



Caroline Loppi Guimarães

**Realigning MoLIC to the
Interaction-as-Conversation Metaphor**

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-graduação em
Informática of PUC-Rio in partial fulfillment of the requirements
for the degree of Mestre em Informática.

Advisor: Prof. Simone Diniz Junqueira Barbosa

Rio de Janeiro
July 2023



Caroline Loppi Guimarães

Realigning MoLIC to the Interaction-as-Conversation Metaphor

Dissertation presented to the Programa de Pós-graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee:

Prof. Simone Diniz Junqueira Barbosa

Advisor

Departamento de Informática – PUC-Rio

Prof. Raquel Oliveira Prates

Departamento de Ciência da Computação - UFMG

Prof. Greis Francy Mireya Silva Calpa

Departamento de Informática – PUC-Rio

Rio de Janeiro, July 4th, 2023

All rights reserved.

Caroline Loppi Guimarães

Bachelor in Information Systems in the Applied Informatics Department of the State University of Rio de Janeiro (UNIRIO). Currently works as a software developer at Tecgraf, collaborating with the progress of the geophysical systems.

Bibliographic Data

Caroline Loppi Guimarães

Realigning MoLIC to the Interaction-as-Conversation Metaphor / Caroline Loppi Guimarães; advisor: Simone Diniz Junqueira Barbosa. – 2023.

181 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2023.

Inclui bibliografia

1. Interaction Design – Teses. 2. Diagram – Teses. 3. MoLIC – Teses. 4. Semiotic Engineering – Teses. 5. System Design – Teses. 6. MoLIC. 7. Design de Interação. 8. Ferramenta Epistêmica. 9. Engenharia Semiótica. 10. Interação Humano-Computador. I. Diniz Junqueira Barbosa, Simone. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To my beloved grandmother Zenyra, who will always be in my heart.

Acknowledgments

To PUC-RIO, for the scholarship that allowed me to pursue the master's degree, and to Instituto TECGRAF for all the support.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Loppi G., Caroline; Diniz Junqueira Barbosa, Simone (Advisor). **Realigning MoLIC to the Interaction-as-Conversation Metaphor**. Rio de Janeiro, 2023. 181p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Since the creation of MoLIC, several extensions have been proposed to improve the language. However, many of those proposals have not been thoroughly evaluated and have drifted away from the original interaction-as-conversation metaphor. In this work, we analyzed and consolidated existing proposals and, based on concepts of Linguistics and conversation and discourse analysis, decided whether and how to include the proposed changes. As a result, we created a revised version of the language, MoLIC V4, better aligned with the interaction-as-conversation metaphor and theoretical concepts. To evaluate MoLIC V4, we conducted a survey study to collect feedback from students, practitioners, and researchers on interaction design. We also created a set of examples and brought to the discussion a brief analysis of the epistemic character of MoLIC. We hope that these discussions and examples will serve as a reference when teaching interaction design through MoLIC diagrams.

Keywords

MoLIC; Interaction Design; Epistemic Tool; Semiotic Engineering; Human-Computer Interaction.

Resumo

Loppi G., Caroline; Diniz Junqueira Barbosa, Simone. **Realinhando a MoLIC à Metáfora de Interação como Conversa.** Rio de Janeiro, 2023. 181p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Desde a criação da MoLIC, diversas extensões foram propostas com o objetivo de trazer melhorias à linguagem. Entretanto, muitas dessas propostas não foram minuciosamente avaliadas e se distanciaram da metáfora original de interação como uma conversa. Neste trabalho, analisamos e consolidamos propostas existentes e, com base em conceitos linguísticos e de análise da conversação e do discurso, decidimos se e como incluir as mudanças propostas. Como resultado, nós criamos uma versão revisada da linguagem, a MoLIC V4, elaborada visando um melhor alinhamento com a metáfora de interação como uma conversa e com os conceitos teóricos. Para avaliar a MoLIC V4, nós conduzimos um estudo através de questionário para coletar avaliações de estudantes, profissionais e pesquisadores da área de design de interação. Nós também criamos um conjunto de exemplos e trouxemos para a discussão uma breve análise do caráter epistêmico da MoLIC. Esperamos que essas discussões e exemplos sirvam como material de referência no ensino de design de interação a partir de diagramas MoLIC.

Palavras-chave

MoLIC; Design de Interação; Ferramenta Epistêmica; Engenharia Semiótica; Interação Humano-Computador.

Table of Contents

1	Introduction	1
1.1	Goal, Research Questions, and Research Method	2
1.2	Dissertation Structure	3
2	Theoretical Foundations	5
2.1	Semiotic Engineering	5
2.2	Interaction Design	6
2.3	Modeling Language for Interaction as Conversation	7
2.4	Epistemic Tool	8
3	MoLIC’s Fourth Version (MoLIC V4)	10
3.1	Conversation Opening Point	13
3.2	Conversation Closing Point	13
3.3	Scene	14
3.3.1	Dialogues and Signs	14
3.3.2	List	16
3.3.3	Dialog Grouping Operands	16
3.3.4	Topical Conversation Completion	17
3.3.5	Alert Scene	18
3.4	Turn-Yielding Elements	20
3.4.1	Designer’s (Turn-Yielding) Utterance	20
3.4.2	Designer’s (Turn-Yielding) (Breakdown) Repair Utterance	20
3.4.3	User’s (Turn-Yielding) Utterance	20
3.4.4	Silent Turn-Yielding	21
3.5	User’s Repair Utterance	24
3.6	System Process	25
3.7	Breakdown Prevention Mechanisms	25
3.8	Breakdown Repair Mechanisms	27
3.9	Simultaneous Speeches	28
3.9.1	Simultaneous Speech in the Designer’s Turn	28
3.9.2	Simultaneous Speech in the User’s Turn	29
3.10	Global Topic Shift	31
3.11	If	32
3.12	Context Feature	35
3.13	Effect	36
3.14	Let	37
3.15	External Interlocutor and Fork	37
3.16	Why	38
3.17	Preferred Conversation	40
3.18	Main Scene	41
3.19	Contact Points	42
3.20	When	42
3.21	Elements Removed from MoLIC V4	43
3.21.1	Combined “Forward-Backward” Utterances	44

3.21.2 Stereotyped Scenes	45
4 Usage Examples	47
4.1 User Roles in MoLIC Diagrams	47
4.2 Successive Refinements	47
4.3 Structuring Scenes	49
4.4 Representing Conditions	49
4.5 Order of Elements in Turn-Yielding Utterances	52
4.6 Adding Conditions inside the Topic Shift Element	52
4.7 Representing Independent and Automatic Events	54
4.8 Using the Topical Conversation Completion	55
4.9 Sequencing System Processes	55
4.10 Omitting Scenes and Dialogues	56
4.11 Using Active and Passive Prevention	56
4.12 Mapping the Diagram onto the User Interface	59
4.13 Considerations About MoLIC's Epistemic Character	59
5 Evaluation of MoLIC Proposals	62
5.1 Research Procedure	62
5.2 Analysis Procedure	64
5.3 Respondents	65
5.4 Scene-Related Questions	66
5.4.1 Alert Scene	66
5.4.2 End of a Conversation Related to a Specific Goal	67
5.4.3 Required vs Optional Signs	68
5.4.4 Dialog Format	69
5.4.5 Sign with Multiple Values	70
5.4.6 Selection of a Sign with Multiple Values	71
5.4.7 Interpretation of a Scene Dialog without Grouping Operand	71
5.5 Questions Related to Turn-Yielding Elements	73
5.5.1 Rename - User's Transition Utterance	73
5.5.2 Rename - Designer's Transition Utterance	74
5.5.3 Name Proposals - User's Repair Utterance	74
5.5.4 Rename - Breakdown Transition Repair Utterance	75
5.5.5 Silent Turn-Yielding	75
5.6 Questions Related to Specific Elements	77
5.6.1 Different Ways to Represent the Status of an Ongoing System Process	77
5.6.2 Name Proposals - User Interruption in the Designer's Turn	78
5.6.3 Designer Interruption in the User's Turn	79
5.6.4 Rename - Ubiquitous Access	80
5.6.5 Rename - Conditional in a Conversation	81
5.6.6 Rename - Consequence in a Conversation	82
5.6.7 Let - Context Initial Value	83
5.6.8 Definition of the Conversational Context	84
5.6.9 Rename - Presup	85
5.6.10 Preferred Conversation	85
5.6.11 Main Scene	87
5.6.12 When - Action Associated with an Event	87

5.7	General Questions about Usage Scenarios	89
5.7.1	Usage Scenarios of Conditions, Consequences (Effects), and Utterances in Turn-Yielding Elements	89
5.7.2	Declaration of Interest	90
5.7.3	Clarification	90
5.7.4	Limitations of the Study	90
6	Conclusion	92
6.1	Main Contributions	92
6.2	Future Work	93
	Bibliography	94
A	History of Changes - MoLIC Elements	99
B	Elements Translation to Portuguese	103
C	Informed Consent - Questionnaire (in Portuguese)	105
D	Questionnaire to Evaluate MoLIC Changes in V4	107

List of Figures

Figure 1.1	Research Method	3
Figure 3.1	Simplified Timeline of MoLIC	11
Figure 3.2	MoLIC Diagram Example	11
Figure 3.3	MoLIC diagram snippet representing the opening point	13
Figure 3.4	MoLIC diagram snippet representing the closing point	14
Figure 3.5	MoLIC diagram snippet representing scenes	14
Figure 3.6	Scene with an optional subtopics sign	15
Figure 3.7	MoLIC diagram snippet representing the scene operands	17
Figure 3.8	Topical Conversation Completion (pictured in red)	18
Figure 3.9	Deprecated Element Monologue	19
Figure 3.10	MoLIC diagram snippet representing an alert scene to prevent errors (case 1)	19
Figure 3.11	MoLIC diagram snippet representing Designer's (Turn-Yielding) Utterances	21
Figure 3.12	MoLIC diagram snippet representing User's Turn-Yielding Utterance	22
Figure 3.13	MoLIC diagram snippet representing a Silent Turn-Yielding	22
Figure 3.14	Comparison between Turn-Yielding and Transition	23
Figure 3.15	MoLIC diagram snippet representing a User's Repair Utterance	24
Figure 3.16	MoLIC diagram snippet representing a system process	25
Figure 3.17	Breakdown Prevention Mechanisms: Active and Passive Prevention	26
Figure 3.18	Breakdown Prevention Mechanisms: Supported Prevention	27
Figure 3.19	Breakdown Repair Mechanisms	28
Figure 3.20	Simultaneous Speeches Variations	29
Figure 3.21	MoLIC diagram snippet representing a Simultaneous Speech in the Designer's Turn	29
Figure 3.22	MoLIC diagram snippet representing a Simultaneous Speech in User's Turn	30
Figure 3.23	Deprecated Element due to Redundancy	31
Figure 3.24	Topic Shift	32
Figure 3.25	MoLIC diagram snippet representing conditions in turn-yielding elements	33
Figure 3.26	MoLIC diagram snippet representing conditions within a scene utterance at sign level and at dialog level	33
Figure 3.27	MoLIC diagram snippet representing an effect	36
Figure 3.28	MoLIC diagram snippet representing the use of the let element	38
Figure 3.29	MoLIC diagram snippet representing External Interlocutor and Fork	39
Figure 3.30	MoLIC diagram snippet representing a Why	39
Figure 3.31	Element Why used in a diagram	40

