Evaluating the Expressiveness of MoLICC to Model the HCI of Collaborative Systems

Luiz Gustavo de Souza[1(✉)], Simone Diniz Junqueira Barbosa[2], and Hugo Fuks[2]

[1] Tecgraf Research Institute, PUC-Rio, Rio de Janeiro, Brazil
luiz@tecgraf.puc-rio.br
[2] Department of Informatics, PUC-Rio, Rio de Janeiro, Brazil
{simone,hugo}@inf.puc-rio.br

Abstract. In this paper we present an analysis on MoLICC, an interaction design language rooted in Semtiontic Engineering that perceives the user-system interaction as a conversation between designer (the system) and user, bringing focus to collaborative systems based on the 3C Model of Collaboration. In our analysis, we present the different aspects of collaboration as defined by the 3C Model of Collaboration, presenting case scenarios and using them to verify the language expressiveness.

Keywords: Interaction design · Semiotic engineering · MoLIC

1 Introduction

Collaborative systems go beyond user-system interaction as they must allow users to interact by cooperating, coordinating, and communicating with each other [1, 2]. These systems present issues and challenges in several research fields regarding the process of designing collaboration, especially considering the interaction design. The 3C Model of Collaboration was first proposed by [1], later improved by [2]. According to Fuks et al. [2], 3C stands for Communication, Coordination and Cooperation, which are the basic elements in collaborative systems, where they interact with one other in a cycle, allowing collaboration to occur. Later, Fuks et al. [3] presented an analysis of the various forms of relationship between each possible set of elements of the 3C model, highlighting the importance of the interplay between the different aspects of collaboration.

In previous works, we proposed MoLICC [4], an extension for the MoLIC language [5] based on the 3C Model of Collaboration. MoLIC [2] supports the designer in modeling the interaction, focusing on the users' goals and serving as an epistemic tool to help the designer understand the problem to be solved. Adopting an interaction-as-conversation metaphor, MoLIC allows designers to represent the interaction as a set of conversations that the user can have with the user interface to achieve his goals. MoLIC is rooted in Semiotic Engineering (SemEng) [6], a theory that considers the human-computer interaction to be a conversation between designer and user, and as such, represents all computational artifacts as a kind of computer-mediated

A. Marcus (Ed.): DUXU 2016, Part I, LNCS 9746, pp. 255–265, 2016.
DOI: 10.1007/978-3-319-40409-7_25

communication. According to SemEng, the user interface is the designer's deputy, conveying a one-shot message from the designer to the user. This message, called a metacommunication message, represents the solution proposed by the designer about her understanding of the users' problems, needs and preferences, thus representing how the user may interact with the system to achieve a range of anticipated (or unanticipated) goals.

In previous research, we studied the use of MoLICC in representative use cases [7, 8] and analyzed the language notation [9]. To address the challenge of representing the interaction of the different types of collaborative systems, we set out to answer the following research question: *How can MoLICC represent the collaborative interaction on different types of systems?*

In this paper, we present an analysis of the MoLICC language considering some prototypical collaborative systems: time scheduling, social networks, and crowdsourcing. We focus on the interaction aspects and on the MoLICC expressiveness, presenting possible interaction design solutions for each case and discussing how MoLICC supports the identification of design issues and the implications of certain HCI design decisions.

The next section presents the background of this research, the MoLICC language, its basis, the Semiotic Engineering and the 3C Model of Collaboration. Next, we present each case scenario considering the 3C Model with a representation using MoLICC, evaluating its efficacy. We conclude by discussing the language's potential and possible problems, pointing to future studies.

2 Background

In this section, we briefly present the SemEng and the 3C Model of Collaboration, followed by a description of MoLICC.

2.1 Semiotic Engineering

As mentioned, SemEng is based on communication, where the designer, using one or more sign systems codified in the user interface of interactive systems, communicates with the user. SemEng brings both designer and user together as interlocutors in the communicative process during user-system interaction. The user interface is the designer's deputy, representing the designer at interaction time. Through the user interface, the designer's deputy must inform the user about the meaning of the artifact (aka interface), expecting the user to understand and respond to it by interacting with him [6].

