Journal on Interactive Systems, 2023, 14:1, doi: 10.5753/jis.2023.2655

Evaluating Human-Computer Interaction at Design Time: an experience report with MoLVERIC Check in a MOOC environment

Beatriz Brito do Rêgo **[SPIDeLab/Universidade Federal da Bahia | beatribr@ufba.br]**
Filipe Adeodato Garrido **[SPIDeLab/Universidade Federal da Bahia | filipe.garrido@ufba.br]**
Jean C. S. Rosa **[ITI/LARSyS | jean.rosa@iti.larsys.pt]**
Ecivaldo Matos **[SPIDeLab/Universidade Federal da Bahia | ecivaldo@ufba.br]**

Abstract

The Modeling Language for Interaction as Conversation (MoLIC) is an epistemic tool inspired by Semiotic Engineering to model (human-computer) interaction through diagrams. MoLVERIC check (MCheck) is a diagram inspection technique developed for MoLIC, whose objective is to identify defects that would propagate in the modeled system's implementation and become communication interruptions. In this sense, this article is an experience report of evaluating a massive open online course (MOOC) environment through MoLIC, and predicting the quality of interaction to mitigate disruptions of metacommunication by MCheck. Although MCheck supports the inspection of MoLIC diagrams, in this article, we noticed the lack of MoLIC elements in the MCheck, indicating that this technique needs to evolve.

Keywords: *Interaction Design, Semiotic Engineering, MoLVERIC Check, Design Time.*

1 Introduction

(Human-computer) interaction (HCI) design can be considered a process for shaping the digital thing for human use (Lowgren 2013). Rosa and Matos (2016) also described interaction design as a process of constructing or manipulating dialog messages between the interaction designer and the user. The Semiotic Engineering theory, presented by (De Souza 2005), characterizes the HCI as a communicative process between the interaction designer and the user through an interface. In this sense, an interface is a designer's intellectual product, composed of messages encoded in signs.

To develop or model communication messages, an epistemological tool called Modeling Language for Interaction as Conversation (MoLIC) is used in Semiotic Engineering (Paula 2003; Silva 2005). This interaction modeling language shows the interaction from the designer's perspective through its agent (system), the dialogue with the user, and the interlocutors in a conversation. In this sense, MoLIC seeks to represent the users' interaction with systems following the conversation metaphor between the designer and the user through diagrams (Araujo 2008), in which design usually starts with defining the profiles, functions, and goals of users. Since its proposal, several software applications have been modeled with MoLIC for different domains and platforms (De Carvalho, et al. 2019). In addition, some extensions were proposed, which triggered a comprehensive review effort, resulting in the 2nd edition of MoLIC (Silva 2005; Da Silva, Barbosa 2007).

To inspect the diagrams produced through MoLIC, (Damian 2016; Lopes, *et al.* 2015a; Lopes, *et al.* 2015b; Lopes, *et al.* 2018) proposed two techniques, namely: MoLVERIC cards and MoLVERIC Check (MCheck). Both aim to identify defects that would spread to the HCI if the communication problems are not detected (Lopes, *et al.* 2015b). However, according to (Lopes, *et al.* 2015b) and (Damian 2016), MoLVERIC cards and MCheck are limited to inspecting the diagrams produced from the 1st version of MoLIC.

MCheck and MoLVERIC cards are diagram inspection techniques developed through MoLIC, whose objective is to identify defects that would propagate and become communication disruption errors after the diagrams are implemented. These disruptions can be avoided or mitigated by identifying them at "design time[1]".

In this sense, we present the experience of evaluating the interaction of a Massive Open Online Course (MOOC) environment using MCheck. This evaluation analyzes the quality of metacommunication, at design time, through MoLIC diagrams, predicting the interaction (user-system-user) to mitigate interruptions *to investigate, from a practical experience, the need for improvements in MCheck considering the evolution of MoLIC.*

The MOOC is an online educational environment with no predefined number of participants. In the study (Garrido, Do Rêgo, Matos 2018b; Garrido, Do Rêgo, Matos 2021), the development of interaction modeling of a MOOC platform was presented, using the MoLIC language, to favor the designer-user and user-user dialogue during the interaction. Considering that the interaction between subjects is essential for staying in the course and learning,

[1] A portion of time/project dedicated to planning, design, prototyping, testing, and implementation. It precedes the interaction in the software by the user (interaction time) when the system is made available for use.

focusing on aspects of human-computer interaction becomes fundamental (Do Rêgo, Garrido, Matos 2018).

The methodology adopted in the present study is divided into three stages: (i) evaluating the diagrams of a MOOC generated through MoLIC (Garrido, Do Rêgo, Matos 2021; Garrido, Do Rêgo, Matos 2018b; Garrido 2018); (ii) reporting the defects; and finally, (iii) indicate which MoLIC components are not supported by MCheck.

From our evaluations, we identified elements of the 2nd version of MoLIC that were not covered by the version of MCheck used. Consequently, an update was required. Therefore, the main contribution of this study is identifying which elements of the second version of MoLIC are not covered by MCheck so that it could be updated accordingly. Another contribution is the evaluation of the MoLIC diagrams and suggesting further improvements. A suggestion for future work is to add new checklists to evaluate the new elements inserted so that the MCheck technique can contribute to the analysis of MOLIC.

The remainder of this paper is organized as follows. Section 2 presents the theoretical foundation of the theme of the present study, which includes a discussion of MoLIC and Mcheck. In Section 3, the methodology used in the present study is presented. Section 4 presents the evaluation results. In Section 5, we present the discussion of the results. In Section 6, our conclusions, directions for future work and threats to the validity of the study are presented.

2 Background

This section is intended to present the theoretical framework that serves as a basis for understanding the topics covered in the study.

2.1 Semiotic Engineering

The term “semiotics” comes from the Greek word semeion, which means signs or sign; it can also be traced back to the Latin word signum, which means “mark” or “notch”. According to (De Souza 2005), semiotics is a discipline dedicated to the study of signs and how they are used in communication, a crucial process in intellectual activity.

Semiotic Engineering defines HCI as a human-human communication (*i.e.* user-designer interaction) through meta-communication via an interface (De Souza 2005). Meta-communication can be understood as the communication process between the designer and user through the system interface designed by the designer[2].

This meta-communication process involves sending the message designed by the designer to be then received by the user (Prates, Barbosa 2007). This message communicates the designer's vision about who the message is intended for, what problems it can solve, and how the user can understand it.

According to (Rosa, Matos 2016), HCI design is a process of conception/manipulation of interaction messages arranged in the interface. To model the communication process between designer and user, these messages were modeled by MoLIC (Paula 2003).

2.2 MoLIC

To model user-designer communication during the interaction design process from the Semiotic Engineering perspective, (Paula 2003) suggests the use of MoLIC. MoLIC represents the users' interaction with systems following the metaphor of conversation between the user and the designer through diagrams (Araujo 2008).

MoLIC was first proposed in a 2013 master's dissertation entitled Human-computer interaction project based on models based on Semiotic Engineering: construction of an interaction model. This dissertation presented the MoLIC, inspired by the unified modeling language (UML), which would serve as an epistemic tool to support the designer's reflection on the conceived interactive solution to fill a gap detected between the task models and the interface models (Paula 2003; Silva 2005).

Paula (2003) highlighted through a case study that their notation is easy to learn. In addition, the proposed models have good expressiveness, mainly because they represent possible disruptions in communication, which can be avoided or mitigated once identified.

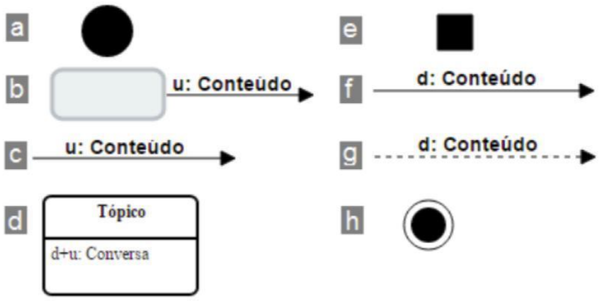

Figure 1. MoLIC 2nd version Elements (De Souza, Barbosa 2015).

