MoLVERIC: An Inspection Technique for MoLIC Diagrams

Adriana Lopes, Anna Marques and Tayana Conte
USES Research Group
Instituto de Computação, Universidade Federal do
Amazonas (UFAM)
Manaus, AM - Brazil
{adriana,anna.beatriz,tayana}@icomp.ufam.edu.br

Simone Diniz Junqueira Barbosa
Semiotic Engineering Research Group
Departamento de Informática, PUC-Rio
Rio de Janeiro, RJ - Brazil
simone@inf.puc-rio.br

Abstract— **During interaction design, interaction models are developed to help design adequate user interaction with the system. MoLIC (Modeling Language for Interaction as Conversation) is a language used to represent an interaction model, which can then be used as a basis for building other artifacts, such as mockups. However, inspections are necessary to verify whether the MoLIC diagrams are complete, consistent, unambiguous, and contain few or no defects, to avoid propagating preventable defects to derived artifacts. In this paper, we present MoLVERIC, a technique for the inspection of MoLIC diagrams that uses cards with verification items and employs principles of gamification. Furthermore, we discuss the results of a pilot study conducted to analyze the feasibility of this technique.**

Keywords-component; Interaction Design; Interaction Modeling; Verification; Inspection Technique.

I. INTRODUCTION

Interaction design aims to design systems that are easy to learn, effective when used and capable of providing a rewarding experience to the user [1]. In this context, Semiotic Engineering [2], a theory of Human-Computer Interaction, deals with interaction as a communication process between the user and the system, through its user interface. Based on Semiotic Engineering, Barbosa and Paula [3] proposed MoLIC (acronym for Modeling Language for Interaction as Conversation), a language to model this interaction. MoLIC diagrams can be used by different practitioners involved in the development of systems for modeling a global view of the application's apparent behavior. Moreover, MoLIC diagrams can serve as a basis for the construction of other artifacts in the development of interactive systems, such as mockups. Santana et al. [4] proposed the use of a communication theory called Grice's Cooperative Principle [5] for inspection of MoLIC diagrams with the focus on user communication.

However, MoLIC diagrams should be also verified with respect to their consistency, completeness and comprehensibility in order to reduce the number of defects and to prevent them from propagating to derived artifacts. To investigate the quality of MoLIC diagrams, we conducted a preliminary study that has identified several defects which had been inserted during interaction modeling. Through such inspection, the propagation of defects in MoLIC diagrams can be avoided, reducing the cost of correcting such defects in later stages of the software development process [6].

In this paper we present MoLVERIC, a technique for inspecting MoLIC interaction diagrams. The MoLVERIC technique was developed based on the defects that were found in a preliminary study. The purpose of MoLVERIC is to provide a simple way to identify defects in MoLIC diagrams, so that the technique can be easily adopted by both academy and industry. With this technique, we intend to prevent possible defects from being transferred to artifacts that are constructed based on the MoLIC diagrams. To assess whether MoLVERIC can support inspectors in detecting defects, we conducted a pilot study, whose results have provided evidence of the feasibility of MoLVERIC to inspect MoLIC diagrams.

The remainder of this paper is organized as follows. Section II presents the MoLIC language. Section III describes the defect types found in MoLIC diagrams in our preliminary study. Then, Section IV presents the MoLVERIC technique, and Section V describes the pilot study conducted to evaluate MoLVERIC. Finally, we present some concluding remarks and discuss future work.

II. MoLIC

MoLIC is based on Semiotic Engineering [2], a theory of HCI with particular focus on the communication between the designer and the user mediated by interactive systems, through the designer's deputy, which is the user interface. The designer's deputy "talks" with the user, enabling the mediated designer-to-user communication about the designer's view and design decisions. Because the designer-to-user message is about the messages users can exchange with the user interface, it is called a *metacommunication message*. MoLIC was devised to represent this message.

