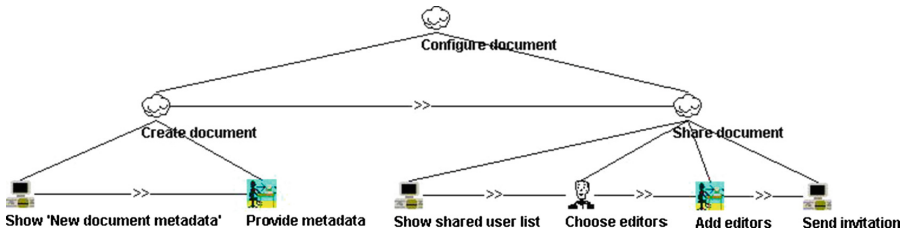

Fig. 1. CTT model to create and share the collaborative document

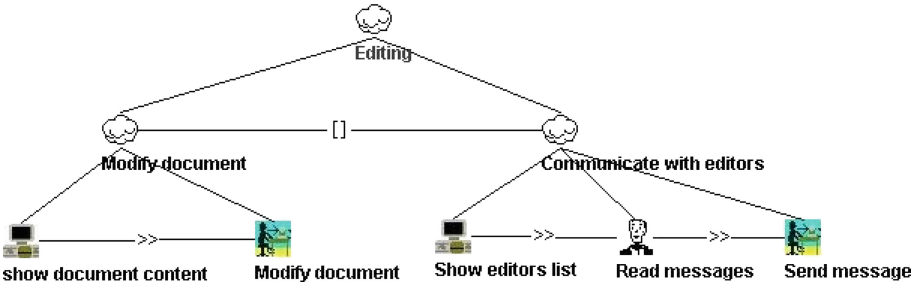

Fig. 2. Collaborative document editing

- Does not offer awareness mechanisms
- Does not allow the notification emission
- Does not explicitly shows which users are participating in the collaborative environment

CTT extension: CIAM. Using CTT as the base interaction model, and defining a complete method and notation to represent collaboration as a set of artifacts, Molina, Redondo and Ortega [14] proposed the Collaborative Interactive Application Method (CIAM). To use CTT, they argued that CTT does not have everything needed to represent collaboration, proposing new types of tasks pursuing the gaps found. This new notation allows to indicate collaborative/cooperative tasks and shared context tasks.

In this method, as the new notation indicates only collaborative tasks, the model is still not capable to solve the first, second and fourth limitations listed for CTT. The Fig. 1 would stay the same, and the Fig. 2 can be is enhanced resulting in Fig. 3.

2.6 SemEng Interaction Model: MoLIC

MoLIC (Modeling Language for Interaction as Conversation) is an interaction design language proposed by Barbosa and Paula [16] based on SemEng. As the theory perceives the interaction as a conversation between the designer's deputy and the user, MoLIC allows the representation of the interaction as a set of conversations that the user can have with the designer's deputy, expecting that the designer present clearly the metamessage. The language also serves as an epistemic tool, leading designers to improve their understanding about the problem to be solved and the artifact to be created.

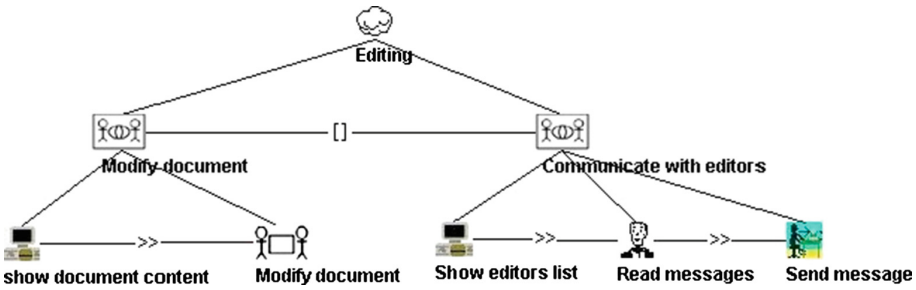

Fig. 3. CIAM extension CTT collaborative document edition

The design project starts with the definition of the user roles, their goals, analysis and interaction scenarios and the signs used in these scenarios. The model is then created to represent how the scenarios may occur, leading the users to achieve their goals during the interaction. In the process, the designer is led to reflect on the met-amessage, creating better ways to solve communication breakdowns and, exploring alternatives before creating the final user interface. Figure 4 presents the traditional MoLIC elements.