The metacommunication message from designer to user can be paraphrased as follows:

"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" [6, p. 84].

MoLIC was devised to encode the second part of the metacommunication message ("This is the system…"). As MoLICC aims to support modelling the collaboration aspects of interaction, the 3C Model of Collaboration was investigated to serve as a conceptual foundation for the MoLICC extensions to MoLIC. The next sub-section presents the 3C Model.

2.2 3C Model of Collaboration

Fuks et al. [2] proposed the 3C Model of Collaboration based on the work of Ellis, Gibs and Reins [1]. They defined that the collaboration process can be described by three elements: communication, coordination, and cooperation. The Model focused on groupware development, supporting the understanding of aspects involved in collaborative work.

Collaboration in the 3C Model occurs when the three elements work together as a cycle, where each element provides a bit of the collaboration and prepares the next element. One example of this application is the Conversation for Action, where members of a work group communicate, negotiating the work and making decisions, so they can coordinate their work in order to cooperate. This way, the communication element provides the communication process among members, allowing them to make commitments on the work to be performed. These commitments are then managed by a coordination process. In the end, the cooperation allows members to work together in a shared environment, where changes during the work take members to review their commitments and possibly renegotiate their work during communication again, thus restarting the cycle. Figure 1 shows how this configuration of the 3C Model works.

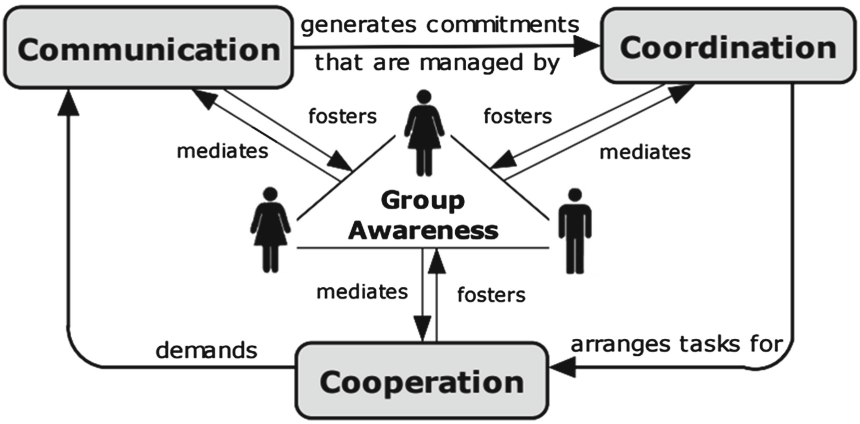

Fig. 1. The 3C model of collaboration for conversation for action

As Conversation for Action, there are other configurations that the 3C Model can achieve, grouping different kinds of collaborative systems, as previous studies have already shown [3]. Based on the 3C Model, the MoLIC language was extended to consider the collaboration aspects of interaction, resulting in a language called MoLICC, presented in the next sub-section.

2.3 MoLICC

MoLIC (Modeling Language for Interaction as Conversation) is an interaction design language based on SemEng, first proposed by Barbosa and Paula [5], later revised by Silva and Barbosa [10] and Souza and Barbosa [4, 9]. The language allows designers to represent the interaction as a set of possible conversations that the user can have with the user interface (considered by the theory to be the designer's deputy), expecting that it is clearly presenting the metacommunication message conceived by the designer. MoLIC serves as an epistemic tool, helping designers to improve their knowledge about the problem to be solved.

In a MoLIC diagram, the designer can define the different possible conversation topics, and the turn-taking utterances that users and designers can issue to advance the conversation towards a goal or to change topic to achieve another goal. Figure 2 presents the main elements of the MoLIC language.