The diagrammatic elements of the first version are: (a) opening point, (b) ubiquitous access, (c) user utterance, (d) scene, (e) closing point, (f) system process, (g) designer utterance, and (h) breakdown recovery utterance (Figure 1) (Paula 2003).

- **Opening point:** indicates the beginning of user interaction with the system.
- **Ubiquitous access:** represents the user's opportunity to change the conversation topic to reach a goal different from the current one.
- **User utterance:** represents an explicit change in the conversation's turn or topic, issued by the user (u:).
- **Scene:** represents the dialogues (conversations) between the user (u:) and the designers'' deputy (d:).
- **System process:** represents the processing that the system must perform after the communicative exchange with the user, and then respond to the user the results of the processing.

[2] In Semiotic Engineering, the designer is the interlocutor of the dialogue.

- **Designer utterance:** represents an explicit change in the turn or topic of the conversation, issued by the preposter (designers' deputy) (d:).
- **Breakdown recovery utterance:** indicates a type of speech from the preposter (d:) for a conversation break recovery situation.
- **Closing point:** indicates the end of user interaction with the system.

Moreover, (Silva 2005) added new elements to MoLIC; the resulting model was then presented as the 2nd version of MoLIC. According to (Silva 2005), the 2nd edition of MoLIC does not differ radically from the first version, and nor does it add a large number of new elements to interaction diagrams. Instead, the MoLIC update seeks to improve the semantics of the interaction diagram elements, providing some new features for the detailing of the interaction (Silva 2005).

Since then, many other elements have been added to MoLIC, mainly to serve multiuser systems (Silva 2005), collaborative systems (Araujo 2008), and design of MOOCs (Garrido 2018; Garrido, Do Rêgo, Matos 2021). These elements are presented in Section 5.

The diagrams developed through MoLIC represent communication possibilities during interaction and possible communication disruptions. These failures can be avoided or mitigated if identified at "design time". However, communication diagrams produced through MoLIC can be faulty. In light of this, MCheck was developed to inspect the communication diagrams.

2.2.1 MoLVERIC Check

To inspect the diagrams produced through MoLIC, Damian (2016) proposed two techniques, MoLVERIC Cards and MoLVERIC Check. Both are diagram inspection techniques developed through MoLIC, whose objective is to identify defects that would be propagated and become communicability disruptions after implementation of (Lopes, *et al.* 2018) diagrams.

The purpose of MoLVERIC Cards is to assist in the identification of diagram defects in a simple way, preventing eventual defects from being propagated to artifacts developed based on these diagrams (Lopes, *et al.* 2015). MCards employs gamification elements in order to motivate evaluators during the inspection of diagrams (Damian 2016).

The MCheck technique has the same purpose as MCards, with the difference that it does not use gamification elements and is a *checklist-based*[3] technique (Damian 2016). Damian (2016) characterizes the possible defects that can be found in the diagrams generated by MoLIC, based on a taxonomy presented by Travassos *et al.* (1999) , with a total of five (05) probable defects, described below.

- **Omission:** Omission or negligence of any necessary information.
- **Ambiguity:** It occurs when a given information is not well defined in the interaction diagram, thus allowing multiple interpretations.
- **Incorrect fact:** Incorrectly using the elements of the interaction small diagram for the interpretation of those involved.
- **Extraneous information:** Unnecessary information included in the interaction diagram.
- **Inconsistency:** It occurs when there is contradictory information between the elements of the interaction diagram and the information needed to resolve the problem.

Elements that are inspected in this study are related to the first version of MoLIC: (a) opening point, (b) ubiquitous access, (c) user utterance, (d) scene, (e) system process, (f) designer utterance, (g) breakdown recovery utterance, and (h) closing point (Damian 2016).

The MCheck technique does not stipulate the required number of evaluators (Damian 2016), and thus MCheck is used by academics and industry professionals who have prior knowledge of MoLIC, interaction design, and Semiotic Engineering.

2.3 Related Works

The search carried out for studies dealing with the definition and execution of the MCheck technique returned few studies on the subject.

During interaction design modeling, models are developed to help design a suitable user interaction with the system; one way to design the interaction design is to use the MoLIC language (Lopes, *et al.* 2015). However, inspections for these models are necessary to verify that the MoLIC diagrams are complete, consistent, unambiguous, and contain few or no defects, to avoid propagating avoidable defects to derived artifacts.

The study by (Lopes, *et al.* 2015) aims to present a feasibility study of Molveric Cards (MCards). In order to evaluate the MCards technique before its transfer to the industry, a feasibility study was carried out in an academic environment to analyze the effectiveness, efficiency and perception of the participants. MCards was evaluated against a type-of-defect (BTD) inspection approach, which allows the identification of defects in MoLIC diagrams, as inspection techniques with the same verification focus as MCards were not found.

The main contribution of the study by Lopes, *et al.* (2015) was proof of the feasibility of the MCards technique and the finding by participants of the feasibility study that MCards was useful and easy to use.

The use of interaction models in the design stage is important due to user perspectives, as problems in user-system interaction can be mitigated. In this context, MoLIC provides the development of interaction solutions; however, in a preliminary study to analyze the use of MoLIC diagrams, different types of defects were identified, such as omission, incorrect fact, inconsistency, ambiguity, and strange information. These results indicate a need to inspect the MoLIC diagrams, so that the propagation of

[3] Checklist

these defects to other artifacts can be avoided. Furthermore, the sooner a defect is detected, the lower the cost to repair it (Damian 2016).

The study by Damian (2016) presents specific techniques for inspecting MoLIC diagrams, called MoLVERIC Cards and MoLVERIC Check. The verification items of the two techniques assess both the consistency of the MoLIC diagrams with the interaction scenario/system requirements, and the notation used in the MoLIC diagrams. The techniques had their construction and evaluation supported by experimentation. The results of this study provide evidence of feasibility for inspecting MoLIC diagrams of both techniques.

Inspection techniques are essential in the software development process, saving staff resources in finding errors before subsequent production phases. (Santos, *et al.* 2018) used the MCheck technique to inspect the elements of the diagrams generated by MoLIC. The application of the technique proved to be efficient in identifying and correcting the interaction diagrams. The application of the technique proved to be efficient in the identification and correction of the interaction diagrams of an academic process management system (Santos, *et al.* 2018).

3 Methodology

The study methodology is divided into three stages: (i) evaluating the diagrams of a MOOC generated by the MoLIC language (Garrido, Do Rêgo, Matos 2018; Garrido, Do Rêgo, Matos 2021); (ii) reporting the breakdowns found; and finally, (iii) indicate which MoLIC components are not supported by MCheck.

In this study, interaction design diagrams of a MOOC platform will be evaluated, which will have a course with computer education practices. The MOOC platform is called Saviesa and is available at www.saviesa.com.br[4]. The course was developed under the Creative Commons CC BY-NC-SA 4.0 license and was the basis of the study by (Garrido, Do Rêgo, Matos 2018a; Garrido, Do Rêgo, Matos 2021). The course diagrams were designed using a hybrid interaction design and instructional design approach using MoLIC (Paula 2003; Silva 2005; Araujo 2008; Garrido 2018).

Three researchers, one of them also a professional in the industry, used the MCheck technique (Damian 2016) under the supervision of a specialist in the field to assess the quality of the diagrams developed in (Garrido 2018) study at “design time”.'

Beatriz Brito, Master in Computer Science, conducts research in HCI, especially communicability and interaction design, like Jean Rosa. Filipe Garrido has a similar profile, additionally researching instructional design. Finally, Professor Ecivaldo Matos is the advisor of both and has extensive experience in research in the area of HCI. He followed the inspection process as well as performed the final review of the results.

As MCards is used with groups with more than one participant, MCheck was selected for the number of inspectors. There are four research participants, but only one was responsible for inspecting the MoLIC diagrams using the MCheck technique.