A MoLIC diagram can be created after the requirements elicitation, within the analysis stage of the software development process. The purpose is to promote the designers' and developers' reflection on the interaction alternatives that they intend to provide to the users [7]. To illustrate the MoLIC diagrammatic notation [7], Figure 1 represents a diagram of a simple system for calculating a person's Body Mass Index (BMI). The basic elements of a MoLIC diagram are the following:
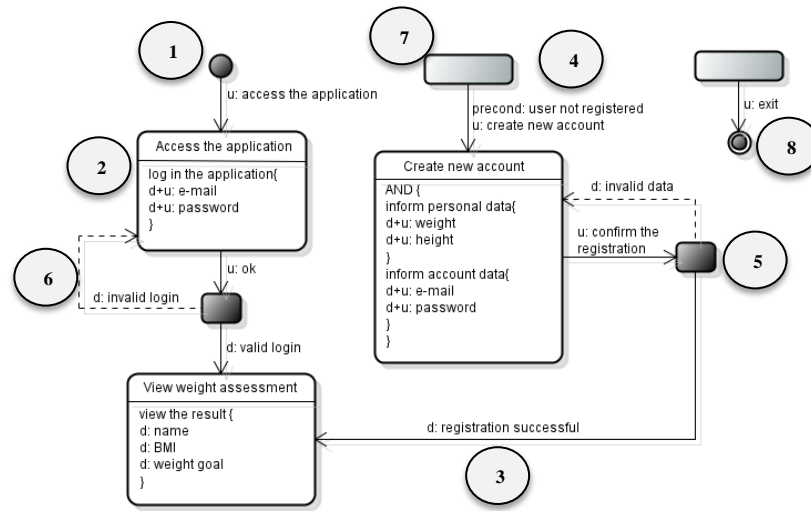
Figure 1. A MoLIC diagrams example.

1. *Opening point:* Indicates where the interaction can start, *i.e.*, when the user accesses the system. It is represented by a filled black circle.

2. *Scene:* Represented in the diagram as a rounded rectangle. The scene shows the moment in the interaction where the user decides how the conversation should proceed. The top compartment contains the topic of the scene and represents the user's goal. The second compartment details the following elements:

   a) *Signs:* represent the information involved in the utterances issued by the user (*i.e.*, user input) and by the designer's deputy (*i.e.*, system output) during the dialogues. In Figure 1, we have the following signs in the "Access the application" scene: "e-mail and password".

   b) *Utterances:* constitute the dialogue and specify who is emitting the sign: whether it is the user (u) or the designer's deputy (d). The signs issued only by the designer's deputy (e.g, system output) are preceded by "d". In Figure 1, we have the following signs and utterances in the "View weight assessment" scene, e.g. "d: name, d: BMI and d: weight goal", all emitted by the designer's deputy, because they are just for providing information to the user. When the designer allows the user to talk about the signs, e.g., when it involves user input, we say that both the designer and the user emit the sign, which is then preceded by "d+u". In Figure 1, we have such signs in the "Access the application" scene: "d+u: e-mail, d+u: password", for example.

   c) *Dialogues:* compose a conversation about a topic, and consist of utterances on signs. In Figure 1, one example of dialogue is "view the result" (in "View weight assessment").

   d) *Structures of dialogues:* in some cases, the dialogues can be composed by other dialogues according to some structure. In these cases, these structures can be represented by the reserved words SEQ, XOR, OR or AND. The SEQ structure represents the dialogues that must be exchanged in the specified sequence. The XOR structure represents mutually exclusive dialogues. The structure OR represents the choice of exchanging one or more dialogues. The structure AND represents the use of all dialogs, but not in a predefined sequence. In Figure 1, the AND structure represents the use of all dialogs "inform personal data" and "inform account data".

3. *Transition Utterance:* Represents turn-taking, or rather turn-giving, where either the user or the designer's deputy gives the turn to the other, for instance, to change the topic of the conversation, as described below:

   a) *User Utterance:* represents the user's intent to proceed with the conversation in a given direction. It is represented by an arrow in the diagram, labeled with a user utterance indicator (u:), e.g. "u: ok" in Figure 1.

   b) *Designer Utterance:* represents the designer's deputy's answer to a user utterance, typically provided after a system process. It is represented by an arrow in the diagram, labeled with a designer utterance indicator (d:) e.g. "d: valid login" in the Figure 1.