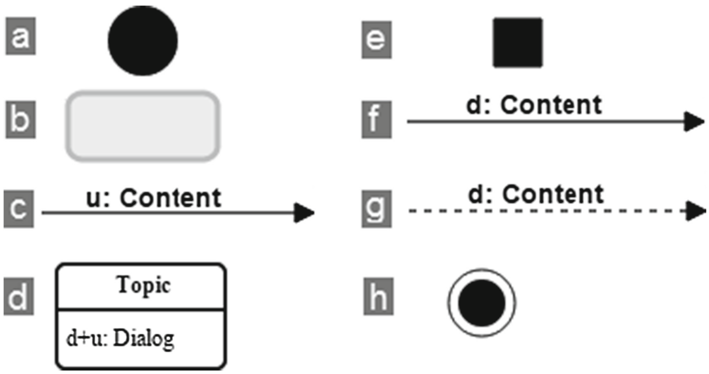

Fig. 2. Elements of the MoLIC language, where: **a** is the opening point; **b** is a ubiquitous access; **c** is a user utterance; **d** is a conversation scene; **e** is a system processing; **f** is a designer utterance; **g** is a breakdown utterance; and **h** is a closing point.

Considering the collaboration aspects, Souza and Barbosa [4] studied how the concepts provided by the 3C Model could be incorporated in the language to allow collaboration modeling. The proposal along with different studies led to a new version of the language, revisiting the current elements and proposing new elements, called MoLICC [7–9]. MoLICC elements allow designers to represent conversation among users and awareness aspects, supporting group work and cooperation. Figure 3 presents the new elements.

The next section presents the methodology we followed to study how collaboration can be represented with the language, analyzing its expressiveness based on the 3C Model of Collaboration.

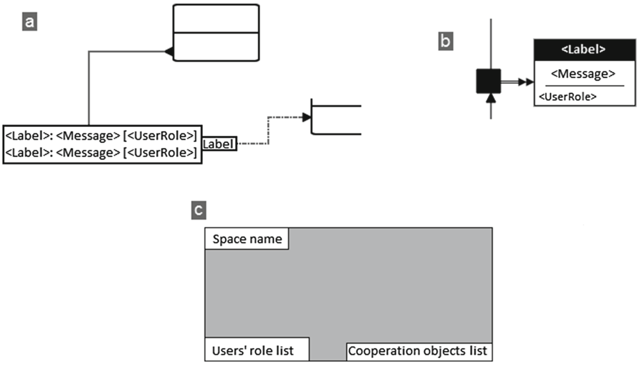

Fig. 3. Collaborative elements incorporated in MoLICC, where: **a** is the Incoming Message Indicator (IMI); **b** is the Outgoing Message Indicator (OMI); and **c** is the Shared Spaced Indicator (SSI).

3 Methodology

The MoLICC language evolved based on the 3C Model, and was tested with potential users [7, 8] and revised using a cognitive approach [9]. So far, we studied how well the interaction aspects are accomplished and how designers understand the new elements.

In this paper, we are interested in understanding how well the language can adapt considering the different kinds of collaboration, as the 3C Model shows. To verify the language usage, we considered three kinds of systems: Time Scheduling, Social Network, and Crowdsourcing.

For each kind of system, we described a practical case based on the 3C model, and proposed a solution using the MoLICC language.

In the next section, we present the results of the study, the proposed model and reflections on the collaboration aspects and how it is presented in the language.

4 Studied Cases and Analysis Results

In this section, we present each type of system studied and, based on the 3C Model, we present a discussion on the collaboration aspects and a possible solution using MoLICC.

4.1 Time Scheduling

Arranging meetings and events in general can be a challenge when having a reasonably large group. The scheduling system takes the 3C form Coordination → Communication → Cooperation, where coordination is the scheduling, communication allows users to negotiate and select dates, and during cooperation users select their available dates.