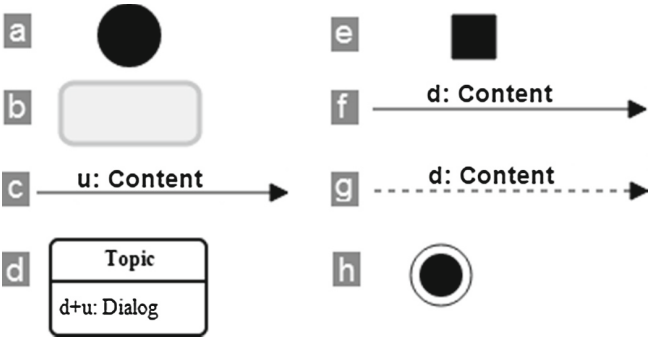

Fig. 4. Elements of the MoLIC language where: **a** is the opening point; **b** is a ubiquitous access; **c** is a user utterance; **d** is a conversation scene; **e** is a system processing; **f** is a designer utterance; **g** is a breakdown utterance; and **h** is a closing point.

3 Collaboration Concepts

Based on the 3C Model, we studied the meaning of each element and how interaction design models can incorporate its underlying concepts.

3.1 Communication

Communication involves the direct (sync or async) message exchange, that is, the ability to talk to another user or member of the collaborative process. Also, as the

members make their commitments during communication, it is interesting that the user be able to access the commitments later in the interaction by storing them. Thus, it is expected that the interaction design:

- Allow synchronous and asynchronous communication
- Provide awareness about the message transmission to and reception by other members
- Allow the awareness of the message transmitted to another member
- Define a communication language among members
- Provide the storage of the communication history so that members can access the commitments

3.2 Coordination

Coordination can be defined as the joint work, promoting the collaboration through the sum of the individual works. This process involves preparing the objectives and their division between the members, managing the collaboration, and continuously renegotiating the commitments and tasks, ending by analyzing the collaboration.

In order to support cooperation, the members organize the tasks temporally, including their interdependencies, identifying also the resources to be jointly manipulated, so that a computational coordinator can manage the tasks and resources.

There are cases where the computational coordination is absent, like when the collaboration occurs simply by chat or teleconference. In these cases, it is necessary that the members be able to use only the communication mechanisms to create the commitments, coordinate themselves based on the awareness elements and cooperate based on the expression elements on cooperation objects. Thus, it is expected that the interaction design:

- Identify the tasks and activities to be achieved by the group
- Identify the interdependency and temporal organization of the tasks
- Identify the objectives and resources to be manipulated and their respective types of temporal interdependency
- Allow the organization and coordination of the members based on communication

3.3 Cooperation

Cooperation is the joint effort in the shared environment, where the group perform the tasks, creating or modifying cooperation objects by using expression elements, so that awareness elements can track each user's work and inform the group, creating a shared environment. Thus, it is expected that the interaction design:

- Provide expression elements
- Provide cooperation objects
- Control the concurrent access to the cooperation objects through the expression elements
- Provide balanced information using the awareness elements

3.4 MoLIC Extension

The second version of MoLIC proposed an alternative to represent the contact between users using a contact point, where a user could reach another user's conversation topic to inform him. This led to some problems based on the user to designer communication process, seeing that a user speaks directly to another. This idea was to be the basis for collaboration, but further studies and with the use of the 3C Model of collaboration showed how collaboration must include other elements and dynamics.

To extend MoLIC allowing the representation of the collaboration based on the 3C Model, we propose the use of three new elements, the Incoming Message Indicator (IMI), Outgoing Message Indicator (OMI), and the Shared Space Indicator (SSI). Figure 5 presents the new elements.