Of the four participants, one was responsible for inspecting the diagrams and reporting the failures found, after which a meeting was held with the other researchers in which one was responsible for creating the MoLIC diagrams, to discuss the failures found and adjustments in the interaction. After discussing the results, the defects were accounted for and reported.

Noting that all communication failures were reported, and no false positives were identified. That is, at design time, all possible breakdowns that were not replicated in the system release version were mitigated.

In the first phase of inspection of the MoLIC diagrams, standard verification items were used for each MoLIC element. Table 1 presents the model used by the technique to evaluate each element of the MoLIC language, similar to the checklist that guided the evaluators. The first line of the table defines which MoLIC element will be evaluated (see Figure 1).

Table 1 presents a brief representation of how each MCheck checklist is characterized. Next to the element description is a visual representation of the element being evaluated. Below this set of information, the evaluation checklist can be found, wherein each item has an identifier referring to the name of the element being evaluated.

Table 2 contains the list of elements and their description. There are eight tables, divided into a scene, open point, close point, system process, ubiquitous access, transition speech, and speech range retrieval (unified into a single table). The scene element is the only one that has subdivisions, such as scene topics, dialogues, and signs (Araujo 2008).

Table 1. Template - Cards of checked MCheck Items (Damian 2016).

<table>
<tr><td colspan="2"><i>Elements</i></td></tr>
<tr><td><b>Description</b></td><td><b>Example</b></td></tr>
<tr><td>Description of how to recognize the MoLIC element</td><td>Graphic example of the MoLIC element</td></tr>
<tr><td colspan="2"><b>Check Items:</b>
<b>E1:</b> Guiding question to find defects
<b>E2:</b> Guiding question to find defects</td></tr>
</table>

MCheck provided check items so that evaluators could inspect the diagram according to each Table 2 (Lopes, *et al.* 2015b; Damian 2016). Each checked item has an identifier for the item inspected. The identifiers of each item are composed of the element initials and numbering. The following are the identifiers for each element: scene topic (ST), scene - dialogues (D), scene - signs (S), transition utterance and breakdown recovery utterance (TUB), opening point (OP), closing point (CP), ubiquitous access (UA), and system process (SP) (Damian 2016). The

[4] (disabled)

templates of each element that will be evaluated can be checked at link.

Table 2. Identification and definition of verification items.

ID	Definition
ST	The topics in the scenes represent the user's goals at the time of the interaction in which the user decides how the conversation should proceed.
D	The dialogs represent the actions of the users regarding the topic of the scene.
S	Signs represent the information involved during the dialogue.
TUB	Transition utterance represents the topic changes from the current scene, either the user or the designer's agent, the rupture recovery utterance represents a type of utterance for the breakup recovery situation.
OP	The opening point indicates the beginning of user interaction with the system.
CP	The closing point indicate the end of user interaction with the system.
UA	Ubiquitous access represents the opportunity for the user to change the topic of the conversation, to reach a goal different from the current one.
SP	The system process represents the internal processing of a user's speech, so that it is possible for the system to interpret the user's utterance to provide appropriate targeting.

4 Results

Our results highlight the elements of MoLIC that MCheck does not cover. To better understand the evaluation of the diagrams, the objective that each diagram sought to achieve is presented, followed by a list of defects identified by Mcheck. The diagrams were created by (Garrido 2018) and were presented in their entirety in their study; as diagram images will not be available in this section, only cuts with highlights of the defects found. But the diagrams presented are the ones with the defects, without modifications after identifying the ruptures found.

The first diagram evaluated was the user registration. This diagram refers to the registration of users on the MOOC platform, as they will only have access to the course content if the user(s) register. In this diagram, three (03) defects were found that can become communicability disruptions after the development of the system.

We highlight in Figure 2 the content of the "User registration" scene. The recovery speech is issued if the user enters an email that is not valid for his registration confirmation. The break **TUB4 - External Information** was identified in a catchphrase that could be summarized only for invalid email, a phrase commonly used in other online systems. This fact would not be considered a serious error, just a suggestion for improvement.

The first defect found was **S6 - incorrect Fact**, as the signs were not used correctly. If the reverse order is correct, d + u :, which references the designer asking the user to respond to their request using "speech" with the information necessary for the student's enrollment.

The second defect identified in the user (student) registration interaction diagram is the defect of the type **UA2 - incorrect fact**. The ubiquitous access element is not related to another element; one of the main rules for using this element is the exclusive permission for the destination of a speech to be ubiquitous access when it is part of the initiated conversation. This was not represented in the diagram of the users' register.

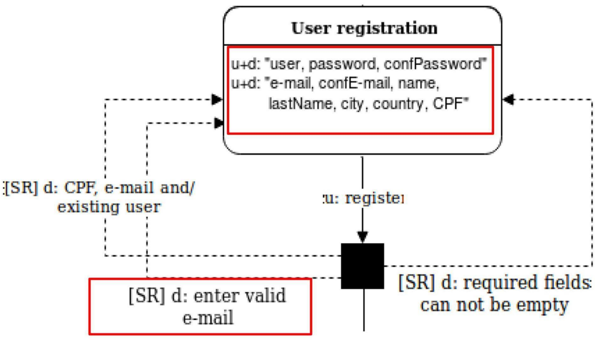
User registration
u+d: "user, password, confPassword"
u+d: "e-mail, confE-mail, name, lastName, city, country, CPF"
:[SR] d: CPF, e-mail and/ existing user
:u: registe
[SR] d: enter valid e-mail
[SR] d: required fields can not be empty

Figure 2. Diagram register user - defect strange information.

Other elements of MoLIC are not identified in the valuation technique as a whole, as noted in Figure 3. These elements are added to the MoLIC language as it evolves, such as the bifurcation and the point of contact with the external system. Thus, the importance of the evolution of MCheck becomes evident so that MCheck can evaluate the diagrams that use the latest version of MoLIC.

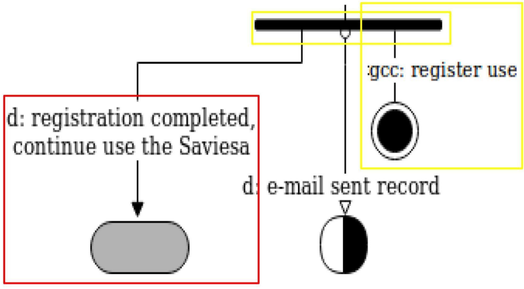

Figure 3. Diagram register user - defect factor incorrect.

The element named fork is represented by a rectangle with round edges and filled. This element was used to combine the change of topic of the conversation and the influence of the interaction of an external actor or the ending of a conversation on a final objective (Araujo 2008).

The point of contact with the external system is represented by a circle with one half white and the other half black. This element is used when user interaction forwards it to an external system, as the name of the element define. In the context of this diagram, contact with the external system occurs when the user completes the registration, as the system will request confirmation by email so that the student has access to the platform.

Finally, we have the element *end goal conversation closure*, where a black circle represents it circumscribed in a larger circle. This element represents the end of the conversation, whether successful or not. This element is accompanied by the expression *gcc: [goal-related conversation closing]*, without any arrow.

In Figure 4, part of the interaction diagram of the course evaluation is represented. The first defect found is highlighted at the top of Figure 4, identifying the defect of type **ST4 - incorrect fact**. The scene has no title; a title suggestion is *review presentation* because the purpose of the scene is to explain how the user will proceed in his evaluation.

Another defect found in this scene is the type **S6 - incorrect fact**. For the signs *u* + *d:* are not represented in the correct order, which is the reverse order *d* + *u:*.

In the system process, after the left side bifurcation (highlighted by an arrow), there is no transition speech linking the bifurcation to the scene named *repeat review.* This leads to two defects, **SP2 - incorrect fact**, which, after processing the system, does not use the designer's agent's utterance for the transition speech and recovery of the conversation break and the second defect, **TUB3 - Default**, the lack of transitional speech content (see Figure 4).

If we consider the callsign for the bifurcation element, a transitional speech is allowed to remain unlabeled if the transitional speech that leaves the bifurcation belongs to the user (Araujo 2008).