In Fig. 4 we present a possible solution, where a user can create schedules and share it with other users. When a new schedule is shared with the user (OMI M1: New Schedule invitation), he can then choose his preferred dates (scenes "View Appointment" and "Solve Scheduling"), and in the process see the other users' interaction with the schedule, and therefore their choices (within the SSI). The user who owns the schedule can finish the process by reviewing the selected dates and choosing the best option (scene "Solve Scheduling").

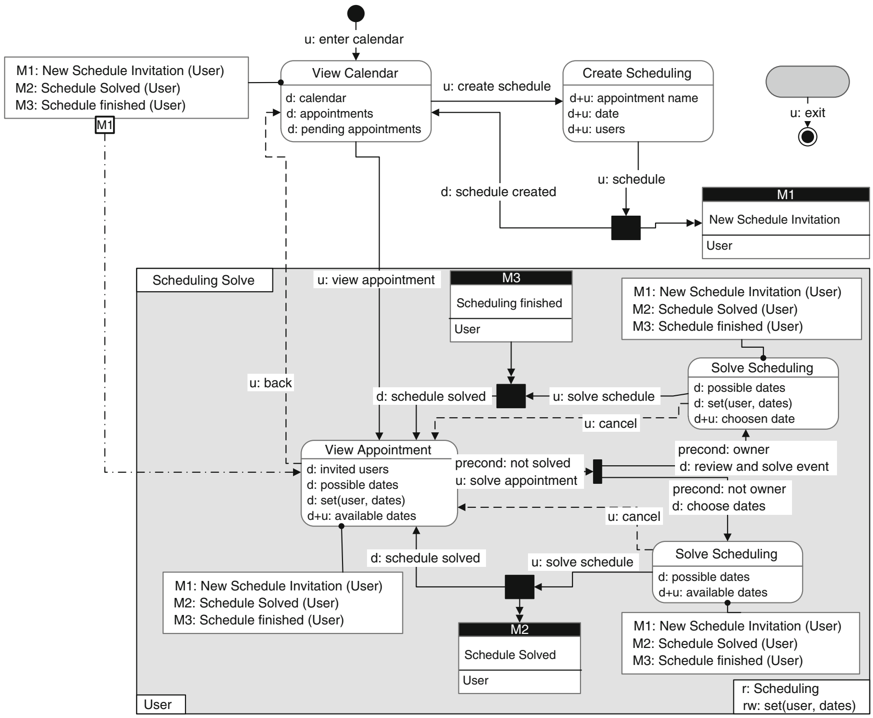

Fig. 4. MoLICC model for the time scheduling example

As the model shows, communication happens through the interaction of each user with the system, and not necessarily based on direct conversation such as chat or video conferencing. Decisions related to closing an event and choosing the best date is up to the users. This way, the interaction model focuses on providing the necessary tools to improve the process of scheduling an event collaboratively.

In conclusion, the SSI allow users to cooperate when choosing their best dates and finishing an schedule, as well as the OMI and IMI inform users about important decisions made for an specific schedule. On the other hand, as the cooperation relies on the users, it is not possible to represent the synchronization of the users (only between each user and the system), along with possible communication breakdown between them.

4.2 Social Network

Social Networks rely on more informal communication and interaction, where the users' main intention is to communicate with others. This way, this kind of system takes the configuration Cooperation → Communication → Coordination, where cooperation is the users' act of sharing their profiles with others, communication provides information exchange among users and coordination occurs when users add or invite others, so they can create and share their profiles.

In Fig. 5 we propose a model where the designers help the users to be aware of who is online and whether they can communicate, in this case using chat.