Figure 3.32 MoLIC diagram snippet representing a Preferred Conversation	40
Figure 3.33 MoLIC diagram snippet representing the Main Scene	41
Figure 3.34 MoLIC diagram snippet representing the concept of Contact Points	43
Figure 3.35 MoLIC diagram snippet representing the usage of the element when	44
Figure 3.36 Combined Utterances - Scenario with Ambiguity	45
Figure 3.37 Separated Turn-Yielding Utterances - Scenario without Ambiguity	46
Figure 4.1 Banking example	48
Figure 4.2 Scene Structuring Differences	50
Figure 4.3 Representing Conditions - Scenarios a to d	50
Figure 4.4 Representing Conditions - Scenarios e to g	52
Figure 4.5 Elements order	53
Figure 4.6 Topic shift With an embedded condition	53
Figure 4.7 Topic shift Without an embedded condition	54
Figure 4.8 Representing Independent and Automatic Events	55
Figure 4.9 Feedback with Topical Conversation Completion <i>versus</i> Regular Scene	56
Figure 4.10 MoLIC Diagram Snippet representing sequential system processes	57
Figure 4.11 Passive Prevention - MoLIC and Mock-up Example	58
Figure 4.12 Active Prevention - MoLIC and Mock-up Example	58
Figure 4.13 Active Prevention - MoLIC and Mock-up Example	59
Figure 5.1 Alert Scene	66
Figure 5.2 Scene without topic	68
Figure 5.3 Topical Conversation Completion	68
Figure 5.4 Marking only the Optional Signs	69
Figure 5.5 Emitters next to the dialogue's signs and the list element	70
Figure 5.6 Choose among a list of signs	72
Figure 5.7 Implicit form of the grouping operand AND	73
Figure 5.8 User's (Turn-Yielding) Utterance	73
Figure 5.9 Designer's (Turn-Yielding) Utterance	74
Figure 5.10 User's Repair Utterance	74
Figure 5.11 Designer's (Turn-Yielding) (Breakdown) Repair Utterance	75
Figure 5.12 Silent Turn Yielding	75
Figure 5.13 Simultaneous Speech in the Designer's Turn	78
Figure 5.14 Simultaneous Speech in the User's Turn	80
Figure 5.15 Global Topic Shift	81
Figure 5.16 If	82
Figure 5.17 Effect	83
Figure 5.18 Let	83
Figure 5.19 Why in a Dialogue	86
Figure 5.20 Why in a Turn-Yielding Element	86
Figure 5.21 Preferred Conversation	87
Figure 5.22 Main Scene	88

List of Tables

Table A.1	Elements' versions	100
Table B.1	Elements' translation to Portuguese	104

List of Abbreviations

IHC – Interação Humano-Computador

MoLIC – Modeling Language for Interaction as Conversation

1

Introduction

Two main areas in the software industry focus on software quality: Software Engineering and Human-Computer Interaction (HCI). While Software Engineering focuses more on the architecture, development, implementation, and maintenance of systems, HCI targets the quality perceived by the user (Barbosa et al., 2021).

Models allow designers to represent and analyze parts of a scenario, in a variety of perspectives and levels of details (Barbosa et al., 2021). Designers can test their decisions and make refinements by using their models as frameworks for decision making (Hoover et al., 1991).

In HCI, several design models are available to work at different aspects of a system, such as domain, application, task, user-system interaction, and user interface (Silveira et al., 2004). In this work, we focus on an interaction model proposed by de Paula and Barbosa (2003): the Modeling Language for Interaction as Conversation (MoLIC). With this language, the authors intended to fill a gap between task models and user interface models, since task models were too abstract and the user interface models included aspects that could be very specific to a platform or device (de Paula and Barbosa, 2003).

MoLIC allows designers to represent user-system interactions as messages exchanged between users and the user interface (Barbosa et al., 2021), where the user interface is the *designer's deputy*, which represents the designer's intentions crystallized at the user interface (de Paula and Barbosa, 2003). MoLIC was conceived within the Semiotic Engineering theory, which perceives the user-system interaction as a conversation between the user and the designer, mediated by the system (de Souza, 2005b). MoLIC was created to help designers reflect on the problem and solution being conceived (Barbosa et al., 2021).

As identified by de Carvalho et al. (2019), MoLIC is widely used and studied in Brazil and internationally. It is currently in its second edition (henceforth MoLIC V2), and several extensions have been proposed to improve the language. However, those proposals have not been thoroughly evaluated and have drifted away from the original interaction-as-conversation metaphor.

In this work, we analyzed and consolidated two concurrent proposals: MoLIC's third version proposed by Araujo and Barbosa (2008) and the work of Pessanha and Barbosa (2018). We also analyzed the HCI textbook written by Barbosa and Silva (2010), to ensure all MoLIC diagram elements presented there were considered when consolidating the new language version. Using MoLIC V2 as our baseline, we analyzed these proposals to realign the MoLIC elements with the interaction-as-conversation metaphor. To do this, we investigated concepts of Linguistics and conversational and discourse analysis.

1.1

Goal, Research Questions, and Research Method

The main goal of this work is to realign MoLIC with its origins: the interaction-as-conversation metaphor and semiotic engineering. We also aimed to expand the expressiveness of the language by consolidating existing unresolved proposals while keeping its simplicity. To accomplish this goal, we proposed MoLIC V4, a version of MoLIC resulting from the consolidation of the works by Silva and Barbosa (2005), Araujo and Barbosa (2008), and Pessanha and Barbosa (2018), a study of relevant linguistic concepts, and a questionnaire study.

In chapter 3 we present a revised version of the language and explain how each element was changed, describing how it was grounded in the linguistic concepts. MoLIC is currently in its second version, and, because the third version, proposed by Araujo and Barbosa (2008), was not incorporated into the language due to the lack of evaluation of their proposals, we consolidated in this work the fourth version of MoLIC, or MoLIC V4.

Thus, the research questions we expect to answer in this work are:

- RQ1: How to realign MoLIC with the interaction-as-conversation metaphor?
- RQ2: How to expand the language without increasing its complexity?
- RQ3: How do users of previous MoLIC versions perceive the changes?

To evaluate the interaction design community's perception of MoLIC V4, we conducted a survey to collect feedback from students, practitioners, and researchers who had at least a basic knowledge of MoLIC.

We also created a set of examples while bringing to the discussion a brief analysis of the epistemic character of MoLIC. We hope these discussions and examples will serve as a reference when teaching interaction design through MoLIC diagrams.

Figure 1.1 illustrates our research method, starting with the study of the proposals from both Araujo and Barbosa (2008) and Pessanha and Barbosa (2018) while in parallel studying concepts of Linguistics, conversation analysis, and discourse analysis that may apply to our work. Next, we proposed MoLIC V4, realigning it with the interaction-as-conversation metaphor and discussing the rationale for the changes. Based on the new version, we created examples to illustrate the usage of MoLIC V4, emphasizing the proposed changes and usage scenarios that are less often used or misunderstood (as noticed in informal observations). In the end, we conducted a questionnaire study to evaluate the perceptions of MoLIC users of the consolidated language version, MoLIC V4.

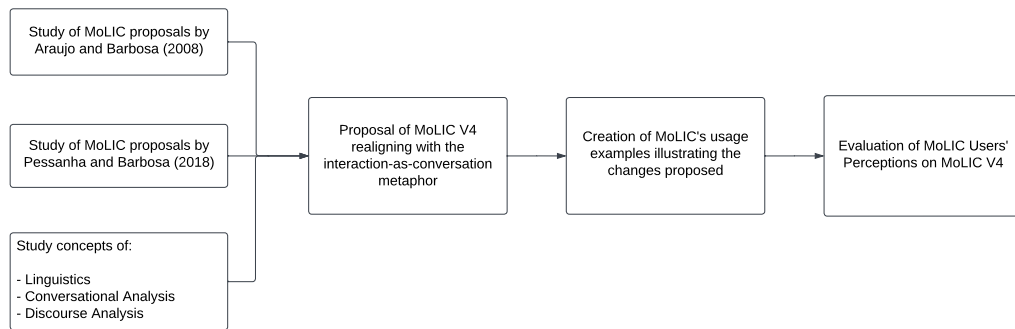


Figure 1.1: Research Method

Our goal with the evaluation was twofold: (i) regarding elements in the language for which there was more than one proposal equally grounded in the literature, we could consider the respondents' opinions and preferences; and (ii) regarding elements that respondents disliked, but that should change to achieve the goal of realigning MoLIC with the interaction-as-conversation metaphor, we gathered valuable information that may inform future instructional material to help reduce misconceptions and ease their effort when using MoLIC V4.

1.2 Dissertation Structure

This document is structured as follows: chapter 2 describes the theoretical foundations of this work, comprising Semiotic Engineering, Interaction Design, and the Modeling Language for Interaction as Conversation (MoLIC). Chapter 3 describes MoLIC V4, discussing each element that constitutes this new language version. In chapter 4, we present examples to address different usage scenarios. chapter 5 details the questionnaire study to evaluate MoLIC V4 proposals, describing the research procedure and discussing the results. Finally,

in chapter 6, we present the main contributions of this work and point to future work.

2

Theoretical Foundations

In this chapter, we present the foundations for our work. In section 2.1, we briefly present the Semiotic Engineering theory and how MoLIC is related to it. In section 2.2, we introduce an interaction design process, narrowing it to the application in the software industry. In section 2.3, we present MoLIC and its evolution over the years.

2.1

Semiotic Engineering

Semiotic Engineering is an HCI theory based on the idea that the interaction between humans and computers is a conversation between humans, but mediated by a computational system (de Souza, 2005b,a). In this theory, the system can be viewed as a set of designer's intentions crystallized (de Souza, 2005b). The system's user interface is then a medium where two levels of communication can occur: user-system, meaning the direct interaction of the user with the system; and user-designer, which occurs through the system, and communicates what the designer understood from the users themselves, their needs, and expectations, as well as the designer's vision and design intent about how users can or should use the system (Barbosa et al., 2021). Based on these understandings, the designer elaborates a message to be communicated to the user; this message is transformed into words, behaviors, concepts, and other communication signs, which are codified into a user interface, allowing the metacommunication to occur at interaction time (Barbosa et al., 2021).

It is during design that the message, composed of signs, is elaborated. The designer aims to ensure communication quality as the system is conceived and implemented. At interaction time, the designer will not be available to help the user understand what is being communicated, thus, configuring the designer's intentions represented by the system as a one-shot message from the designer to the user (Barbosa et al., 2021; de Souza, 2005b).

Regarding communication quality, the anticipation of breakdowns and alternative interactions is a major aspect to consider during the design stages. Therefore, de Paula and Barbosa (2003) proposed an interaction modeling language, MoLIC, to help designers reflect on the message they will communicate

to users and with which users will communicate with the user interface (viewed by semiotic engineering as the *designer's deputy*) (Barbosa et al., 2021). MoLIC allows the designers to better understand the conversation flow they are creating by giving a global view of the system (Barbosa et al., 2021), its mapped communicative breakdowns, and alternative conversations being defined.