In the evaluation diagram (*cf.* Figure 4) the defect **SP5 - incorrect fact** is identified. After system processing, there is no break recovery speech in the evaluation diagram. Another defect reported is **TUB6 - ambiguity**, characterized in the speech *gcc: perform activity*. The expression "perform" denotes that the student has not performed the activity at all, but this transition speech refers to completing the activity's attempts. The designer can be more explanatory by modifying the expression to "send activity" or equivalent synonym.

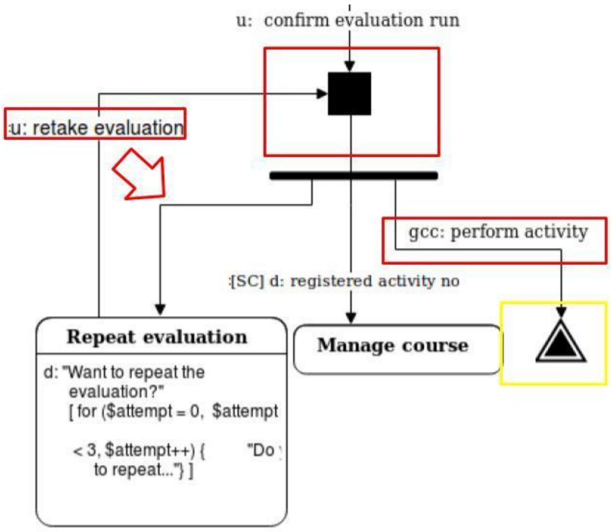
u: confirm evaluation run
:u: retake evaluation
gcc: perform activity
:[SC] d: registered activity no
Repeat evaluation
d: "Want to repeat the evaluation?" [for ($attempt = 0, $attempt < 3, $attempt++) { "Do to repeat..."}]
Manage course

Figure 4. Evaluation Diagram.

The defect **TUB5 - inconsistency**, highlighted in the transition statement *u: retake evaluation*, was identified because the speech was not consistent. The content of this speech does not present the requirements and information of the corresponding scenario. This transitional speech should have another meaning, such as "`do the evaluation again" or "retry". Another highlight is **TUB1 - incorrect fact**, wherein the speech direction should be linked to the first scene (without objective).

Finally, the defect of the type **ST1 - omission** is highlighted because there is no evaluation scene in which the designer explains how the user should carry out the evaluation modeled by the instructional designer (Garrido 2018). In this scene, it is necessary to specify what type of assessment the student will perform (questionnaire, comments in forums). Possibly the evaluation scene would be epistemic because it is the main connection of the diagram, situated between the first scene and the system process, interconnecting the others.

In this diagram, the presence of a new element was also highlighted (in yellow) (see Figure 4). This triangle-shaped element was conceived by (Garrido 2018) for the modeling of courses applied in a MOOC environment. This new element represents the action of the student to complete an assessment, different from the purpose of closing the closing point, because, when completing an assessment, a grade is stored (**[SR - score capture]**) and will influence the student's course (dialogue) throughout the course.

Figure 5 presents the forum interaction diagram for each module of the course. For this MOOC course, it was defined that there would be two types of forums: General, with varied subjects about the topic of the course, opinions, criticisms, and suggestions from students; and **Module**, a forum that will serve as a complementary assessment for each course module (Thaler, Fialho 2015; Garrido 2018).

The forum of each module aims to restrict the topics that will be discussed for subjects related to each topic covered in each module. Thus, the evaluation in pairs is possible (Costa, et al. 2015), an approach widely used in MOOC courses. On the other hand, the general forum aims to encourage students to interact spontaneously to discuss the topics introduced in the course or indicate subjects and content of interest to their peers.

Three highlights can be seen in the diagram referring to the forum of the course module (Figure 5). Two defects were identified: **SP2 - incorrect fact** and **SP5 - incorrect fact**. In the two highlights of the system process image, the identified defect is about the rupture recovery speeches, in which both outputs contain only one speech of the transition type.

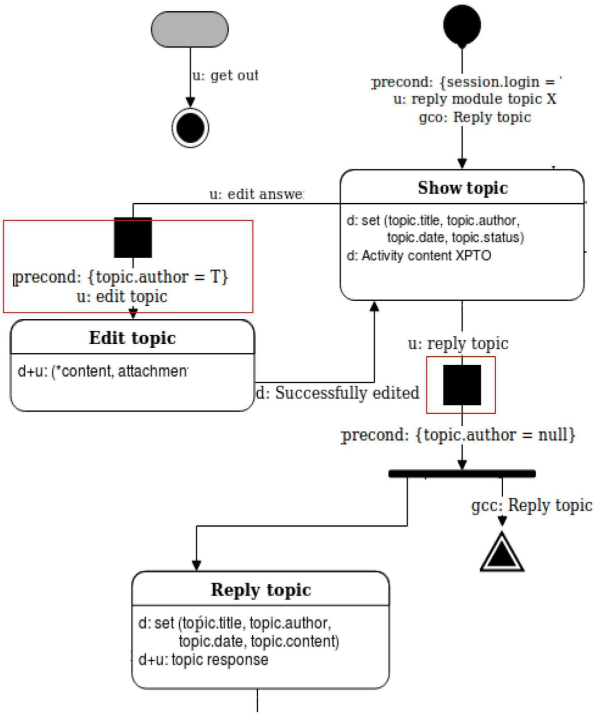
u: get out
precond: {session.login = '
u: reply module topic X
gco: Reply topic
u: edit answe
Show topic
d: set (topic.title, topic.author, topic.date, topic.status)
d: Activity content XPTO
precond: {topic.author = T}
u: edit topic
Edit topic
d+u: (*content, attachmen
d: Successfully edited
u: reply topic
precond: {topic.author = null}
gcc: Reply topic
Reply topic
d: set (topic.title, topic.author, topic.date, topic.content)
d+u: topic response

Figure 5. Diagram forum module.

Figure 6, is the **SP3 - Omission** defect. There is no systematic process for interpreting user transition speech (read topic). Typically, in educational platforms forums, when the user selects a topic they want to read, the system processes this action and redirects the user to another interface with the content of the selected topic.

Figure 6 shows the diagram of the general forum. The defects found in this diagram are highlighted in red; the first one identified, in Figure 6, is also of the omission type, **TUB3 - omission**; there is no user transition speech for the system process, located above the two system processes, as highlighted in Figure 6. This defect incurs the defect **SP1 - incorrect fact**, as the system process was not used to interpret a user's transition speech.

The second defect found was **SP2 - incorrect fact**, highlighted in Figure 6, in the two user transition lines, *u: cancel*. The designer's proposal's rupture recovery speech must be enunciated; therefore, the designer's transition speech.

Finally, the defect **SN6 - ambiguity**, identified in the scene read topic, is represented twice in the diagram. It is suggested to link the transition speech *u: read topic found*, as well as the speech *u: delete topic* with the scene read topic at the top of the diagram. During the evaluation, we considered it as a failure linked to the aesthetic organization of the diagram.

In this diagram, a new element is highlighted in yellow, (*cf.* Figure 6). This element was created to consider the need for a user to influence the interaction flow of another user (during the interaction). This element is named *outgoing message indicator* (OMI), in which a label and the message to be sent to other user(s) are defined (De Souza, Barbosa 2014; De Souza, Barbosa 2015).

In the course interaction modeling diagram, seven defects were identified, as shown in Figure 7. The first two, highlighted on the top of the diagram, refer to the defects **TUB3 - omission** and **SP1 - incorrect fact**, referring to the lack of content in the transition speech. The statement d: do want to complete the evaluation? was identified as the second defect, **SP1 - incorrect fact**, as the system process was not used to interpret a user's speech, but a designer's speech.

The lines u: perform evaluations, u: submit evaluation, u: abandonment form, and *u: yes* were also identified as defects **SP2 - fact incorrect**. The designer's agent's statement was not used for a transition speech, but a user speech. The highlights in the system processes elements have the defect of the type **SP5 - incorrect fact,** as the speech of rupture recovery was not identified in the system processes.