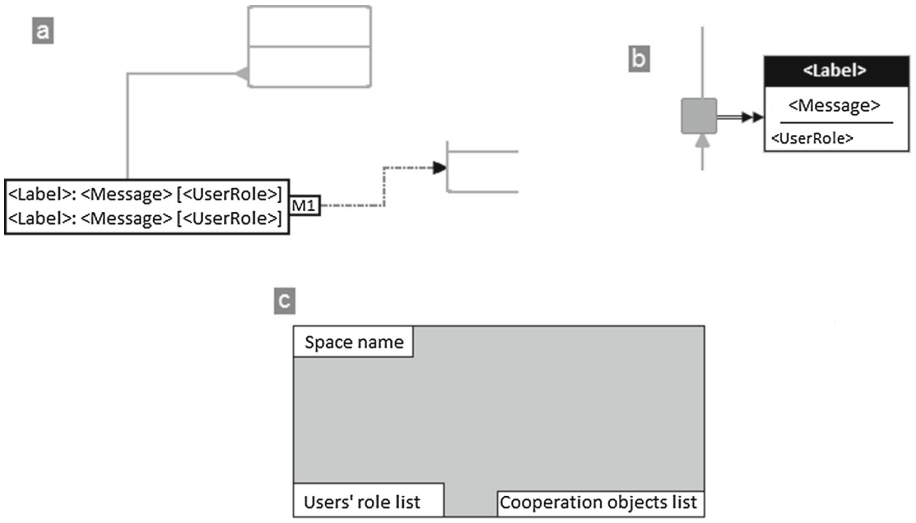

Fig. 5. New proposed elements, where: **a** is the incoming message indicator; **b** is the outgoing message indicator; and **c** is the shared space indicator.

The IMI indicates which message the user can receive during his conversation on the attached scene, and transitions from IMI indicate that when the user receives some message, he can or must change his current conversation topic to attend to the message. The OMI indicates that when the user make a request, the system will send some message to other users, to be received using the IMI.

The OMI is mandatorily attached to a system processing, indicating that the message to be sent is to be performed by the system, being either synchronous or asynchronous. The IMI is mandatorily attached to a scene, indicating that the user will receive a notification for some message only when an IMI attached to the scene has that message defined.

The SSI element represents one space area instance that all users share, and are able to seeing which topic the others are during their communication with the designer's deputy. The top left text describes the name of the space, the bottom left lists the user roles that can participate in that space, and the bottom right lists the cooperation objects that the users share inside the space. Modifications on a cooperation object are seen by all users inside the space, not specifying the type of synchronized access when modifying it. We understand that concurrency to access these objects might change based on different implementations, and the interaction model does not need to specify it at the interaction design phase. The expression elements to interact with the cooperation objects are defined by the SSI space itself and the conversations the user can have with the designer's deputy inside the space, allowing the extension to express the main components of each of the 3C's elements.

Based on the proposed extension, we designed the collaborative document editor system, presented in Fig. 6. As presented in the Fig. 6, the users can communicate with each other when documents are opened and closed, as one enters of exits the shared space to edit the document. While editing the document, users can see what the others are doing, which can be editing the document, sharing it, or sending a message to the user group inside the space. Users who are editing the document receive notifications for new messages sent by other users, as well as to users who entered of left the space.

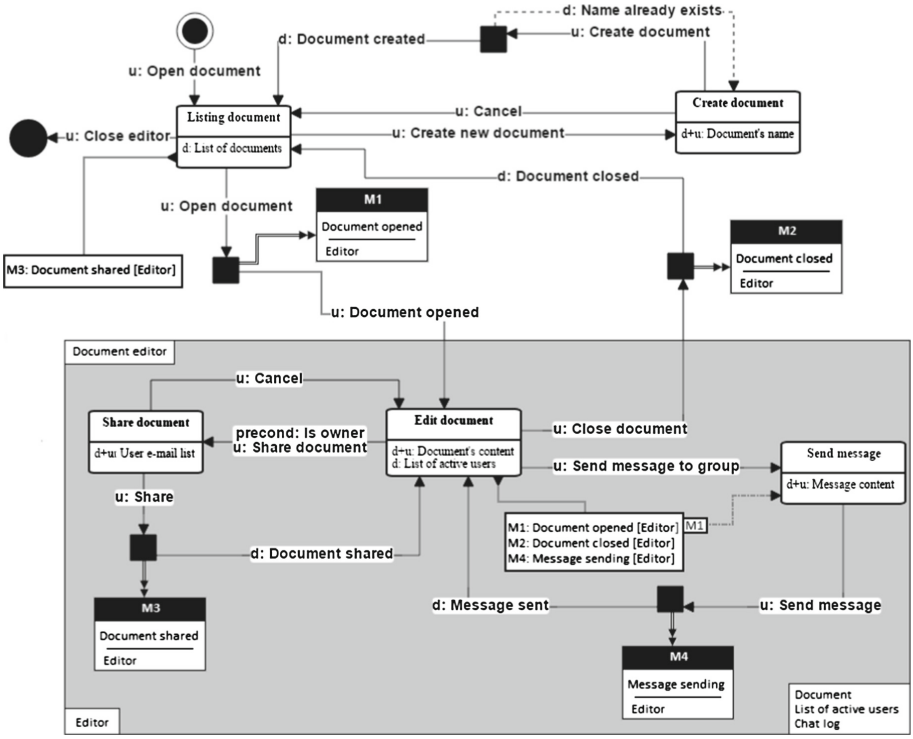

Fig. 6. Collaborative document editor design with MoLIC

The cooperation objects are referenced by name using plain text on the scenes, as well as other objects that can be defined using simply the original MoLIC language.