2.2

Interaction Design

The goal of interaction design is to support people in their work, communication, and interactions through the design of interactive products (Preece et al., 2002). During the design process, it is imperative to consider the user's perspective in order to provide a high-quality computational artifact; otherwise, the result may be a solution that may functionally work but is difficult or unpleasant to use (Preece et al., 2002). As a field, interaction design has the intent to include usability (and other quality criteria) in its process, to then generate products that are straightforward and pleasant to be used (Preece et al., 2002).

Interaction design is a vast field, which includes any kind of interactive product. In this work, we focus on interaction models, artifacts that represent all possible user-system interactions, with the purpose of planning the actions that can be executed during interaction time by both users and the system – *i.e.*, conversations between users and the designer's deputy (de Paula and Barbosa, 2003). According to de Paula and Barbosa (2003), interaction models proposed before MoLIC represented interaction at a low abstraction level, including user interface elements and interaction mechanisms dependent on the platform, such as clicks, typing, etc. This continues to be true after nearly two decades (Faltaous et al., 2021). This practice tends to affect the design results by including too many details that should be considered later in the process (Barbosa and de Paula, 2003a), as they distract designers from higher-level issues that impact the perceived quality of the computational artifact. de Paula and Barbosa (2003) argue that, first, interaction models should allow the designer to avoid inconsistencies in the interactions in a more platform-independent way, closer to the users' goals and tasks than to the system's input and output capabilities.

In the next section, we explain MoLIC, an interaction modeling language created to fill this gap.

2.3

Modeling Language for Interaction as Conversation

The Modeling Language for Interaction as Conversation (MoLIC) was proposed by de Paula and Barbosa (2003) to fill a gap between platform-dependent user interface models and task models, which can be very abstract (de Paula and Barbosa, 2003). MoLIC aims to be an epistemic tool to help the designer gain an understanding of both the problem being solved and the solution being designed, helping designers to plan the range of interaction possibilities through the reflection on the users' goals and strategies that the interactive solution being conceived should support (de Paula and Barbosa, 2003; Barbosa et al., 2021).

Two years after the original version of the language (henceforth MoLIC V1), Silva and Barbosa (2005) revised the language and proposed a second version (MoLIC V2), improving the semantics of the language elements; they also focused on allowing a better detailing of the interaction and explored a few asynchronous multiuser scenarios. In the next year, Silva and Barbosa (2007) created a practical guide to explain MoLIC's foundations and present a set of scenarios to help designers create MoLIC diagrams.

One year later, Araujo and Barbosa (2008) conducted research with a group of designers to uncover some of the language limitations. It resulted in a set of proposals to improve MoLIC V2. They proposed a third version, which was not incorporated into the language because the changes proposed in their work were not empirically evaluated. In this dissertation, we took MoLIC V2 as a baseline and revisited Araujo and Barbosa's work to evaluate their suggestions with MoLIC users.

Ten years after the work of Araujo and Barbosa (2008), Pessanha and Barbosa (2018) proposed several changes to MoLIC after reverse engineering a set of selected systems. In their work, Pessanha and Barbosa (2018) identified situations where MoLIC was not able to clearly and succinctly represent the interaction scenarios existing in the analyzed systems. They focused on expanding the language's expressiveness while keeping its simplicity (Pessanha and Barbosa, 2018). However, like Araujo and Barbosa (2008), Pessanha and Barbosa (2018) did not empirically evaluate their proposals.

MoLIC diagrams represent the potential user-system interactions as a set of conversations that the user may establish to achieve their objectives with the application (de Paula and Barbosa, 2003; Barbosa et al., 2021). From a semiotic engineering perspective, the system is the materialization of the designer's intention from what they understood of the user's goals, needs, values, and expectations (de Souza, 2005b). This intention is materialized

at the user interface as the *designer's deputy* (de Paula and Barbosa, 2003), meaning that the system is a representative of the designer (Prates et al., 2000).

In chapter 3, we describe in detail the elements of MoLIC's interaction diagram and their usage.

2.4

Epistemic Tool

Epistemic tools help their user comprehend the problem and provide alternative solutions to it; these tools are not expected to directly provide an answer to a problem (de Souza and Leitão, 2009). In Semiotic Engineering, examples of epistemic tools are the Semiotic Inspection Method (SIM) and the Communicability Evaluation Method (CEM) (de Souza and Leitão, 2009).

When it comes to the understanding of MoLIC as an epistemic tool, de Souza and Leitão (2009) state that the epistemic use of MoLIC diagrams is expected to help the designers communicate the “logic of conversations” that may occur between their deputy and users. Araujo and Barbosa (2008) clarifies that although MoLIC was proposed with the aim of being an epistemic tool, its epistemic features had not been explicitly explored.

Araujo and Barbosa (2008) addressed this lack of analysis by proposing a set of questions to be used alongside the diagram to motivate the designer's reflection on both the underlying problem and the solution being designed. In their study, these questions were expected to help the design and redesign activity and the understanding of solutions presented in MoLIC artifacts (Araujo and Barbosa, 2008) by provoking a reflection that results in a better understanding of the consequences of the decisions made during the design process and represented in MoLIC diagram(s).

Araujo and Barbosa (2008) proposed 25 questions that a designer could use to reflect on a MoLIC solution created either by themselves or others at any moment of the interaction modeling process. These questions were categorized into 11 groups, indicating how MoLIC helps to think about a problem and its possible solutions, thus, how the epistemic character of MoLIC unveils in the design process. The groups of reflective questions are the user's goals, how to start the conversations about them, which dialogues should be exchanged to achieve each goal; conversation (with the application) openings and closings; topics and subtopics of the conversation; breakdowns in the communication; turn exchanges between user and designer's deputy, and vice-versa; conversations between users playing the same or different roles; and finally, which conversations may be exchanged with external systems. The

related questions are presented and detailed in (Araujo and Barbosa, 2008, Chapter 4).

3

MoLIC's Fourth Version (MoLIC V4)

MoLIC diagrams should allow representing every possible interaction scenario (Barbosa and de Paula, 2003a). During the twenty years since the publishing of the original language version, several works have proposed changes to it. In addition, some researchers aimed to extend MoLIC to other application types, such as Collaborative Systems (de Souza et al., 2015), AI-based systems (Ferreira et al., 2019), and Conversational Agents (Fernandes et al., 2021).

In this research, we gathered exclusively the publications that proposed changes to MoLIC while maintaining its original scope: deterministic, single-user interactive systems without synchronous interactions and communication. Considering this criterion, we selected three works: the MoLIC V2, from Silva and Barbosa (2005), the dissertation published by Araujo and Barbosa (2008) (who proposed a third version, but several elements of which have never been adopted), and the study of Pessanha and Barbosa (2018) (which was also not publicized or adopted). All these works aimed to evolve MoLIC while keeping its original application scope.

We also inspected the HCI textbook by Barbosa and Silva (2010), which has as authors two researchers responsible for the creation and evolution of the language. This inspection aimed to ensure all MoLIC elements presented in the book were included in our study. Figure 3.1 places the selected works in a simplified MoLIC timeline.

This chapter presents a MoLIC diagram example already expressed in MoLIC V4 and analyzes each element of the new version in its own section. In Appendix A, we detail how each element changed over time. We evaluated MoLIC users' perceptions of the changes in a questionnaire, considering the original elements and the concurrent proposals of names, representations, and even new elements). This evaluation study is described in chapter 5.

In Figure 3.2, we present a simplified example of a MoLIC interaction diagram. This example does not contain all elements from the MoLIC language. Each element is exemplified in its own section, and in chapter 4 we present additional diagrams when discussing various usage scenarios.

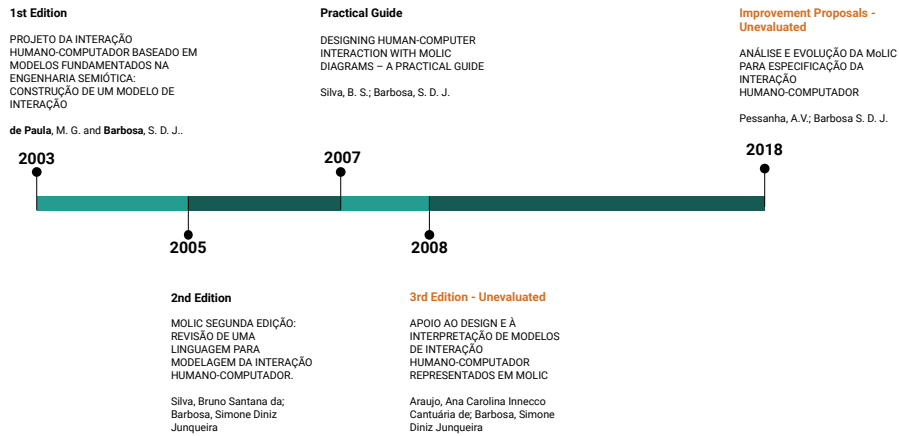


Figure 3.1: Simplified Timeline of MoLIC

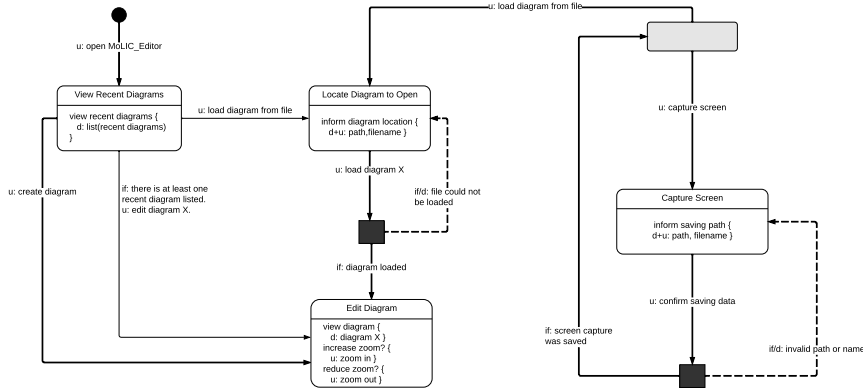


Figure 3.2: MoLIC Diagram Example

Figure 3.2 shows a MoLIC diagram modeling the possible user interactions with a diagram editor application. The circle filled with black color is a conversation opening point and is used to represent the start of the conversation, meaning, in practice, where the user enters the system. The rounded rectangles are the *scenes*, which comprise two compartments: their topic, in the first compartment, in which the designer presents the goal a user can achieve by holding the dialogues described within square brackets in their second compartment. When the user wants to advance in the conversation topic or switch to a new topic, they have to emit a turn-yielding utterance (which is represented by the idea of following the arrow in the diagram). These arrows with text prefixed by **u:** are the user's turn-yielding utterances. The arrows marked with a **d:** represent the moment when the designer's deputy is "talking" and it is now their turn to decide the conversation flow. The designer's deputy's utterances (henceforth *designer's utterances*) are perceived by the

user as the system behavior that surfaces at the user interface (as predefined by the designer).