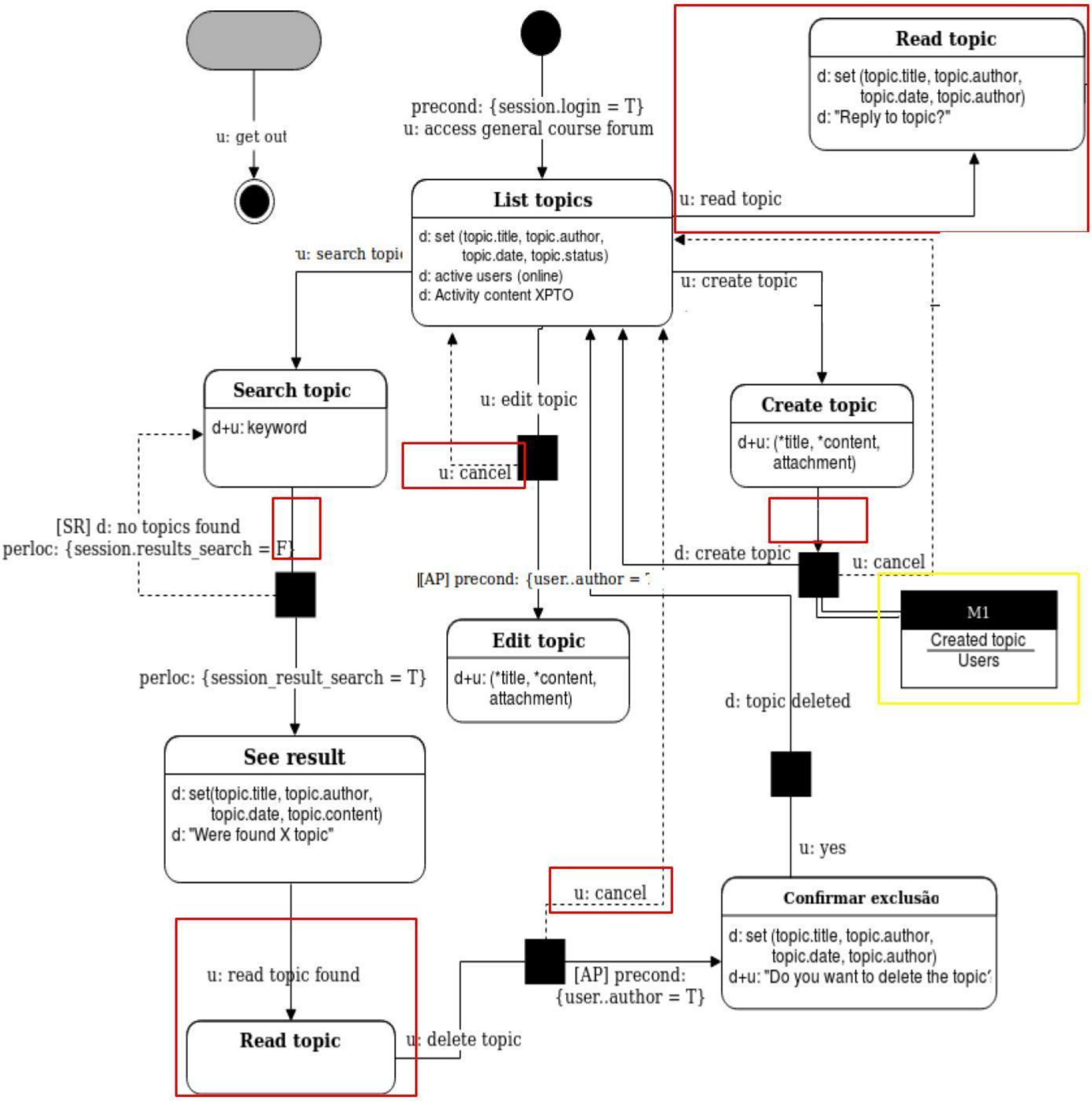
precond: {session.login = T}
u: access general course forum
u: get out
Read topic
d: set (topic.title, topic.author, topic.date, topic.author)
d: "Reply to topic?"
u: read topic
List topics
d: set (topic.title, topic.author, topic.date, topic.status)
d: active users (online)
d: Activity content XPTO
u: search topi
u: create topic
Search topic
d+u: keyword
u: edit topic
u: cancel
Create topic
d+u: (*title, *content, attachment)
[SR] d: no topics found
perloc: {session.results_search = F}
d: create topic
u: cancel
[AP] precond: {user..author = T
M1
Created topic
Users
Edit topic
d+u: (*title, *content, attachment)
perloc: {session_result_search = T}
d: topic deleted
See result
d: set(topic.title, topic.author, topic.date, topic.content)
d: "Were found X topic"
u: yes
u: cancel
Confirmar exclusão
d: set (topic.title, topic.author, topic.date, topic.author)
d+u: "Do you want to delete the topic?
u: read topic found
[AP] precond: {user..author = T}
Read topic
u: delete topic

Figure 6. Diagram General forum.

In Figure 7, the closing element of the conversation about the final goal is highlighted in purple. This element is not included in the evaluation by MCheck; for this reason, it is highlighted in another color. The representation of this element must not contain arrows, as shown in Figure 7.

The last diagram created for the development of a MOOC platform was the course module. Defects of the type **SP1 - incorrect fact**, **SP2 - incorrect fact**, **TUB1 - incorrect fact**, and **TUB3 - omission** were identified.

In Figure 8 the first defect highlighted is in the direction of the recovery speech *u: cancel*, being of the type **TUB - incorrect fact**. The direction of speech must be presented in the opposite direction. After processing the system, a defect of the type **SP2 – incorrect fact** was identified, in which the transition speech after this element belongs to the user (*u: confirm evaluation execution*).

After the select content scene, there is no user transition speech before the system process. Classifying as defects of the type SP1 - incorrect fact and TUB3 - omission due to the lack of content in the transition speech.

Another highlight is the designer's speech d: perform module evaluation, which is not associated with any transition or scene speech. After the fork (*cf.* Figure 8), there are three transition lines: u: watch video, u: listen to audio, and read text. These lines were identified with the defect **SP2 - incorrect fact**. After the process of the system, transition speeches from the designer and not the user should be used.

The defect **SP5 - incorrect fact** was identified in the system process below the user transition statement *download file X*. The message did not use burst recovery speech in case the file download is not completed successfully.