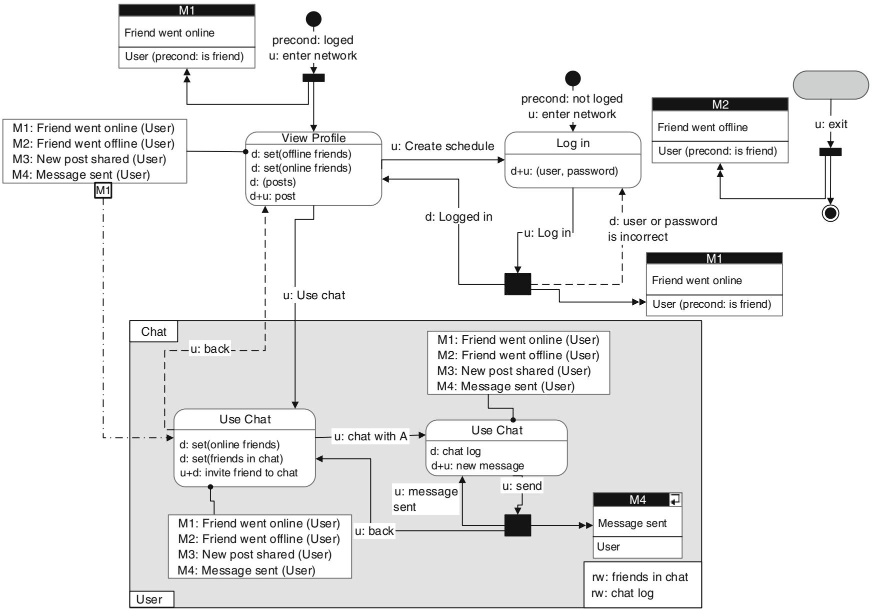

Fig. 5. MoLICC model for the social network example

As the model shows, the users are informed through the IMI and OMI elements when someone changes their status to online or offline (OMIs M1: Friend went online and M2: Friend went offline). There is no guarantee that the user will be aware during interaction and that the information is either enough or too much, but as we consider this as a concrete user interface (and not interaction) issue, it is out of the scope of MoLICC diagrams. In this example, we can consider MoLICC capable of representing what the system will present to the user, and that the user is able to receive and understand the message. On the other hand, the awareness that OMI and IMI provides with messages from other users, as well as the other users' status, is subjective and may distract the user. MoLICC does not provide an element to handle this kind of breakdown, different from communication breakdowns between user and designer's deputy [11].

4.3 Crowdsourcing

Crowdsourcing systems are more domain specific, focusing on cooperation, where users work together to create a product or knowledge based on this cooperation.

In this domain, the 3C Model takes the configuration Cooperation → Coordination → Communication, where cooperation provides ways to share an object with users, coordination guarantees that there is no conflict between different works, and communication contributes on planning and work division.

Let us consider the example of a translation-on-demand service, where users can find language specialists to request the translation of a document (scene "Request Translation" in Fig. 7). This way, translators share their profile and receive requests from users. During the translation work, the user who made the request can intervene in the translation process and be informed about its progress (SSI in Figs. 6 and 7). Figure 6 presents a model for the translator role, and Fig. 7 presents a model for the requester's role.