Analyzing the diagram depicted in Figure 3.2, immediately after the user starts the conversation with the system, they can view the recent diagrams saved in the application (scene **View Recent Diagrams**). At this moment, the user can choose between three possible conversations: create a diagram (utterance u: **create diagram**); load a diagram from a file (utterance u: **load diagram from file**); or edit one of the listed diagrams (utterance u: **edit diagram X**). This latter option is only possible if at least one recent diagram is listed (as stated by the **if** element in the respective user's turn-yielding utterance). If the user decides to load a diagram from a file, they will be asked (in the **Locate Diagram to Open** scene) to inform the diagram location (signs d+u: **path, filename** in the **[inform diagram location]** dialogue). When they are ready to move forward with this conversation and ask the system to load the diagram (utterance u: **load diagram X**), they will give the conversation turn to the system, which will process their request and may lead to one of two different results: one successful, having the effect of loading the diagram (leading the conversation to the scene **Edit Diagram**), and another unsuccessful, meaning a communication breakdown occurred (leading the conversation back to the scene **Locate Diagram to Open**). In this case, the designer's deputy will inform the user that a failure occurred when loading the diagram (designer's utterance d: **file could not be loaded**, combined with the identical condition defined by the element **if**).

On the upper right of the diagram, we have the Topic Shift element, a gray-filled rounded rectangle used to represent the start of conversations that can be held at any moment. In the example, it means the user can capture the screen at any time. Again, after confirming their request in the scene **Capture Screen**, the designer's deputy (the designer's "spokesperson" as materialized at the user interface) will be in charge to process the request (black square in the diagram) and lead the conversation to the corresponding result.

As an example of the reflection motivated by the representation, after creating this diagram we noticed a problem in the depicted solution: we had considered only a subset of the usage situations, namely, that the user has already created and saved a diagram with the application. Instead of always starting the conversation at the **View Recent Diagrams** scene, if the user has never saved a diagram (*i.e.*, there is no "recent diagram" to show), the conversation might start at the **Edit Diagram** scene, for the user to start creating their diagram (and, from there, be able to open load a diagram from a file, if they so choose to do). This solution would make the interaction more

efficient when the user wants to create a new diagram (as is usually the case in the first use of a system) and equally efficient when the user wants to load a diagram from a file. We decided to leave the example with that inefficiency to illustrate the epistemic nature of MoLIC and how it involves successful refinements, as does any design artifact.

In the next sections, we describe and analyze each interaction diagram element.

3.1

Conversation Opening Point

Drawn as a black-filled circle, this represents the entry point of an application, which leads the user to the first scene (Figure 3.3). It is possible to have more than one opening point when an application can be started in different states or scenes. For instance, a second opening point could lead directly to the scene **Edit Diagram** after the user opens a file associated with the diagram editor in their operating system.

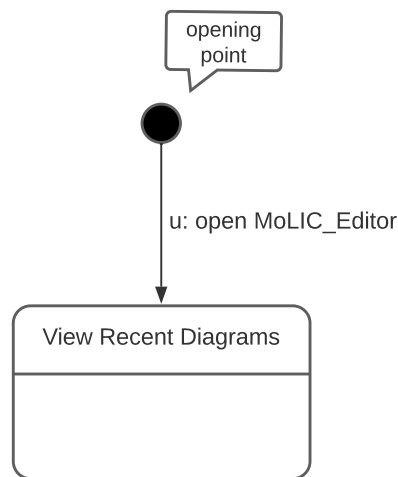


Figure 3.3: MoLIC diagram snippet representing the opening point

3.2

Conversation Closing Point

Drawn as a black circle centered inside a white circle (Figure 3.4), it represents the exit point of an application, indicating where the conversation ends (*i.e.*, the user exits the system).

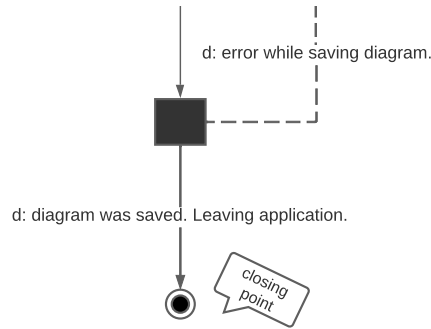


Figure 3.4: MoLIC diagram snippet representing the closing point

3.3

Scene

A *scene* represents a topical conversation the user may have with the designer's deputy to advance a certain goal and possibly exchange data. The scene is structured in two containers: the upper one has the **topic**: a phrase from the designer's viewpoint about what the user is able to (or should) achieve at that moment (de Paula and Barbosa, 2003; Barbosa et al., 2021). In the second container, below the scene topic, there are subtopics, which consist of the *dialogue(s)* between the user and system that may or should be held in order to achieve the goal described in the scene topic (Figure 3.5). In a scene, it is the user's turn to decide how the conversation will proceed (within constraints predefined by the designer).

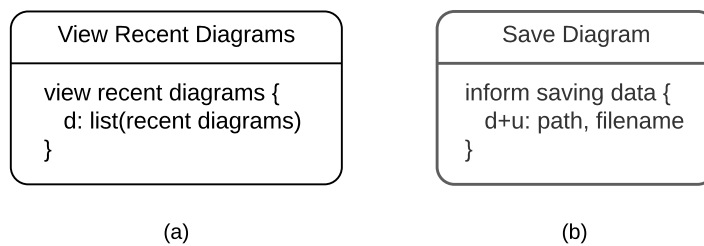


Figure 3.5: MoLIC diagram snippet representing scenes

3.3.1

Dialogues and Signs

Each dialogue in a scene can be elaborated in a minimal form containing only the subtopic (*e.g.*, **inform saving data**) or in a more detailed form that allows the designer to make clear the sign(s) of the dialog(s), as well as whether they are emitted only by the designer, or by both the designer and the user (*e.g.*,

inform saving data {d+u: path, filename})(de Paula and Barbosa, 2003; Silva and Barbosa, 2007; Barbosa et al., 2021).

When the sign is emitted by the designer only (prefixed by **d:**) (*i.e.*, it represents a designer's utterance), the goal is achieved by the user simply "reading" the sign. In other words, the user does not need to and cannot add any information to the conversation related to that sign. In the scene where the user can view all the diagrams available, the dialog can be detailed in the scene as **view recent diagrams** {d: list(diagrams)}, as pictured in Figure 3.5a. When the user needs to provide a specific piece of information to proceed in that conversation – for instance, when they want to save a file but have to inform the path and filename to do so –, the designer first prompts the user and then the user provides the information, *i.e.*, both designer and user emit the sign(s) (prefixed by **d+u:**). The dialog can then be detailed in the scene as depicted in Figure 3.5b: **inform saving data** {d+u: path, filename}.

From a Linguistics perspective, the **d+u:** prefix of a sign can be understood as an answer-response pair (an adjacency pair), where the response is expected to fit the answer (Sacks, 2010; Hutchby and Wooffitt, 1988).

As stated before, in the detailed form of the dialogues, the designer is able to add the signs involved in that conversation (Barbosa et al., 2021). In the example **inform saving data** {d+u: path, filename}, *path* and *filename* are the signs involved in the conversation *inform saving data*.

In the case of signs involving both the user and the designer, the designer may define whether the information about each sign is required or not. If a sign is suffixed by a question mark, it is an **optional sign**, meaning that it is up to the user to decide whether to inform or not the corresponding data without impeding the achievement of the corresponding interaction goal (Figure 3.6). By contrast, if the sign has no mark attached, it must be informed to move the conversation forward (*i.e.*, it is a **mandatory sign**).

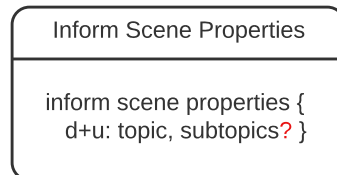


Figure 3.6: Scene with an optional subtopics sign

Changelog - Optional vs Required Signs In MoLIC V2, the required signs were marked with an asterisk; in this revision, we accepted the suggestion of Pessanha and Barbosa (2018) to specify only the optional signs with a question mark, *i.e.*, if it is not specified as optional, the sign is required.

3.3.2

List

In MoLIC, inside a scene dialog, when the designer wants to ask something for the user to respond, we use the **d+u:** prefix. In some cases, the designer will provide a range of predefined responses (previously marked in MoLIC with both **list** and **set**, for ordered and unordered items, respectively), which is now represented in the dialog as **d+u: list(sign)**. One may also inform whether the response options are ordered (either ascending or descending, as follows: **d+u: list(sign, asc)**). If no order is informed, then the order of the items has no need to be preserved. If the designer intends to be more specific, they can include the number of items that the user should select through the element **choose** (originally proposed as **pick**). An example would be:

d+u: choose (1~4, list(diagram-stencil, asc))

This means that the designer expects the user to choose from 1 to 4 diagram stencils out of a list of options presented in ascending order at the user interface.

Changelog - List In MoLIC V2, there were two elements: **list** (or sequence) and **set**. In this work, we merged both concepts and use the element *list* to represent both ordered and unordered elements. We suggested the replacement with the term *response options* that is, from a Linguistics perspective, the set of possible answers in an adjacency pair (marked by **d+u:**). (Eggins and of Sydney, 1990; Stokoe et al., 2020). This term was later identified as inadequate for replacing the term **list**, as explained in subsection 5.6.5. We also included Pessanha and Barbosa (2018)'s suggestion of cardinality, adding the prefix **choose** (originally proposed as *pick*) and using a tilde as the separator for the cardinality lower and upper bounds.

3.3.3

Dialog Grouping Operands

The following operands can be used to group dialogues or signs within a dialogue (Figure 3.7).

- **AND** This operand indicates that the user has to hold all the dialogues within the AND group (Silva and Barbosa, 2005; Barbosa et al., 2021). This operand is considered the default structure and can be omitted from a scene.
- **SEQ** A dialogue may be composed of a group of dialogues that must be held in a specific order (Silva and Barbosa, 2005; Barbosa et al., 2021). In this case, the operand SEQ should be used to represent a sequence of dialogues that should take place in order to achieve the intended goal.

- **OR** In a scene, some dialogues may be optional, for instance, when the user has to share their contact information, and they are able to decide between e-mail, phone, or neither. In this case, the dialogue can be structured with the OR operand (Silva and Barbosa, 2005; Barbosa et al., 2021). If the designer wants to force at least one option, they can add this restriction, as shown in Figure 3.7 (OR (must inform at least one) {...}).
- **XOR** When a user has to hold only one of a set of dialogues or assign values to one of a set of signs, we use the operand XOR (Silva and Barbosa, 2005; Barbosa et al., 2021). For instance, a user may set a filter based on a specific date interval or for the last n days. Only one of the dialogues can occur.

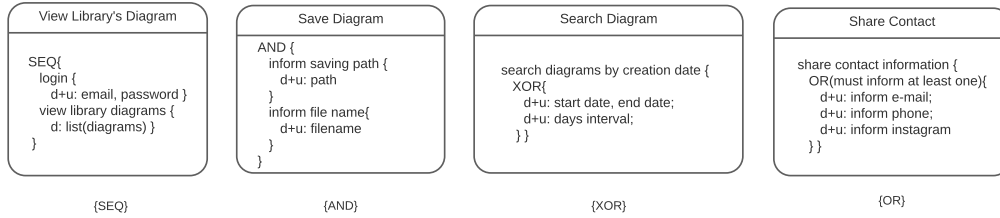


Figure 3.7: MoLIC diagram snippet representing the scene operands

3.3.4

Topical Conversation Completion

When the conversation about a specific topic is finished and there is nothing more the user can do about their goal (Silva and Barbosa, 2007), a designer's turn-yielding utterance leads to a **topical conversation completion** (a rounded rectangle with a horizontal line inside) as pictured in Figure 3.8. It is important to note that this is a “local end”, meaning the user is not leaving the application (Silva and Barbosa, 2007), but the conversation regarding that topic is over (successfully or not). In the example of Figure 3.8, adapted from (Silva and Barbosa, 2007), if the user requests the creation of an account, they have to wait until the administrator asynchronously contacts them. Therefore, there is nothing more the user can do to achieve this goal within the system.