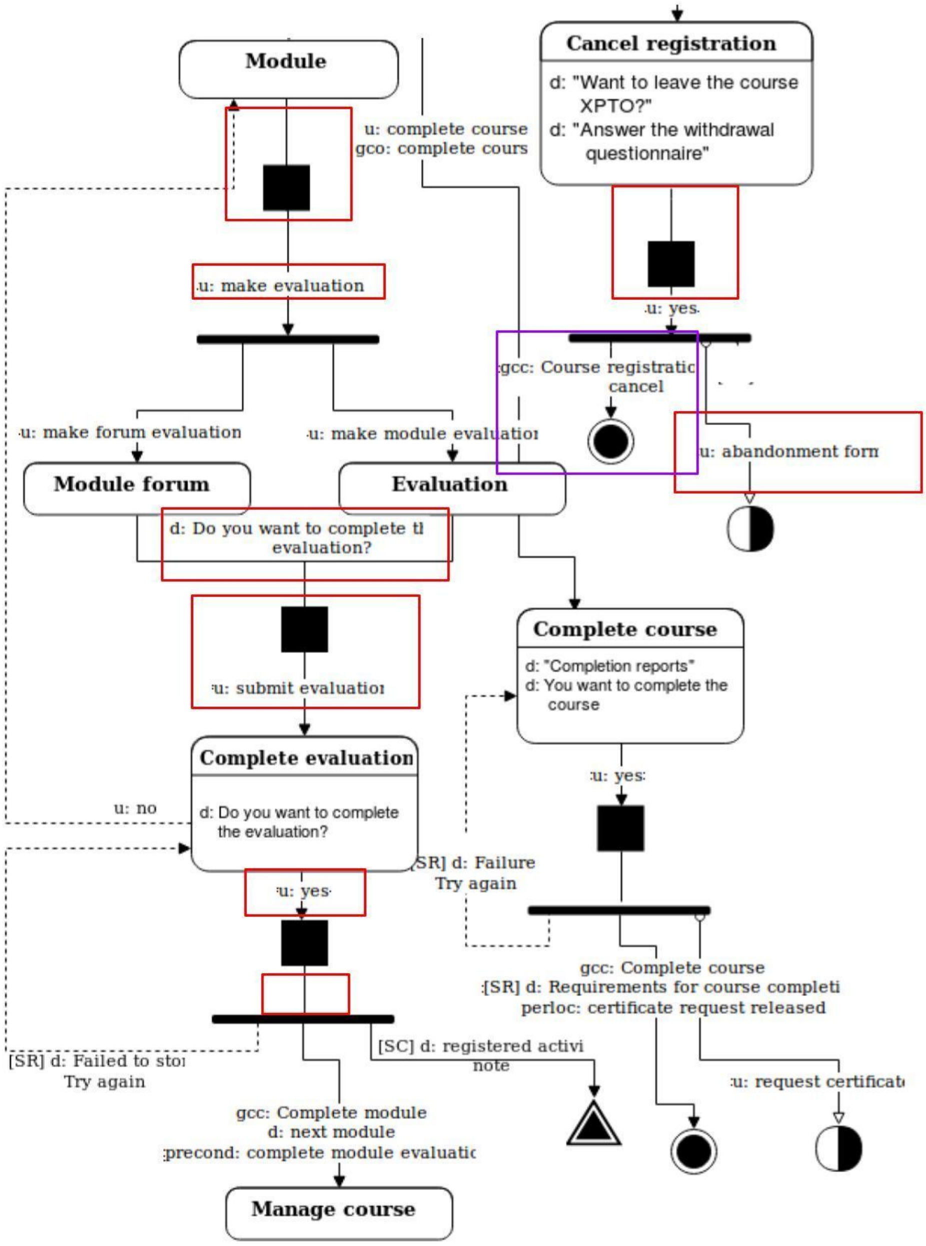
Module
u: complete course
gco: complete cours
Cancel registration
d: "Want to leave the course XPTO?"
d: "Answer the withdrawal questionnaire"
u: make evaluation
u: yes
gcc: Course registratic cancel
u: make forum evaluation
u: make module evaluatioı
u: abandonment forı
Module forum
Evaluation
d: Do you want to complete tl evaluation?
u: submit evaluatioı
Complete course
d: "Completion reports"
d: You want to complete the course
Complete evaluation
d: Do you want to complete the evaluation?
u: no
u: yes
SR] d: Failure Try again
u: yes
gcc: Complete course
[SR] d: Requirements for course completi
perloc: certificate request released
[SR] d: Failed to stoı Try again
[SC] d: registered activi note
u: request certificat
gcc: Complete module
d: next module
precond: complete module evaluatic
Manage course

Figure 7. Diagram course.

Highlighted in Figure 8 in purple color are two closing elements of the final goal conversation. The elements were represented wrongly, as the two arrows are with the direction represented in the diagram, which should not happen in the representation of this element.

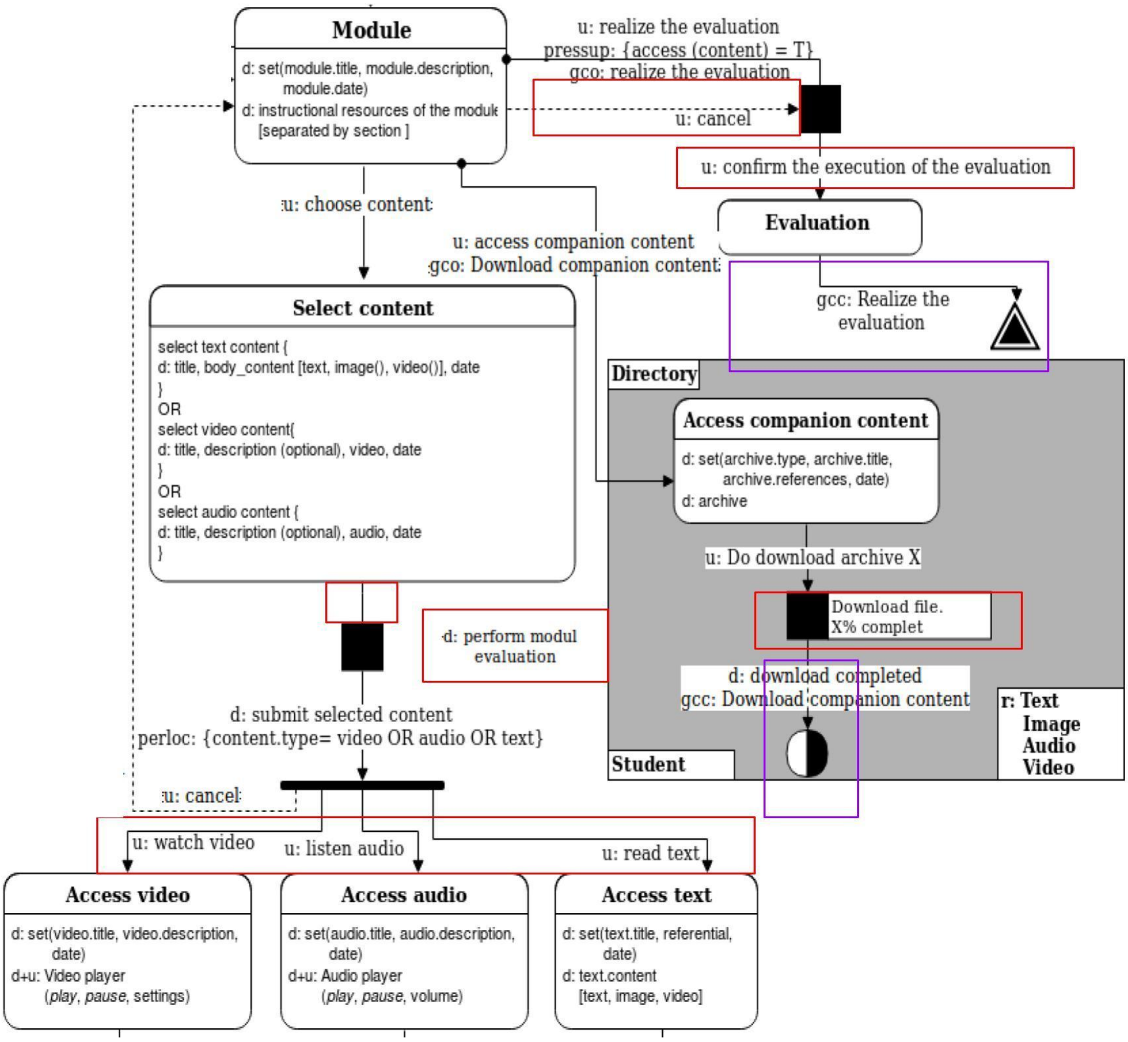
Module
d: set(module.title, module.description, module.date)
d: instructional resources of the module [separated by section]
u: realize the evaluation
pressup: {access (content) = T}
gco: realize the evaluation
u: cancel
u: confirm the execution of the evaluation
Evaluation
gcc: Realize the evaluation
u: choose content
u: access companion content
gco: Download companion content
Select content
select text content {
d: title, body_content [text, image(), video()], date
}
OR
select video content{
d: title, description (optional), video, date
}
OR
select audio content {
d: title, description (optional), audio, date
}
Directory
Access companion content
d: set(archive.type, archive.title, archive.references, date)
d: archive
u: Do download archive X
Download file. X% complet
d: perform modul evaluation
d: download completed
gcc: Download companion content
r: Text Image Audio Video
Student
d: submit selected content
perloc: {content.type= video OR audio OR text}
u: cancel
u: watch video
u: listen audio
u: read text
Access video
d: set(video.title, video.description, date)
d+u: Video player (play, pause, settings)
Access audio
d: set(audio.title, audio.description, date)
d+u: Audio player (play, pause, volume)
Access text
d: set(text.title, referential, date)
d: text.content [text, image, video]

Figure 8. Course module diagram.