3.5 Concluding Remarks

Using the 3C Model of collaboration as the foundation theory for collaboration, we created a set of resources that an interaction model should provide to fully represent the collaboration. Based on these concepts, we analyzed related models, designing an example to verify their efficiency on representing such concepts. This study motivated our intention to extend the MoLIC language to represent the collaboration as defined by the 3C Model.

Based on the concepts and the related works, we defined new elements to represent the collaboration along with the complete MoLIC language, designing the proposed example to verify its expressiveness.

We see that using the elements can cause the diagram to be too large depending on the problem, by the number of IMI and OMI needed, as well as the size that the SSI might need to take. It shows that further studies on our solution are needed to evaluate this other kinds of problems, as well as to use it in a real design environment to evaluate its ease of use and its epistemic capacity when thinking of collaboration.

For future works we intend to test this extension with students in an academic setting and in professional settings.

References

1. Fuks, H., Raposo, A.B., Gerosa, M.A., Lucena, C.J.P.: Applying the 3C model to groupware engineering. In: Monografias em Ciência da Computação. PUC-Rio, Rio de Janeiro (2004)
2. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware - some issues and experiences. Commun. ACM **34**(1), 38–58 (1991)
3. Grudin, J.C.S.C.W.: History and focus. IEEE Comput. **27**, 19–26 (1994)
4. Rama, J., Bishop, J.: Survey and comparison of CSCW groupware applications. In: Proceedings of SAICSIT, pp. 1−12 (2006)
5. Barbosa, C.M.A.: Manas: Uma Ferramenta Epistêmica de Apoio ao Projeto da Comunicação em Sistemas Colaborativos. Doctoral thesis. Department of Informatics IIn (In Portuguese), PUC-Rio, Rio de Janeiro (2007)
6. Iacob, C. Using Design Patterns in Collaborative Interaction Design Processes. In: Proceedings of the ACM 2012 conference on Computer Supported Collaborative Work Companion, pp. 107–110. New York, (2012)
7. Alarcon, R., Guerrero, L.A., Ochoa, S.F., Pino, J.A.: Analysis and design of mobile collaborative applications using contextual elements. Comput. Inform. **25**, 469–496 (2006)
8. Barbosa, C.M.A.: MetaCom-G*: Especificação da Comunicação entre Membros de um Grupo. Master thesis. Depatment of Informatics (In Portuguese), PUC-Rio, Rio de Janeiro (2002)
9. Molina, A.I., Redondo, M.A., Ortega, M.: A methodological approach for user interaface development of collaborative applications: a case study. Sci. Comput. Program. **74**, 754–776 (2009)

10. Prates, R.O.A: Engenharia Semiótica de Linguagens de Interface Multi-usuário. Doctoral thesis. Department of Informatics (In Portuguese), PUC-Rio, Rio de Janeiro (1998)
11. Iacob, C.: Identifying, relating and evaluating design patterns for the design of software for synchronous collaboration. In: Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Systems, pp. 323−326, New York (2011)
12. Schadewitz, N.: Design patterns for cross-cultural collaboration. Int. J. Design **3**(3), 37–53 (2009)
13. De Souza, C.S.: The Semiotic Engineering of Human-Computer Interaction. The MIT Press, Cambridge (2005)
14. Molina, A.I., Redondo, M.A., Ortega, M.: A methodological approach for user interface development of collaborative applications: a case study. Sci. Comput. Program. **74**(9), 754–776 (2009)
15. Paternò, F.: ConcurTaskTrees: an engineering notation for task models. In: The Handbook of Task Analysis for Human-Computer Interaction, pp. 483−503 (2004)
16. Barbosa, S.D.J., de Paula, M.G.: Designing and evaluating interaction as conversation: a modeling language based on semiotic engineering. In: Jorge, J.A., Jardim Nunes, N., Falcão e Cunha, J. (eds.) DSV-IS 2003. LNCS, vol. 2844, pp. 16–33. Springer, Heidelberg (2003)