Changelog - Topical Conversation Completion and Monologue

In MoLIC V2, the monologue element (Figure Figure 3.9) contained the message intended to be shown in case of the end of the conversation about a certain topic. Since the same message once shown in the monologue can be represented by the designer's turn-yielding utterance leading to the **topical**

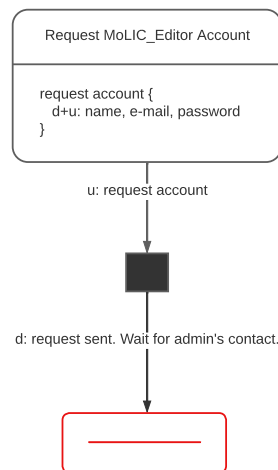


Figure 3.8: Topical Conversation Completion (pictured in red)

conversation completion, the monologue element was deprecated, as suggested by Araujo and Barbosa (2008), and replaced with the **topical conversation completion** (originally proposed with the name *empty scene*). To align this new element with the interaction-as-conversation metaphor, we analyzed linguistic concepts and found the notion of *utterance completion*, which means that the utterance is concluded before the turn is taken by the other speaker (Hutchby and Wooffitt, 1988). Thus, this element should be used in cases where the designer concludes their utterances about a topic and gives the turn back to the user through a designer's turn-yielding utterance. At this point, the user can decide on the next topic of conversation.

3.3.5 Alert Scene

Pictured as a scene with a dashed outline, this element is used to represent a situation where the designer foresees a potential cause of communication breakdown (Figure 3.10). For instance, when the user wants to save a file overwriting an existing one, the designer cannot anticipate whether the user wants to overwrite the file or not; it might be catastrophic to seamlessly overwrite it and annoying to prevent the user from doing so if they so want. In either case, the user can be alerted about the existing file in an alert scene, allowing them to decide whether they want the file to be overwritten or not, preventing errors and unexpected outcomes. This use of alert scenes is known as *supported prevention* and is described in section 3.7

Changelog - Alert Scene This element was extracted from the HCI

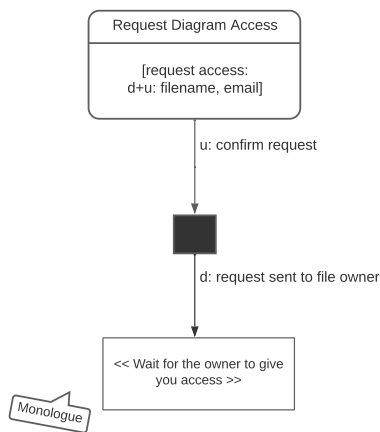


Figure 3.9: Deprecated Element Monologue

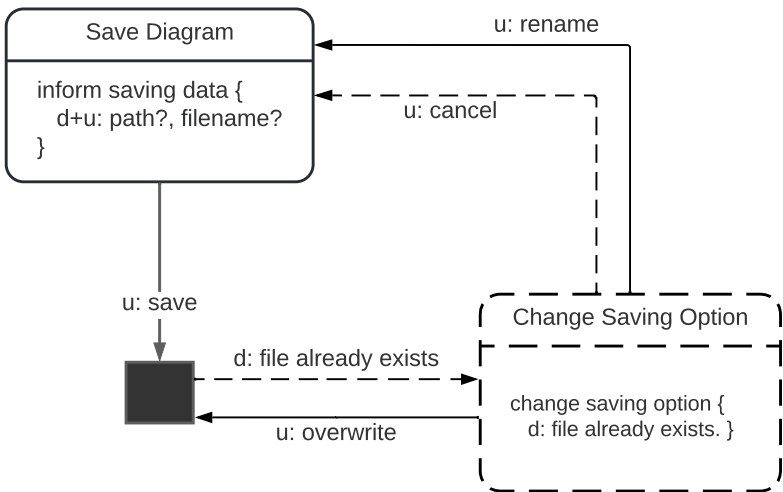


Figure 3.10: MoLIC diagram snippet representing an alert scene to prevent errors (case 1)

textbook by Barbosa and Silva (2010).

3.4

Turn-Yielding Elements

Pictured as an arrow with a label, the turn-yielding element represents the exchange of conversational turns between the user and the designer. It may occur between two scenes, a scene, and a system process, or two system processes. The turn-yielding element may contain an associated text indicating the sender (either user or designer's deputy, depending on the case) and the content of the utterance to be transmitted. In the following subsections, we describe the five types of turn-yielding elements:

3.4.1

Designer's (Turn-Yielding) Utterance

In MoLIC V4, this represents the designer's communication about the results of a system process, leading the user to a scene where they can continue the conversation. It is pictured as a solid arrow containing a label that starts with **d:** (*e.g.*, **d: diagram was saved** in Figure 3.11). As the arrow represents the turn yielding, this element can also be called simply *Designer's Utterance*. It can also be used in association with the element **when**, when responding to events, as presented in section 3.20. When there is a breakdown, the designer's turn-yielding utterance is dashed and is known as a *Designer's Repair Utterance*, described next.

3.4.2

Designer's (Turn-Yielding) (Breakdown) Repair Utterance

This element represents the designer's communication about an unexpected result of a system process or a communication breakdown, *i.e.*, a problem during system processing (*e.g.*, **d: invalid path or filename** in Figure 3.11). The designer anticipates this situation as a breakdown – a deviation from the user's original intention – that could occur during the user-system interaction. It is pictured as a dashed arrow with the corresponding utterance (*i.e.*, **d: what went wrong (and why, and what to do about it)**). As the arrow indicates the turn yielding and the dashed pattern indicates a breakdown occurred, this element can be called simply *Designer's Repair Utterance*.

3.4.3

User's (Turn-Yielding) Utterance

This element represents an opportunity for the user to advance or switch the conversation topic. It is pictured as a solid arrow containing a label that starts with **u:** (*e.g.*, **u: edit diagram X** in Figure 3.12). Through this turn-yielding

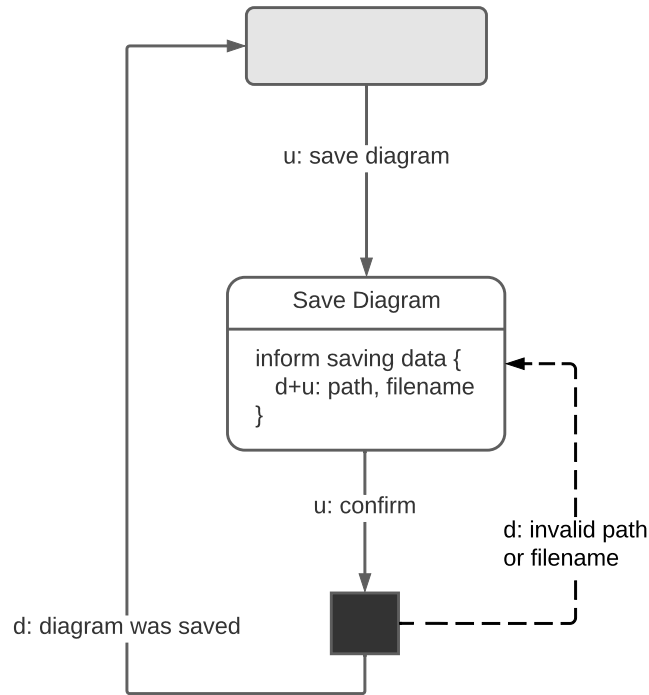


Figure 3.11: MoLIC diagram snippet representing Designer's (Turn-Yielding) Utterances

utterance, the user gives the turn back to the system, which will then process the request. Similar to the previous cases, as the arrow represents the turn yielding, this element can also be called simply *User's Utterance*.

3.4.4 Silent Turn-Yielding

This element is pictured as an arrow with the label having only a condition indicated by an *If* element. This element is expected to be used in cases where the designer's turn yielding does not require an explicit utterance by the designer. For instance, the designer may decide that, after a successful system process (*i.e.*, the result is what the user expected), no utterance about the result is needed and, instead, they will only present the next scene topic to the user. In these cases, MoLIC provides the *Silent Turn-Yielding* element (*e.g.*, if: filename and path are valid in Figure 3.13).

Changelog - Utterances Renamed In MoLIC V4, to be congruent with the interaction-as-conversation metaphor, we removed the word "transition" from these elements, based on the work of Hutchby and Wooffitt (1988). According to them, transitions are seen as something that occurs *between* turns, whereas this element occurs *during* a turn: from a scene, the **u:** utterance is still emitted during the user's turn; likewise, from a system process, the **d:** ut-

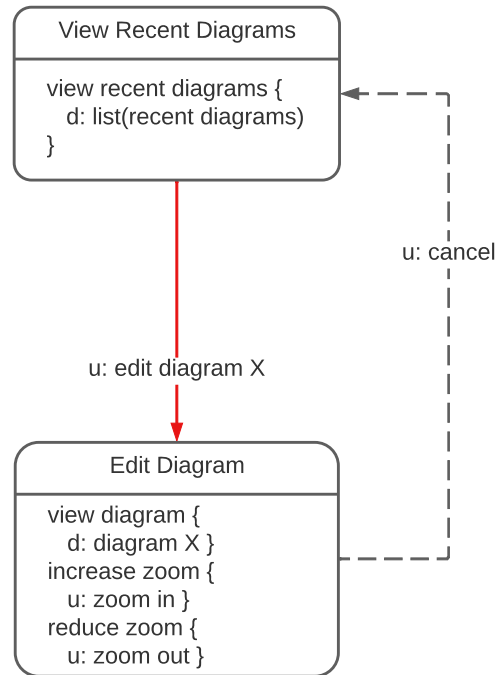


Figure 3.12: MoLIC diagram snippet representing User's Turn-Yielding Utterance

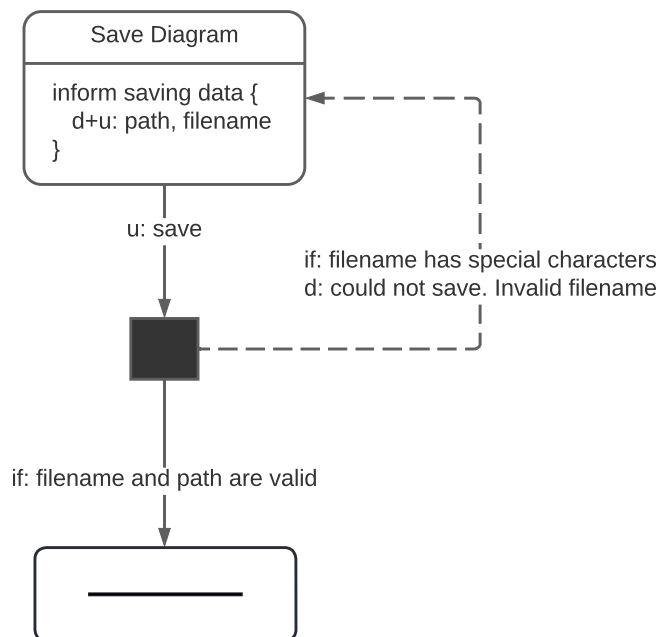


Figure 3.13: MoLIC diagram snippet representing a Silent Turn-Yielding

terance is still emitted during the designer's turn, and only later the transition takes place. As exemplified in Figure 3.14, the utterance part represents what the emitter speaks (*e.g.*, card requested) until the "snippet" of the transition-

ing part (*e.g.*, wait), which yields the turn. Therefore, we decided to replace the term *transition* with the more suitable term *turn-yielding*, which means, in the Linguistics' turn-taking system, the cases where the speaker gives the turn spontaneously and, in some cases, may even appeal for a response (Stenström, 1994), which is precisely what MoLIC depicts.