5 Discussion of Results

After the inspection, 35 defects, spread across incorrect fact, omission, ambiguity, extraneous information, and inconsistency, were identified (*cf.* Figure 9). Furthermore, in the inspection of the first version of MoLIC, there were 23 defects only of the incorrect fact type.

The Incorrect Fact defect identifies that in several inspected elements, it is not possible to identify a pattern for the “appearance”. The Incorrect Fact was identified in the elements: System Process, Transition Speeches, Signs, Ubiquitous Access, and Scene. In the Dialogue, Opening Point, and Closing Point elements, no defects of this nature were identified.

Possibly the Incorrect Fact was the most evident defect due to the repetition of elements, such as the Ubiquitous Access that was inserted in several diagrams to represent the designer's intention to allow the user to exit any interface.

The System Processing element had the highest number of errors, with Incorrect Fact being the label indicated to specify the failure. In this case, we realized that the interaction flow was incorrect and needed adjustment before it was actually conceived.

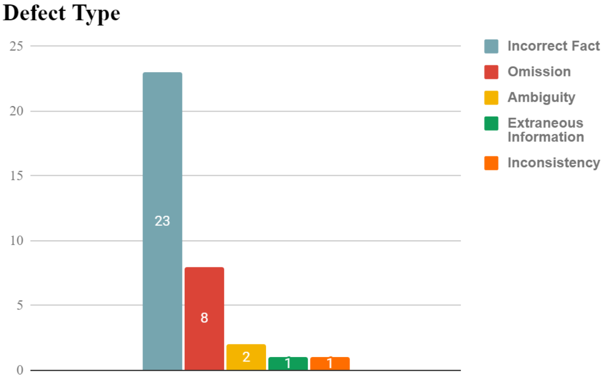

Figure 9. Number of defects found.

The system process was the element that reported the most defects, adding a total of 18 reported defects. Note that bifurcation's element may have influenced the number of defects identified as the system process was the only element directly affected by the insertion of new elements to MoLIC (see Figure 9).

The elements that did not report any defects were dialogues, opening point, and closing point, as can be seen in the Figure 10.

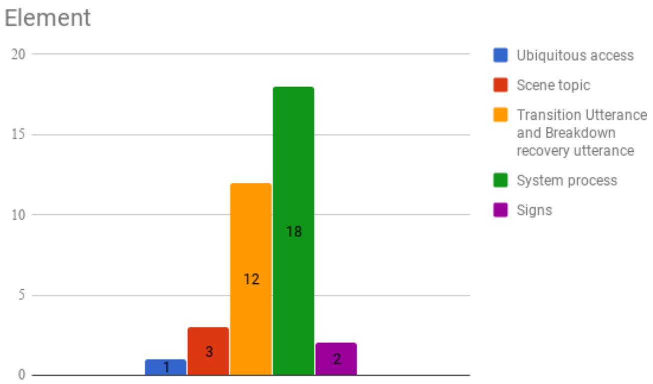

Figure 10. Number of defects of each element.

The check item that returned the most number of defects was the **SP2**, regarding the lack of statements by the designer's agent, both for transition and recovery. The transition and recovery speeches were improperly used in the evaluated diagrams; most of them contained addresses by the user instead of the designer. Another well-mentioned verification item in the identification of the defects was the **TUB3**, referring to the lack of content of the transitional utterance (*cf.* Figure 11).

Through the evaluation, it was possible to identify elements in the third version of MoLIC that the technique does not support.

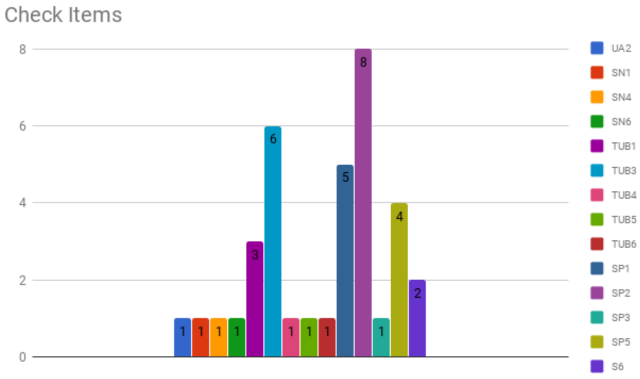

Figure 11. Number of defects per check item.

Three of the defects identified in the interaction design evaluation using the MCheck technique could be propagated and become communicability disruptions in the user's interaction with the implementation of the artifact.

With the results of the evaluation, it was possible to identify the need for evolution of MCheck, so that it incorporates the elements that have been added to MoLIC in the course of its evolutions (Do Rêgo 2018). Next, it will be indicated which MCheck does not support elements of MoLIC: (a) point of contact with another (role of) user, (b) point of contact with external system, (c) speech of influence, (d) talks about transition with opening of conversation about final goal (f) bifurcation, (g) scene in its minimum form, (h) empty scene and (i) Outgoing Message Indicator (OMI), (j) Incoming Message Indicator (IMI), (l) Shared Space Indicator (SSI) and (m) evaluation completed (see Figure 12).

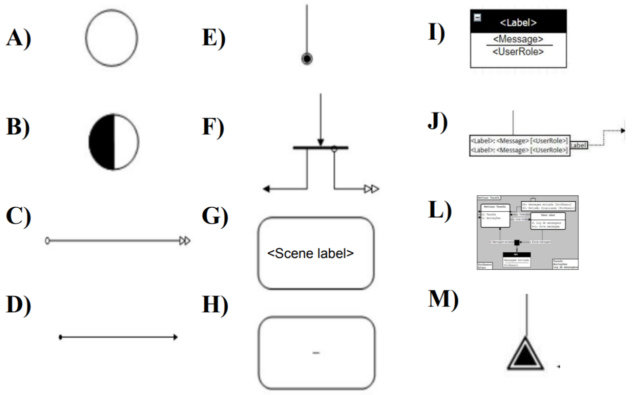

Figure 12. New MoLIC elements.

Thus, verified checklist cards were created for each new element added throughout the evolution of the MoLIC and for the elements added in the version created for modeling a MOOC. These elements are shown in Figure 12.

The new checklist cards of the new version of MCheck are presented in the Appendix link.

6 Final Considerations and Future Work

We presented the experience report of the evaluation of diagrams generated by the MoLIC language; the evaluation aimed to analyze the quality of metacommunication, at design time, of MoLIC diagrams using the MCheck technique.

However, considering that the application of MCheck is limited to the inspection of diagrams of the first version of MoLIC (Paula 2003), attesting that the technique was developed for this version, but it is necessary for MCheck to follow the evolution of MoLIC to identify defects. Therefore, in this study, we indicate new elements to be added to MCheck (Silva 2005; Araújo 2008; Garrido 2018) and present the suggested evolution of the technique, which is the main contribution of the article.

The MCheck technique was used to achieve this goal. However, during the evaluation process, the need to update the technique was identified. To follow the evolution of MoLIC, it should be noted that the modeling of the evaluated diagrams added elements inherent to the educational context as a result of the hybrid design approach (instructional and articulated interaction).

This assessment mitigated some communicability breaks that would be propagated in the course interface. In particular, the lack of recovery talks should help the student-user during their interaction. These transition lines must be direct from the designer's agent, informing them how to proceed during their dialogue with the system, thus impacting the communication quality of the MOOC.

The students' abandonment in MOOCs can be attributed to social, pedagogical problems, interests of the participants, and technology. Therefore, the importance of designing and evaluating the interaction design of online educational platforms is justified. Thus it is possible to anticipate several problems that favor the abandonment of students of these courses.