4. *Precond:* represents a necessary precondition in the diagram. In Figure 1, the user can only create an account if (s)he is not yet registered. This precondition is represented before the user utterance through "precond: user not registered".

5. *System process:* It is represented through a black box in the diagram. It represents the internal processing (of a user request) which needs to provide adequate feedback to the user, *i.e.*, when there are different outcomes possible.

6. *Breakdown recovery utterance:* is a type of utterance provided to help the user recover from a communication breakdown. It is represented by a dashed directed line in the diagram with the corresponding utterance, e.g. "d: invalid data" in Figure 1.

7. *Ubiquitous access:* represents an opportunity for the user to change the topic of the conversation from any other scene, to achieve an objective different from the current one. It is represented through a gray rounded rectangle.

8. *Closing point:* represents the end of the interaction, when the user leaves the system. It is represented as a filled black circle within a circle with no padding.

III. TYPES OF DEFECTS IN MOLIC DIAGRAMS

The types of defects that can be found in a MoLIC diagram were defined based on the taxonomy presented by Travassos et al. [6], as shown in Table I. Using a taxonomy of defects is important in order to assist the inspectors (practitioners who carry out the inspection) in the identification and categorization of defects.

TABLE I.   DEFECTS TAXONOMY FOR MoLIC DIAGRAMS

Types of Defects	Description of Defects
Omission	Omission or negligence of any information necessary to solve the problem in the interaction diagram.
Ambiguity	Unclear definition of a certain information in the interaction diagram, which may lead to multiple interpretations.
Incorrect Fact	Misuse of the interaction diagram elements.
Inconsistency	Conflicting information between the interaction diagram elements and the information needed to solve the problem.
Extraneous Information	Unnecessary information included in the interaction diagram (i.e., information that is not needed to solve the problem).

We conducted a preliminary study with 13 subjects in order to evaluate the quality of some sample MoLIC diagrams. Study subjects were undergraduate students (in their final year) and graduate students in a Computer Science course. Before the study, all subjects received training in using the MoLIC language to model the interaction of a system, because the subjects had no experience with MoLIC diagrams. Each subject individually built the MoLIC diagram, using computers with a MoLIC designer tool[1] installed.

After the study execution, two experts in MoLIC diagrams verified the produced diagrams. These experts found the defects in the artifacts and categorized them according to the taxonomy shown in Table I. During the analysis, the repeated defects were united. In the total we identified: 13 omissions, 5 extraneous informations, 1 ambiguity, 7 incorrect factc and 5 inconsistencies.

TABLE II.   DEFECTS FOUND IN MOLIC DIAGRAMS FOR EACH TYPE OF DEFECT

Type of Defect	Subjects (S)	Example of Defect Found in the MoLIC Diagram
Omission	S2, S4, S5, S12	Did not use the notation of the utterance for the user (u:) and the designer's deputy (d:).
Ambiguity	S12	Used two user transition utterances for the same goal, thus providing multiple interpretations for the user request.
Incorrect Fact	S4, S8, S9, S11	Used verbs that do not represent the user goals.
Inconsistency	S1, S5, S6, S11,	Used a transition arrow direction inconsistent with the sequence of interaction scenes.
Extraneous Information	S4, S8,	Represented some scenes that were not in the context of the interaction scenario.

In Table II, we present one detailed example of each type of defect found in MoLIC diagrams. In addition, the defects are associated with the subjects who developed the diagrams containing the defects. As noted in this study, MoLIC diagrams

may contain defects, which can impair the understanding of the practitioners involved during the development of the systems. Therefore, it is important to perform inspections in MoLIC diagrams before they are used in the next phases of the systems development, for creating new artifacts such as mockups, for instance.

IV. MOLVERIC

Inspection of system artifacts during development has been shown to improve the quality of the system and reduce development costs. An inspection is a method for identifying defects in early stages of the development [6]. Conducting inspections is essential because design defects can directly affect the quality of the systems [6]. One of the most widely used methods in the inspection is Checklist [8]. Checklists provide support for inspectors during the defects detection through verification items [8].