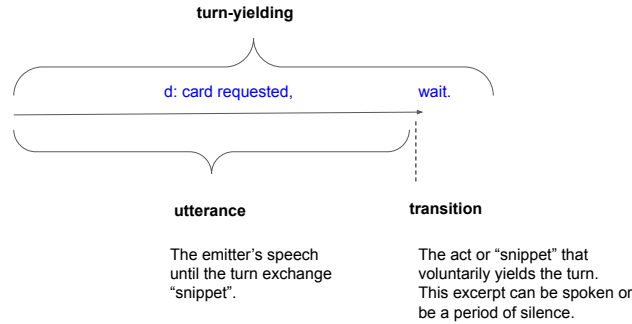


Figure 3.14: Comparison between Turn-Yielding and Transition

Changelog - Designer's (Turn-Yielding) (Breakdown) Repair

Utterance Repair mechanisms are corrections to the conversation, usually meant to deal with turn-taking errors and violations (Sacks et al., 1974). Schegloff et al. (1977) stated that corrections may come from the utterer (meaning a *self-correction*) or from the other agent of the conversation (called *other-correction*). This element represents the latter case: the *other-correction*, through which the designer guides the user to repair a conversation breakdown. Therefore, we renamed this element (previously called *Breakdown Recovery Transition Utterance* in MoLIC V2) to **Designer's (Turn-Yielding) (Breakdown) Repair Utterance**, or simply **Designer's Repair Utterance**.

Changelog - Silent Turn-Yielding This element was previously represented in the HCI textbook by Barbosa and Silva (2010) and in Pessanha and Barbosa (2018), both without an associated name. Based on our studies to realign the language to the interaction-as-conversation metaphor, we decided to turn this representation into an element. To name this element we considered the strategy known as *yielding the turn* (Stenström, 1994). No utterance is emitted during the turn yielding, but the designer needs to identify in which condition the turn yielding occurs. This is represented by prefixing the condition with **if:** (*e.g.*, if: filename and path are valid in Figure 3.13).

3.5

User's Repair Utterance

When the designer wants to allow the user to amend or give up on a conversation, such as editing data they just entered or canceling an action, they represent this by a dashed arrow containing a label that starts with **u:** (e.g., **u: cancel** in Figure 3.15). This element should not be confused with a breakdown, since this is not necessarily an error; it is just an intentional deviation in the conversation, for instance, because they changed their mind or checked that a certain outcome was not what they had intended. Differently from the previously presented utterances, the term turn-yielding was not adequate for this element, since the turn is kept by the user after this utterance is emitted, thus, there is no exchange of turns caused by this element.

Changelog - User's Repair Utterance This element was already listed in MoLIC V2 but used the same name as the *User's Transition Utterance*. However, repairing a conversation is conceptually distinct from advancing a conversation on the same topic or shifting the conversation topic. Therefore, we used here the same reasoning as in the *Designer's Repair Utterance* element and renamed this element to *User's Repair Utterance*. This element represents the self-correction case: the user is repairing their own previous conversation, making a correction to realign the conversation with the desired topic.

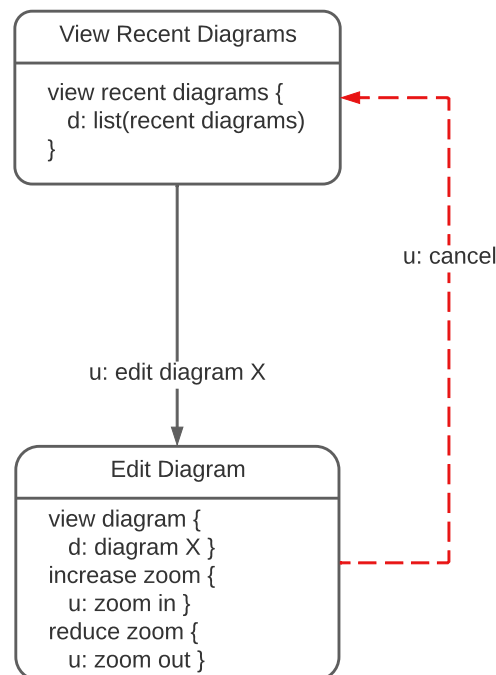


Figure 3.15: MoLIC diagram snippet representing a User's Repair Utterance

3.6 System Process

To convey the idea of a *black box*, this element is pictured as a black-filled square representing the moment when the system processes a request received from the user (Figure 3.16). The outcome depends on the input received and the current system state. After processing, the designer may lead the user to another scene or back to the previous one. When connected to a **Designer's Repair Utterance**, the **System Process** represents the cases where a communication breakdown occurred (*e.g.*, an error due to wrong input or process failure). There were no change proposals for this element.

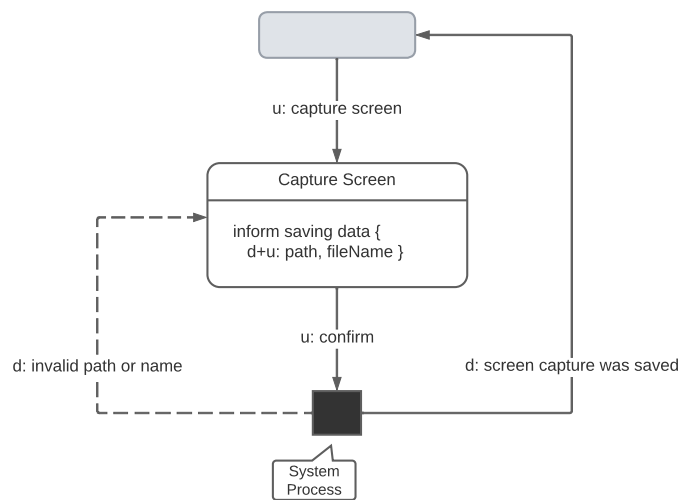


Figure 3.16: MoLIC diagram snippet representing a system process

3.7 Breakdown Prevention Mechanisms

Designers can specify diverse mechanisms to prevent communication breakdowns. For instance, they may inform the user what is expected of them or even restrict their expressiveness and prevent a certain interaction from taking place. Described below are three different options provided by MoLIC to prevent conversation breakdowns: *passive prevention*, *active prevention*, and *supported prevention*.

Passive Prevention (PP) - The goal here is to prevent a breakdown by informing users what is expected of them. For instance, the designer may convey that the value of a sign is required or must be provided in a certain format. This form of prevention does not block the user from proceeding with the conversation, meaning the user can still proceed in a way that will result in a breakdown. In the example of Figure 3.17 (Passive Prevention), the email is

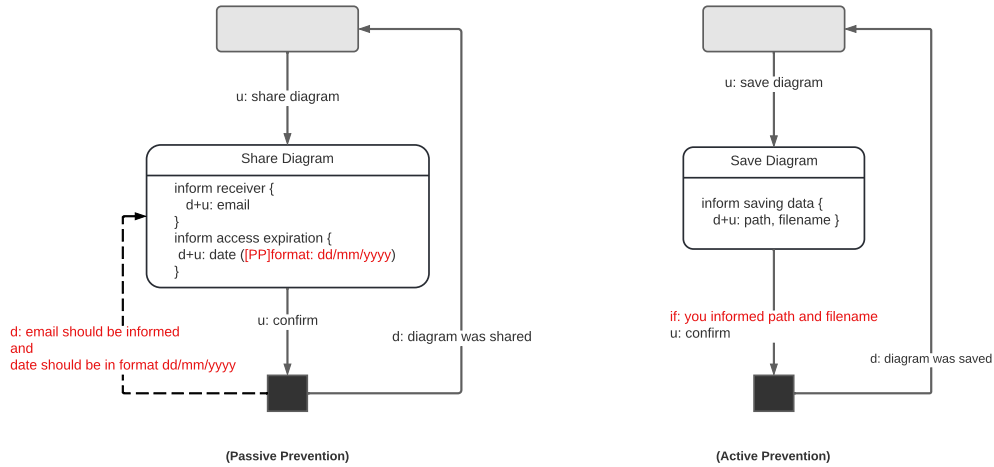


Figure 3.17: Breakdown Prevention Mechanisms: Active and Passive Prevention

required (given the absence of the optional marker) and the date must follow a specific format (*e.g.*, represented by ([PP] format: dd/mm/yyyy) next to **d+u: date** in the figure) to avoid a breakdown. Note that the user can still submit the request without providing an email or by providing the date in a different (invalid) format. However, this would result in a breakdown and the user would be asked to repair the conversation (by being led back to the previous scene).

Active Prevention (AP) - In this scenario, the idea is to block the user from continuing their utterance if this will lead to a breakdown. In the example of Figure 3.17 (Active Prevention), the user can only confirm they want to save the diagram (*i.e.*, emit the **u: confirm** utterance) if the specified condition is met (**if: you informed path and filename**). At the user interface, this solution would likely involve a submit button which is enabled only if the path and filename are provided.

Supported Prevention (SP) - In this form of prevention, the idea is to help the user with a decision that could lead to an error, but ultimately, it is up to the user to decide. An example, pictured in Figure 3.18, is when the user wants to save a file but chooses the name and path of an existing file. In this case, the designer will indicate that there already exists a file with the same name at that path and provide some options for how to proceed, such as renaming the file or path (**u: rename**), overwriting the existing file (**u: overwrite**), or even canceling the operation (**u: cancel**). This extra conversational turn, depicted with an *alert scene*, helps prevent breakdowns that could occur, in this case, by lack of knowledge of the existing file or even by distraction.