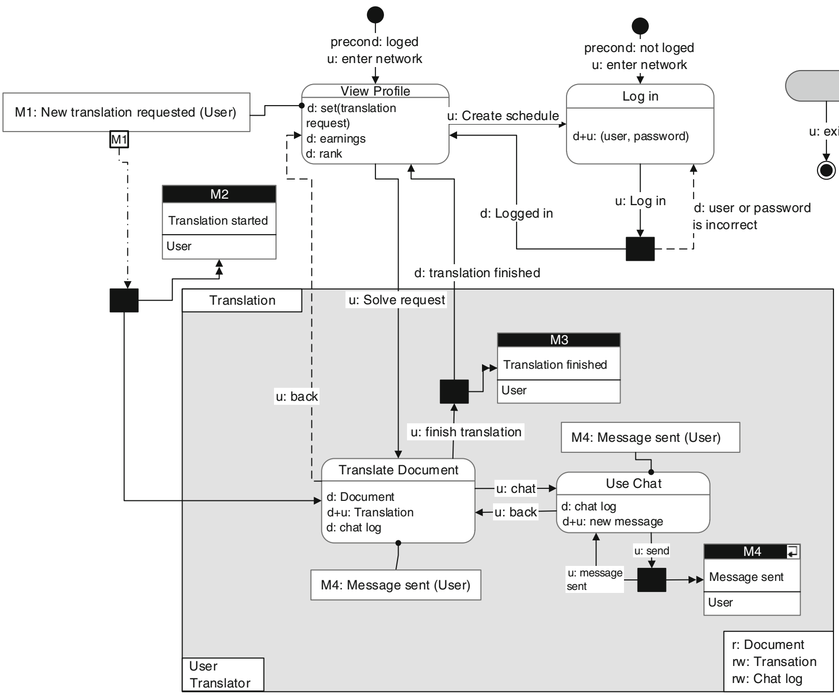

Fig. 6. MoLICC model for the crowdsourcing example of a translation request system, presenting the translator's role view

As Fig. 6 presents, the main feature of the system is related to cooperation, where the translator receives requests from users (OMI M1: New translation requested), and whose work can be observed during translation (OMI M2: Translation started, and both SSIs). In this scenario, the main object is shared (Object "Document" in SSI) (other examples include crowdfunding systems), but only one user has writing privileges during cooperation, avoiding any possible conflict.

In cases where one or more cooperation objects are shared with many users, such as coding, mapping or group translation, a design solution more oriented to concurrency is required, which we can somehow represent with MoLICC, due to the concurrent information propagation that the SSI provides.

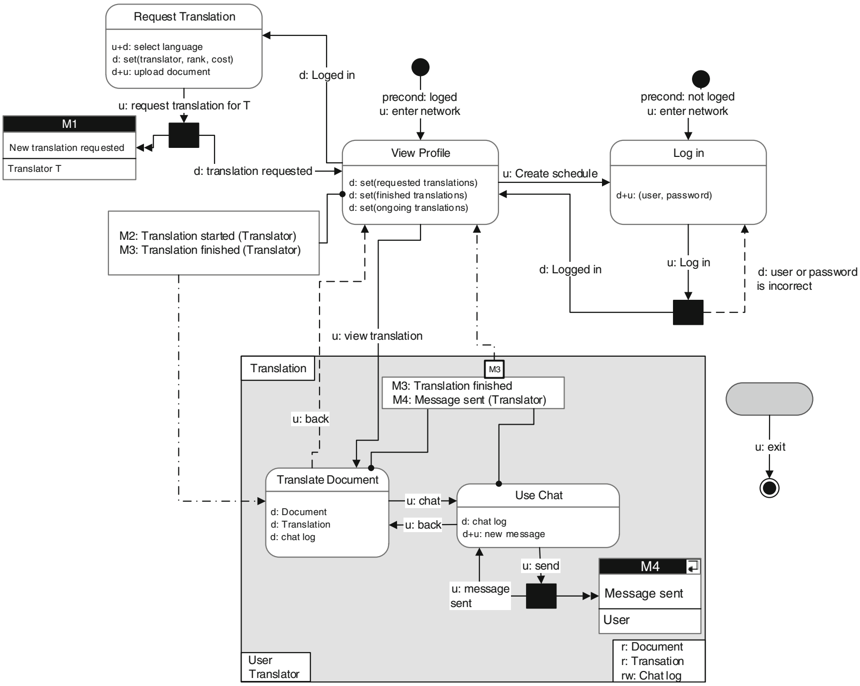

Fig. 7. MoLICC model for the crowdsourcing example of a translation request system, presenting the requester's view

MoLICC provides ways to represent concurrency with SSI, and information sharing with OMI and IMI, allowing an interaction model to focus on cooperation. Along with the previous cases, we can see that communication among users can be achieved by different alternatives, including interaction with OMI and IMI to share information a chatting. MoLICC can also represent video conferencing, providing the audio and video as shared object inside the SSI.

4.4 Conclusions

As we delved into the aspects of the 3C Model, collaborative use cases became more complex, pushing the MoLICC representation to its limits. As human factors are essential to collaboration, we are aware that MoLICCs' models, such as the ones we presented, cannot define or predict how the system will influence group work, or how the expected awareness can work. Moreover, cultural differences that cannot be modeled can also influence group work.