MoLVERIC is a checklist-based inspection technique, developed with the goal of assisting practitioners in the inspection of MoLIC diagrams. All verification items of MoLVERIC were developed based on the defects found in the preliminary study, as described in Section III. The verification items assess both the consistency of MoLIC diagrams with the interaction scenario/system requirements; and the notation used in the MoLIC diagrams. To motivate the inspection of MoLIC diagrams, MoLVERIC employs gamification [10] techniques. Each card corresponds to a verification item and includes the number of points awarded to the inspector each time he/she finds the defect described in the card. Each verification item assists the inspector in reporting the type of the identified defect. With respect to the score, items that verify defects that compromise the purpose or understanding of the diagram award 20 points, whereas items that verify a syntax defect, which does not compromise the objective or understanding of the diagram, award 10 points. The verification cards are divided into categories corresponding to the elements of the MoLIC diagram, such as Scene, Transition Utterance and Signs. During the development of the verification items for each category of the MoLIC diagram elements, we noted that some defects were related to more than one element. Therefore, we developed different verification items for single elements and for related elements. To do so, we developed two types of cards: Regular Cards and High Cards.
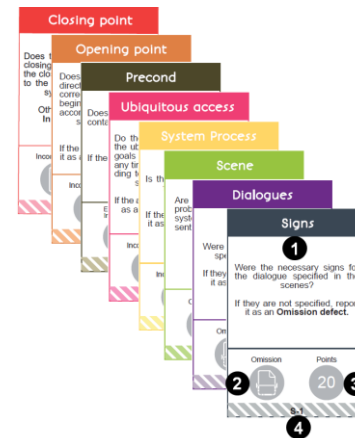

Figure 2: Verification items from MoLVERIC: Regular Cards

Regular Cards have one-to-one correspondences to elements of the MoLIC diagram. There are Regular Cards for the following elements: Scene, System Process, Opening Point, Closing Point, Ubiquitous Access, Precond, Dialogues and Signs. In Figure 2, the Regular Cards are presented using the following structure: (1) Description of the verification item, to assist the inspector in the identification of defects; (2) Type of defect to be reported; (3) Points of the card and (4) Code of the verification item.

High Cards have verification items for elements related to other elements in the MoLIC diagram. The goal of the High Cards is reduce the inspection time for elements that are related in the diagram. There are High Cards for the elements Transition Utterance and Breakdown Recovery Utterance. A Transition Utterance element is related to the elements: Scene, System Process, Opening Point, Closing Point and Ubiquitous Access. Figure 3 shows an example of an inspection using a High Card for the Transition Utterance element, where the following verification item is used for the related elements: "*Do the utterances show who uttered them ("u:" for the user and "d:" for the designer's deputy)? If the answer is negative, report it as an Omission defect.*"
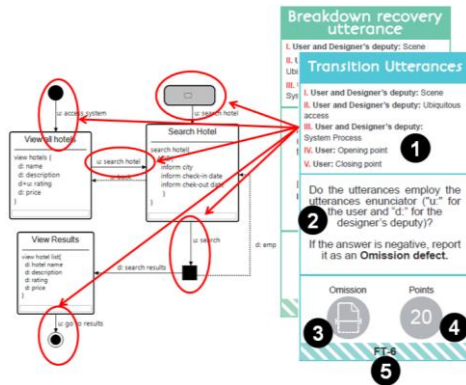

Figure 3: Verification items from MoLVERIC: High Cards.

A Breakdown Recovery Utterance is related with the elements: Scene, System Process and Ubiquitous Access. The High Cards have the following structure in each card, as shown in Figure 4: (1) Elements in the MoLIC diagram related with the elements of the card; (2) Description of the verification item, to assist the inspector in the identification of defects; (3) Type of defect to be reported; (4) Points of the card and (5) Code of the verification item. Furthermore, there are instructions for the inspectors in order to show how to use MoLVERIC.