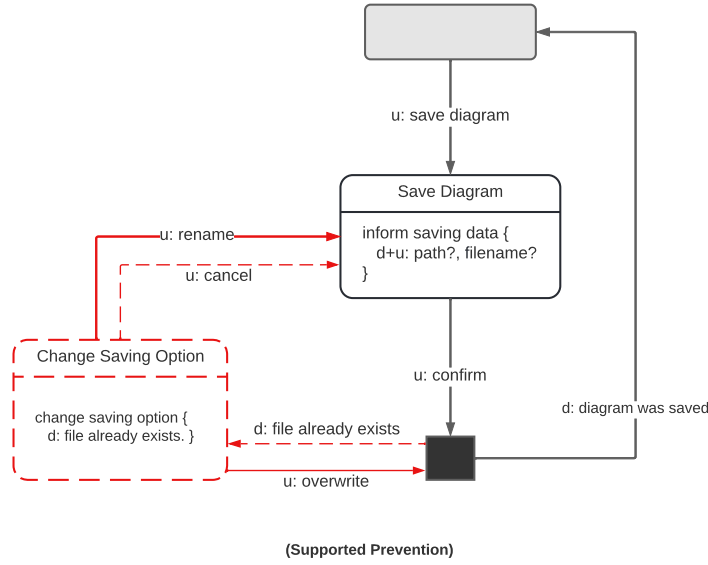


Figure 3.18: Breakdown Prevention Mechanisms: Supported Prevention

3.8

Breakdown Repair Mechanisms

Two breakdown repair mechanisms exist since MoLIC v2: *supported recovery* (now called *supported repair*) (SR) and *error capture* (EC). Both are described below.

Supported Repair (SR) - If the user cannot advance the conversation toward their goal because of either a communication breakdown (*e.g.*, caused by violating a passive prevention specification) or an unexpected result from a system processing (*e.g.*, no items found in a search), the designer should convey the issue to the user and help them achieve their original goal or give them an opportunity to pursue a different goal. Figure 3.19 (Supported Repair) exemplifies this mechanism, allowing the user to try saving the file again when the name is invalid. These scenarios are represented by a ***designer's turn-yielding repair utterance*** returning to the scene where the error occurred.

Error Capture (EC) - This scenario occurs when a breakdown or error occurs that cannot be resolved within the system in order for the user to achieve their original goal. For instance, when a required service is unavailable (*e.g.*, network is down, server overloaded) or when the file is corrupted. In this case, the designer may emit a ***designer's turn-yielding repair utterance*** leading to a *topical conversation completion*, just to inform the user about the error (and, if possible, its causes and means of repair) so that they can try to solve it outside the system, as there is nothing the user could do within the system to fix the issue.

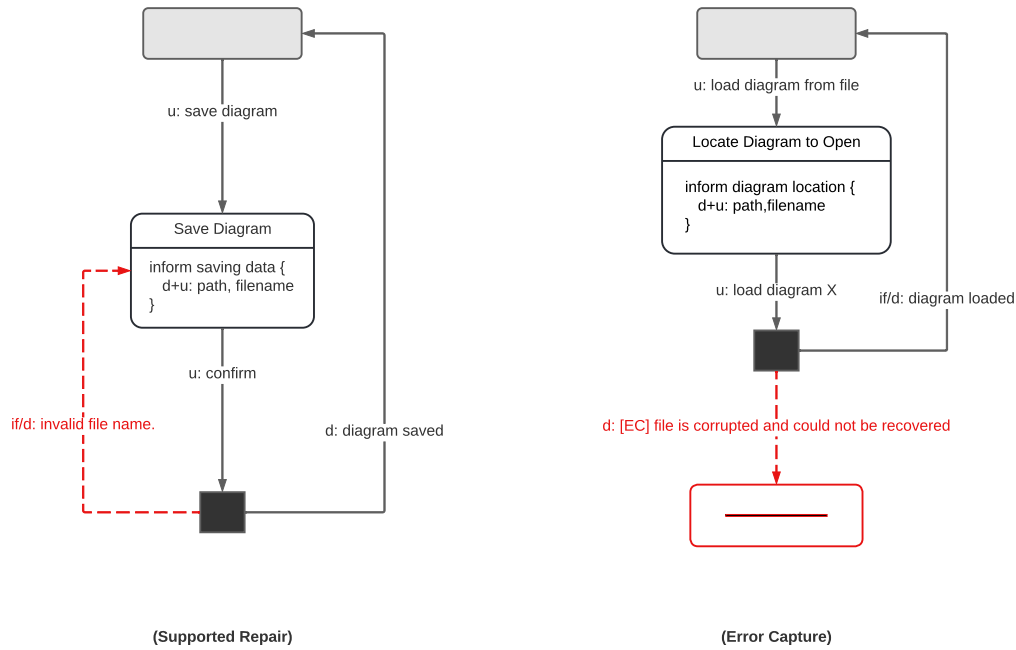


Figure 3.19: Breakdown Repair Mechanisms

Changelog - Error Capture Previously, this scenario was represented with a monologue, which was deprecated, as explained in subsection 3.3.4.

3.9

Simultaneous Speeches

Simultaneous Speeches are elements that allow the design of interruptions by either the user or the designer. Each type of interruption is represented by an element and its respective representation as explained in the following subsections and pictured in Figure 3.20.

3.9.1

Simultaneous Speech in the Designer's Turn

This element is used to represent the progress of a system process (akin to “opening the black-box”) and, optionally, give the user an opportunity to interrupt it. It is pictured as a scene connected by two parallel lines with the system process being “overlapped”. Figure 3.21 exemplifies the uses of this element in the scene **View Ongoing Search Status**, where the user can, while the search/processing is happening, check its status (*e.g.*, view the partial result of diagrams found (*e.g.*, dialogue **view partial results**), view the percentage of the search that is completed (*e.g.*, dialogue **view search status**), and even interrupt the current conversation (in this case, the search) by means of a User's Repair

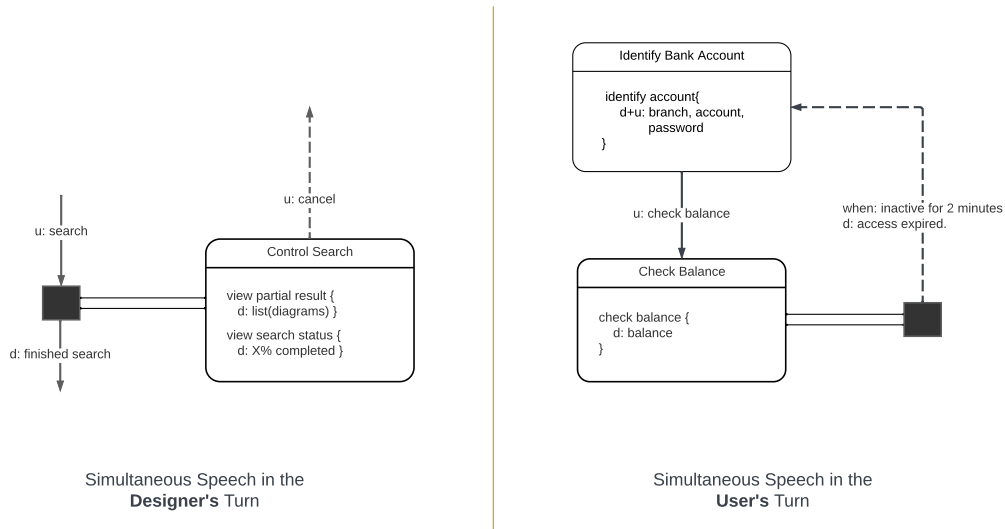


Figure 3.20: Simultaneous Speeches Variations

Utterance (*e.g.*, **u: cancel**), thus effectively grabbing the conversational turn.

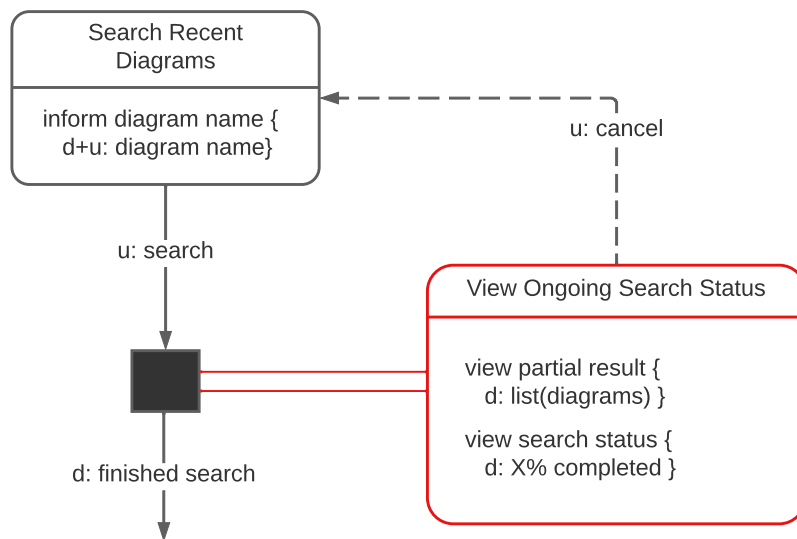


Figure 3.21: MoLIC diagram snippet representing a Simultaneous Speech in the Designer's Turn

3.9.2

Simultaneous Speech in the User's Turn

This element allows the designer to define an interruption in the conversation during the user's turn. This interruption occurs when a specific condition is met, as defined by the term **when**. This element is represented by a system process connected by two parallel lines with the scene that is being “over-

lapped”. This system process is paired with a *designer’s repair utterance*, which leads the conversation to a different topic when the specified condition is met (*e.g.*, when: inactive for two minutes) in Figure 3.22).

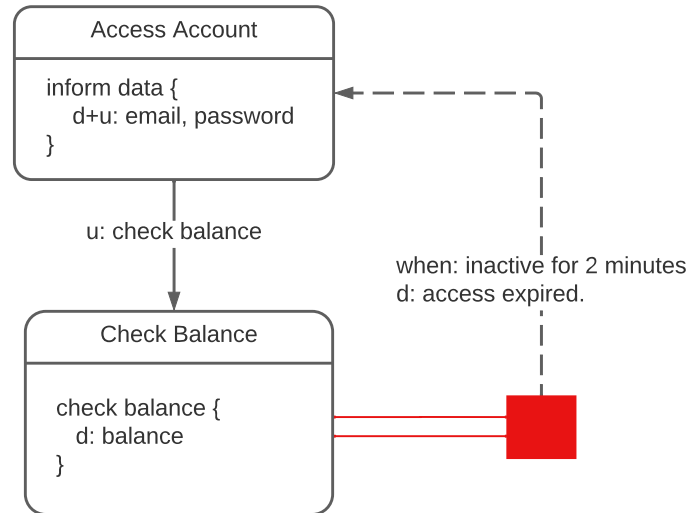


Figure 3.22: MoLIC diagram snippet representing a Simultaneous Speech in User’s Turn

Changelog - Simultaneous Speech

Silva and Barbosa (2005) defined two elements to communicate synchronously about a system process status. Figure 3.23 represents the first element, which allows the designer to inform the user of a *single* sign during the process execution. The second element, a system process coupled with a scene (exemplified in Figure 3.21), can communicate about one or more signs. Thus, we decided to deprecate the element presented in Figure 3.23, as the latter makes the overlap explicit and represents the user’s overlapped turn as a conventional scene, instead of an extraneous “black box / white box” element.

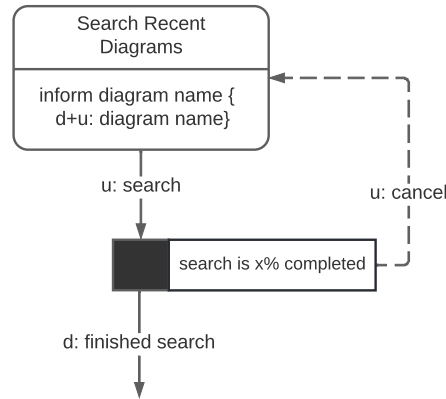


Figure 3.23: Deprecated Element due to Redundancy

In this work, we also added the element ***Simultaneous Speech in the User's Turn***. A variation of the original element (called ***Simultaneous Speech in the Designer's Turn***), where the designer will be able to define an interruption to the user's turn in a composition with the element **when**, described in section 3.20.