From the design point of view, the ability to conceive and model human interaction can shed some light on interaction problems earlier, as well as collaboration problems. Considering the intention of documentation and planning, MoLICC was shown to support modeling collaboration. The next section presents the concluding remarks of the research.

5 Concluding Remarks

Considering the different collaboration aspects provided by the 3C Model of Collaboration, we were able to depict use cases using MoLICC. In previous research, Souza and Barbosa [4] demonstrated that the language is capable of representing another 3C Model configuration, Cooperation → Coordination → Communication, designing a system for synchronized document editing.

Designing each use case led us to uncover two limitations of the language. MoLICC showed no explicit support to recover from possible breakdowns related to collaboration, as well as aspects related to the users' understanding and information overload. We still do not know if the current elements can allow this representation or the cost to adapt the language.

Regarding the second problem, MoLICC still lacks a tool to support inspection of the collaboration design. Such tool could support designers to understand possible problems in collaboration, such as different cooperation dynamics for the same model and possible information overload. Regarding interaction inspection, Lopes et al. [12] proposed a technique to support inspection of a MoLIC diagram based on gamification, not including the collaboration extension.

In future works, we will propose a way of representing collaboration breakdowns, studying how breakdowns occur. Also, we intend to investigate how to provide support to verify and locate collaboration-related problems in a MoLICC diagram.

References

1. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware - Some Issues and Experiences. Commun. ACM **34**(1), 38–58 (1991)
2. Fuks, H., Raposo, A.B., Gerosa, M.A., Lucena, C.J.P.: Applying the 3C model to groupware development. Int. J. Coop. Inf. Syst. **14**(2), 299–328 (2005)
3. Fuks, H., Raposo, A., Gerosa, M.A., Pimentel, M., Filippo, D., Lucena, C.J.P.: Inter- and intra-relationships between communication coordination and cooperation in the scope of the 3C collaboration model. In: Proceedings of the 2008 12th International Conference on Computer Supported Cooperative Work in Design, CSCWD, vol. 1, pp. 148–153 (2008)
4. Souza, L.G., Barbosa, S.D.J.: Extending MoLIC for collaborative systems design. In: Proceedings of the 17th HCI International Conference, Los Angeles, pp. 271–282 (2015)

5. 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)
6. de Souza, C.S.: The Semiotic Engineering of Human-Computer Interaction. The MIT Press, Cambridge (2005)
7. Souza, L.G., Barbosa, S.D.J.: Evaluating a MoLIC extension for collaborative systems design. In: Proceedings of the 2015 IEEE Symposium on Visual Languages and Human-Centric Computing, Atlanta, USA (2015)
8. Souza, L.G., Barbosa, S.D.J.: An Empirical study of MoLICC: a MoLIC extension for collaborative systems design. In: Proceedings of the XIV Brazilian Symposium on Human Factors in Computer Systems, Salvador, Brazil (2015)
9. Souza, L.G., Barbosa, S.D.J.: Avaliando a notação da MoLICC utilizando o framework cognitive dimensions of notations. In: proceedings of the XIV Brazilian Symposium on Human Factors in Computer Systems, Salvador, Brazil (2015)
10. Silva, B.S.: MoLIC 2a Edição: Revisão de uma Linguagem para Modelagem da Interação Humano-Computador. Dissertation (Master's in Informatics). PUC-Rio (2005)
11. Silva, B.S., Barbosa, S.D.J.: Designing human-computer interaction with MoLIC diagrams - a practical guide. In: Monografias em Ciência da Computação. PUC-Rio, Rio de Janeiro (2007)
12. Lopes, A., Marques, A.B., Conte, T., Barbosa, S.D.J.: MoLVERIC: an inspection technique for MoLIC diagrams. In: Proceedings of SEKE (2015)