V. PILOT STUDY OF MOLVERIC

In this pilot study, we did not use other inspection technique for MoLIC diagrams to compare with the MoLVERIC. The reason for this is that the only other known technique (Grice's Cooperative Principle) has a different focus. The pilot study activities are described as follows.

A. Pilot Study Planning

During the planning stage, we defined the resources needed for implementing the study. Therefore, we planned the execution environment, as well as the artifacts, as follows:

- *Environment:* The study was conducted in an academic environment, where new technologies are tested before being transferred to industry [11].

- *Artifacts:* (i) The MoLIC diagrams built in the preliminary study described in Section III; (ii) forms for the subjects to report the identified defects; (iii) instructions for using the technique; (iv) post-study questionnaire to be answered by each subject, to collect their opinions about the technique.

- *Subjects:* For the study, two subjects who had developed MoLIC diagrams in previous projects were chosen to inspect the diagrams. The subjects were graduate students in Computer Science.

- *Training*: The subjects received training on the types of defects and on the use of MoLVERIC.

B. Pilot Study Execution

During the study, each subject executed the inspection individually. After the study, we analyzed the defects reported in the forms and the post-study questionnaires.

C. Results Analysis

After the execution of the pilot study, we verified whether the technique achieved the goal of detecting defects. The oracle of defects contained a total of 24 defects (some defects are repeated in the elements signs, scenes and transition utterance). The number of defects, the inspection time and the indicators of effectiveness and efficiency of each subject are described in Table III. The effectiveness was calculated using the number of defects found by the subjects divided by the total number of defects from the oracle. The efficiency was calculated on the number of defects found divided by the time of inspection of each subject.

TABLE III. RESULTS PER SUBJECTS

Subjects	Number Defects	False Positive	Time Hours	Effectiveness	Efficiency
S1	17	1	1.61	70.83%	10.55
S2	16	3	1.15	66.66%	13.91

Analyzing the effectiveness indicator, we can see that the inspectors were able to identify more than 66% of the defects. This is a good result in terms of effectiveness when compared to the indicators achieved by other inspection techniques [12] and, as such, it indicates the feasibility of MoLVERIC. However, it is still necessary to perform a controlled experiment to compare the effectiveness of MoLVERIC with other techniques for identifying defects in interaction models. Regarding efficiency, the subjects found 10.55 and 13.91 defects per hour. However, as the number of defects is directly dependent on the inspected models, is not suitable to compare the results of efficiency from this pilot study with the results of other techniques.

To understand the opinions of the subjects, the answers to the post-questionnaire were analyzed. Regarding the ease of use of the technique, the subjects indicated the following:

"*MoLVERIC helps to remember the elements that I should verify and the types of defects I should inspect, for example:*

Omission, Incorrect Fact, Inconsistency, Extraneous Information and Ambiguity." (S1)

"The technique provides an inspection guide. This guide does not leave the inspector lost. The technique makes the inspection process fun." (S2)

However, subjects also reported negative aspects regarding the ease of use of the technique:

"I think that it takes a long time to learn to use the technique." (S1)

"I had trouble remembering some terms." (S2)

Regarding the quote from S1 about the negative aspects of the technique, this can be related to the amount of categories and the related items. However, due to the small number of subjects in this pilot study, this result cannot be considered conclusive. This aspect will be examined in future research with MoLVERIC. To understand how the subjects use MoLVERIC during the inspection, they answered the following question in the post-questionnaire: *"Is the structure of MoLVERIC suitable for the inspection of the MoLIC diagrams in the way you inspect an artifact?"*

"I think this is a good way to inspect. The structure of the cards is good." (S1)

"Yes, it directs the structure according to the MoLIC diagram." (S2)

However, during the study, the two subjects had difficulties in understanding the Precond category, an element used to specify a necessary precondition together with the Transition Utterance element. Regarding the other categories during the study, the subjects had no difficulty in the use of the other cards. During the analysis of the defects reported by the subjects, we verified that the subjects had no problems in understanding each inspected element, i.e., they reported them correctly according to the code for each card. However, observing the quotation from S2 about the negative aspects, there is a certain difficulty in understanding the term "issuer" of the signs, which refers to the "d" for the designer and "u" for the user. Furthermore, subjects responded positively to the question "Would you recommend this technique for designers who use the interaction modeling with MoLIC?" Both inspectors indicated that they would recommend the use of MoLVERIC. Based on the analysis of the results of this pilot study, it was possible to obtain indicators of the feasibility of using the technique.