The discontinuity of the evolution of MCheck concerning MoLIC, impaired the effectiveness of this technique, mainly in the design of multiuser systems, a category in which interactive educational systems such as the MOOC are included. They are evidencing the need to adapt/promote the MCheck technique in alignment with MoLIC.

The main difficulties were the evolution of the technique after the identification of the MoLIC elements, because apart from the elements that were added throughout the evolution of the MoLIC, elements referring to the context of the system that was being modeled were inserted, in this case a MOOC.

Even being promoted, it is suggested the evaluation of the MCheck proposed by the authors and a new application to guarantee its effectiveness, both being suggestions for future work.

6.1 Threats to Validity

In this study, possible threats to validity were identified that may limit the ability to understand the data obtained (Alves, *et al.* 2010). Therefore, these should be minimized.

- Evaluators' bias: the evaluation of the results found in the execution of the MCheck according to the evaluators' understanding. The identified communicability ruptures contain the applicator's understanding bias.

One way to mitigate this threat is to conduct the review by an evaluator external to the inspection who knows the modeling technique using MoLIC diagrams but did not participate in the modeling process of the diagrams created in this study. Thus, communication breakdowns and bias on the part of the evaluators involved would be avoided.

- Application context: the identified ruptures fit the context of a MOOC course and developed platform. Not applicable to other contexts of online courses.

Highlight that the modeling of the evaluated MoLIC diagrams does not apply to any online course platform, as the evaluated diagrams were explicitly created to meet the needs of many participants, as provided for in a MOOC. Other factors are also relevant in the evaluation and were highlighted in the report.

Acknowledgements

We thank the members of the Semi-Participatory Interaction Design Laboratory (SPIDeLab), Research and Extension Group in Informatics and Education and Society - Onda Digital.

We also thank the Coordination for the Improvement of Higher Education Personnel (CAPES) for funding this research, under the codes nº 88887.464930/2019-00 DS and nº 88882.453912/2019-01 DS to the National Council for Scientific and Technological Development (CNPq) for the two Scientific Initiation grants and the Research Productivity Grant.

References

Alves, V; Niu, N; Alves, C; Valença, G. (2010). Requirements engineering for software product lines: A systematic literature review. Information and Software Technology, v. 52, n. 8, p. 806-820.

Araujo, A. C. I. C. (2008). *Supporting the Design and the Interpretation of Human-Computer Interaction Diagrams Represented in MoLIC.* Master's thesis. Department of Informatics, Pontifical Catholic University of Rio de Janeiro.

Costa, F. A; Santos, A. M; Silva, A. G; Viana, J. (2015). Guidelines for designing mooc courses. MEC. *Experiências de Inovação Didática no Ensino Superior. Lisboa: MEC*, p. 327–342.

Da Silva, B. S; Barbosa, S. D. J. (2007). *Designing human-computer interaction with molic diagrams–a practical guide.* Monograph in Computer Science. Department of Informatics, Pontifical Catholic University of Rio de Janeiro.

Damian, A. L. (2016). *Techniques for inspection of MoLIC diagrams.* Master's thesis. Federal University of Amazonas.

De Carvalho, A. P; Pereira, F. H. S; Assunção, O. B; Pereira, A. F; Prates, R. O. (2019). An analysis of molic's consolidation. In: *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems*, ACM, New York, NY, USA, p. 1–15. doi: 10.1145/3357155.3358461.

De Souza, C. S. (2005). *The Semiotic Engineering of Human-Computer Interaction.* MIT Press.

De Souza, C. S; Leitão, C. F. (2009). Semiotic Engineering methods for scientific research in hci. Synthesis Lectures on Human-Centered Informatics 2, p. 1–122.

De Souza, L. G; Barbosa, S. D. J. (2014). Extending molic to support the design of collaborative systems. In: *Companion Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems, Sociedade Brasileira de Computação*, p. 25–28. doi: 10.5555/2738165.2738174.

De Souza, L. G; Barbosa, S. D. J. (2015). *Extending MoLIC to Support the Design of Collaborative Systems*. Master's thesis. Department of Informatics, Pontifical Catholic University of Rio de Janeiro.

Do Rêgo, B. B; Garrido, F. A; Matos, E. de S. (2018). Moodle as a mooc environment: guidelines for interaction redesign. RENOTE - Revista Novas Tecnologias na Educação, v. 16, n. 1, p. 73-82. doi: 10.22456/1679-1916.86050.

Garrido, F. A. (2018). *A unified design approach to educational and communication modeling in MOOC: an exploratory study.* Master's dissertation federal university of bahia. Federal University of Bahia.

Garrido, F; Do Rêgo, B. B; Matos, E. (2018a). Artifact-oriented instructional design: a participatory and distributed approach. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, v 29. n. 1, p. 258-267. doi: 10.5753/cbie.sbie.2018.258.

Garrido, F. A; Do Rêgo, B. B; Matos, E. de S. (2018b). Modeling the (human-computer) interaction of a discussion forum for mooc: Molic in use. *RENOTE - Revista Novas Tecnologias na Educação*, v. 16, n. 2, p. 321–330. doi: 10.22456/1679-1916.89250.

Garrido, F; Do Rêgo, B. B; Matos, E. (2021). Interaction design integrated with instructional design: modeling a mooc environment through molic. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, p. 598-608. doi: 10.5753/sbie.2021.217573.

Lopes, A; Marques, A. B; Barbosa, S. D. J; Conte, T. (2015a). Molveric: An inspection technique for molic diagrams. In: *Software Engineering and Knowledge Engineering (SEKE)*, p. 13–17. doi: 10.18293/SEKE2015-069.

Lopes, A; Marques, A. B; Barbosa, S. D. J; Conte, T (2015b). Molveric cards feasibility study: A techniques for inspecting molic diagrams. *Proceedings of XIV Brazilian symposium on Human factors in computing systems*, p. 203–212. doi: 10.1145/3148456.3148477

Lopes, A; Valentim, N; Moraes, B; Zilse, R; Conte, T. (2018). Applying user-centered techniques to analyze and design a mobile application. Journal of Software Engineering Research and Development, v. 6, n. 1, p.1-23. doi: 10.1186/s40411-018-0049-1

Lowgren, J. (2013). Interaction Design-brief intro, in: Soegaard, M., Dam, R.F. (Eds.), *The Encyclopedia of Human-Computer Interaction. 2 ed*. The Interaction Design Foundation, Aarhus, Dinamarca. chapter 1.

Paula, M. G. (2003). *Human-computer interaction design based on models based on Semiotic Engineering: construction of an interaction model.* Master's thesis. Department of Informatics, Pontifical Catholic University of Rio de Janeiro.

Prates, R. O; Barbosa, S. D. J. (2007). Introduction to the theory and practice of human computer interaction based on Semiotic Engineering. In *T. Kowaltowski & K. Breitman (orgs.). Jornadas de Atualização em Informática, JAI,* p. 263–326.

Rosa, J. C. S; Matos, E. (2016). Semio-participatory framework for interaction design of educational software. In: *Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems*, p. 1–10. doi: 10.1145/3033701.3033734.

Santos, S. H. N; Vieira, E. S; Serra, A. C; Rivero, L. (2018). Applying interaction modeling inspection techniques: Lessons learned by undergraduate students. *JIM 2018 - VII Jornada de Informática do Maranhão (VII Maranhão Computer Journey).*

Silva, B. S. d. (2005). *MoLIC Second Edition: Revision of a Modeling Language for Human-Computer Interaction.* Master's thesis. Department of Informatics, Pontifical Catholic University of Rio de Janeiro.

Thaler, A; Fialho, F. A. P. (2015). Interaction design: An approach to user-centered educational games. *Blucher Design Proceedings*, v. 2, n. S/N, p. 223–230.

Travassos, G; Shull, F; Fredericks, M; Basili, V. R. (1999). Detecting defects in object-oriented designs: using reading techniques to increase software quality. ACM sigplan notices, v. 34, n. 10, p. 47-56. doi: 10.1145/320385.320389