VI. CONCLUDING REMARKS AND FUTURE WORK

The purpose of this paper was to present the results of the pilot study to evaluate the feasibility of MoLVERIC. The analysis study allowed us to identify problems during the use of the technique, as well as terms which were not clear and verification items that were not appropriate. Based on these results, we are improving MoLVERIC, making the verification items clearer. The results of the pilot study provided evidence that MoLVERIC assists in detecting defects. However, these results cannot be considered conclusive, and it is necessary to carry out a controlled experiment with a larger quantity of subjects. Analyzing the perception of the subjects on recommending MoLVERIC, there was mostly positive feedback from the subjects. This may be an indication that the technique is suitable for the inspectors of MoLIC diagrams. The results of the pilot study provided initial evidence to the feasibility of MoLVERIC to inspect MoLIC diagrams.

As future work, we intend to carry out a controlled study with MoLVERIC to reinforce the results obtained in the study pilot. In this next study, we expect to evaluate more precisely the effectiveness and efficiency of MoLVERIC, so that it can be adopted by the industry and academy in the future. Furthermore, we intend to conduct a empirical study to analyze the quality of artifacts developed from the MoLIC diagrams, after the inspection with the MoLVERIC.

REFERENCES

[1] Y. Rogers, H. Sharp, J. Preece. Interaction Design: Beyond Human-Computer Interaction, 4th Edition. John Wiley & Sons, 2015.

[2] C. S. De Souza, The Semiotic Engineering of Human-Computer Interaction (Acting with Technology). The MIT Press, 2005.

[3] S. D. J. Barbosa, M. G. Paula, Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering. In Interactive Systems. Design, Specification, and Verification. 10th DSV-IS Workshop, pp. 16–33, 2003.

[4] B.S. Silva, V.C.O. Aureliano, S.D.J. Barbosa. Extreme designing: binding sketching to an interaction model in a streamlined HCI design approach. In Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, pp. 101 – 109, 2006.

[5] B. L. Davies. Grice's cooperative principle: meaning and rationality. Journal of Pragmatics 39, 2308–2331, 2007.

[6] G. H. Travassos, , F. Shull, , J. Carver, Working with UML: A Software Design Process Based on Inspections for the Unified Modeling Language. Advances in Computer, vol. 54, pp. 35 – 98, 2001.

[7] A. Lopes, A. B. Marques, S. D. J. Barbosa and T. Conte. Evaluating HCI Design with Interaction Modeling and Mockups: A Case Study. In Proocedings of International Conference on Enterprise Information Systems, pp. 79-87, 2015.

[8] G. H.Travassos , F. Shull, M. Fredericks, V. Basili. Detecting defects in object-oriented designs: using reading techniques to increase software quality. ACM SIGPLAN Notices, vol. 34, n. 10, pp. 47-56, 1999.

[9] R. Cunha, T. Conte, E. S. Almeida, and J. C. Maldonado, A Set of Inspection Techniqueson Software Product Line Models. In Proocedings of International Conference on Software Engineering and Knowledge Engineering, pp. 657-662, 2012.

[10] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and D. Dixon, Gamification: Using Game Design Elements in Non-Gaming Contexts. In Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems, pp. 2425-2428, 2011.

[11] F. Shull, J. Carver, G. H. Travassos, An empirical methodology for introducing software processes. In 9th ACM SIGSOFT international symposium on Foundations of software engineering, pp. 288 – 296, 2001.

[12] L. Rivero, T. Conte, Improving Usability Inspection Technologies for Web Mockups through Empirical Studies. In Proocedings of International Conference on Software Engineering and Knowledge Engineering, pp. 172-177, 2013.