The treatment for the designer's and user's simultaneous speech depends on who has the conversational turn. First, we must consider that not all simultaneous speeches cause an *interruption* (Roger et al., 1988). In this sense, the overlap in the designer's turn may or may not allow the user to interrupt that turn. If the user only engages in the simultaneous conversation to check the system process status, this is not characterized as an interruption and is seen as "continued listener attention and interest" Roger et al. (1988). If the user decides to cancel the system process (through a user's repair utterance), then it is perceived as an interruption in the designer's turn (Roger et al., 1988).

Another case is the overlap in the user's turn, which can be represented using the element **when**. In this case, when the condition defined in the **when** element is met, the user's turn is disrupted and the designer takes the turn. As both terms *overlap* and *simultaneous speech* can be used interchangeably (Laskowski et al., 2012), we included both in the questionnaire to gather MoLIC users' thoughts on them.

3.10

Global Topic Shift

This element allows the user to shift the conversation topic at any moment during the interaction. Through the user utterances going out of this element, the user may move the conversation to the scenes they connect. This element

is pictured as a gray rectangle with rounded corners (Figure 3.24). It is important to mention that, in each utterance available from a **Global topic shift** element, the designer may represent a condition to be met (using the **if** element) in order for the user to be able to emit that utterance and start the conversation on that topic.

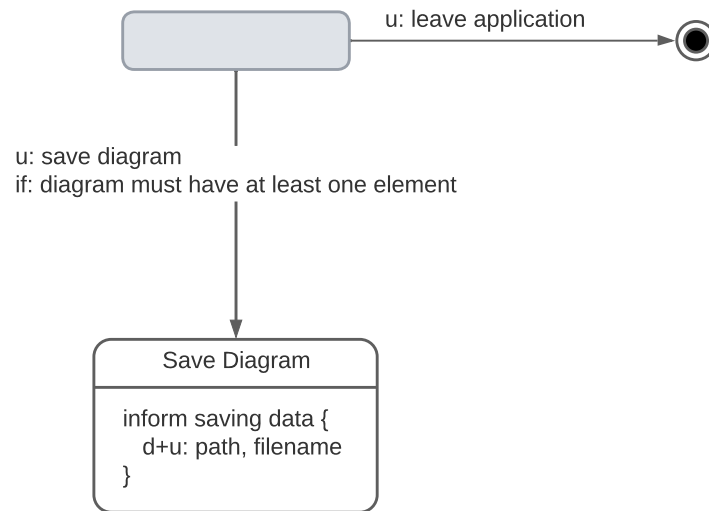


Figure 3.24: Topic Shift

Changelog - Global Topic Shift This element was previously called *Ubiquitous Access* since it allows the user to change the conversation topic at any moment. However, to better align the language with the interaction-as-conversation metaphor based on linguistic concepts, we renamed it to **Topic Shift**, as informally suggested by Pessanha and Barbosa (2018). Brown and Yule (1983) discusses the notion of *topic* as related to the separation of conversational discourse into “chunks” (which can be viewed as different things depending on the context). They also define *topic shift* as symbolizing the migration from one topic to another (Brown and Yule, 1983), which is the idea underlying this element.

3.11

If

This element represents a condition associated with another element, either a turn-yielding element, a dialogue, or a sign. The term **if** is used to create a condition that must be met to proceed with the conversation. Figures 3.25 and 3.26 show examples of this element in various scenarios.

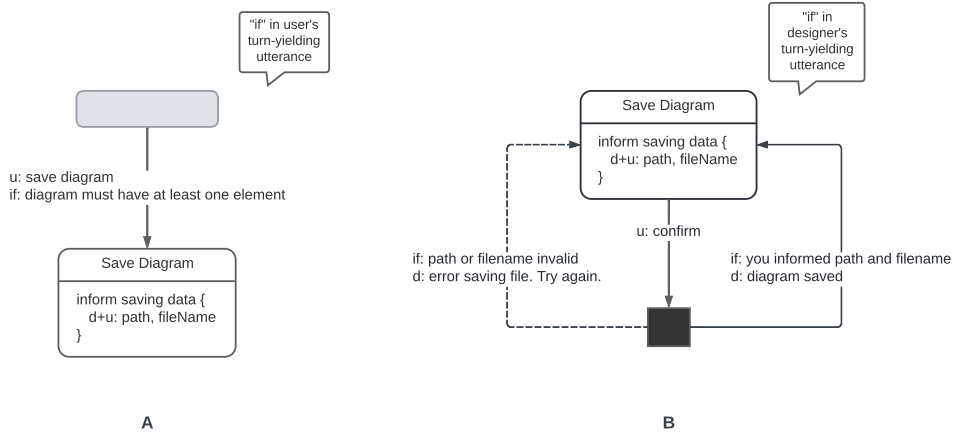


Figure 3.25: MoLIC diagram snippet representing conditions in turn-yielding elements

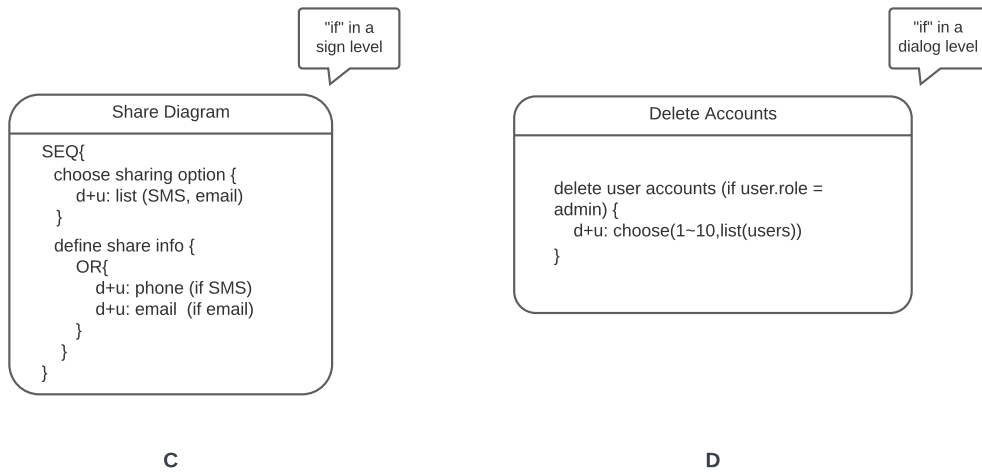


Figure 3.26: MoLIC diagram snippet representing conditions within a scene utterance at sign level and at dialog level

As pictured in Figure 3.25a, the **u: save diagram** utterance to shift the conversation topic will be available only if the diagram that the user is working on has at least one element (specified in the **if** clause). Otherwise, the user will not be able to save the diagram. In Figure 3.25b, **if** is used in both the designer's turn-yielding utterance (if: you informed path and filename; d: diagram saved) and the designer's turn-yielding repair utterance (if: path or filename invalid; d: error saving the file. Try again). In this example, depending on the result of the system process, one of the conditions will be met, defining which turn-yielding utterance will be emitted by the designer and, therefore, the next scene in the conversation flow.

In the examples of Figure 3.26, both cases are used inside a scene. The

difference is that, in 3.26c, **if** is paired with a sign. In this case, the sign could be emitted in the dialogue only if the condition is met. Analyzing the **Share Diagram** scene, the user has to choose a sharing option (**SMS** or **email**) and, subsequently, define the sharing info. Thus, if the user selected the option **SMS** in the **choose sharing option** dialogue, they will only be able to inform the phone due to the restriction imposed by the **if** element.

In Figure 3.26d, the element **if** is also used inside a scene but at a dialog level. In this case, the dialog **delete user accounts** will be available in the conversation only if the **user** is associated with the **admin** role.

The example in Figure 3.26d uses a strategy that allows the designer to map multiple user roles in the same MoLIC diagram. This strategy should be used sparingly and with caution, and typically only when these roles differ little. To do so, the designer used `if: user.role = admin` in the dialog **delete user accounts** to limit the access according to the user's role. This approach requires designers to pay close attention to which user role has access to which elements. When the users' roles differ at multiple points of the interaction, creating one diagram for each role is better. In section 4.1, we detail these usage scenarios.

The element **if** also helps to record the business rules and software logic in the diagram, helping the communication with the developer team, making clear the difference between the condition to be verified (in the software logic) and the information to be communicated, as in `if: account.balance < 10; d: insufficient funds to request card`. In contrast, if the condition coincides with the designer's utterance, they can be combined, as in `if/d: invalid email address`. These examples evidence the need for both representations (separate **if:** and **d:** *versus* combined **if/d**).

Changelog - If Previously represented with the expression *pre* (in MoLIC v1 and v2), this element received two change proposals: Araujo and Barbosa (2008) proposed to change the expression to *precond* and Pessanha and Barbosa (2018) to *if*. This element is supposed to represent the conditions that should be met in order to allow either the user or designer to emit their corresponding utterance (or turn-yielding utterance) (de Paula and Barbosa, 2003). With this in mind, it is possible to relate to Linguistics' sentential connectives, such as "If A then B", meaning that the conditions(s) defined in A must be true in the current context in order to allow the occurrence of (the utterance) B (Karttunen, 1974). In our case, the condition in A is associated with the element **if** in the utterance that, if satisfied, allows the utterance defined in B to be emitted. Furthermore, since the element was already informally called *if* by several HCI students, we accepted Pessanha and Barbosa (2018)'s proposal.

3.12

Context Feature

This element allows the designer to specify aspects of the interaction context that can influence the conversation (Araujo and Barbosa, 2008). For instance, as exemplified in Figure 3.27 with `effect: {session.search_results: true}`, the designer can record for the current session that the user has search results available. This result can be accessed at any moment in the conversation through the **Global Topic Shift** element. By using **context features**, the designer can explore any partition of the context. When the context feature is related to a *session* (as in `session.` prefix), it is volatile, meaning that when the conversation ends (*i.e.*, the user leaves the system), that feature value is discarded. In cases where the context feature is related to more than one component of the context (for instance a configuration related to a certain document when being edited by a certain user), we associate the signs that represent this composition as **(sign1+sign2)**. The following items exemplify various specifications of **context features**:

- `session.search-results` (*e.g.*, to represent an active list of search results)
- `account` (*e.g.*, to assess whether an account has been defined)
- `account.withdrawal-limit` (*e.g.*, to represent the withdrawal limit of the current account)
- `(account+user).statement-period` (*e.g.*, to represent the statement period configured for the current user's and for the current account)
- `(user+document).default-palette` (*e.g.*, to represent the current user's default palette for the current document)

Changelog - Context Features Originally treated as *context*, this concept was first introduced by Silva and Barbosa (2005), but only later, in Araujo and Barbosa (2008) this concept was represented through the following proposals: **articulated sign**, and **context sign**. The *articulated sign* was intended to represent a partition of the context (*e.g.*, session, user, document), while the concept of *context sign* was expected to be related to one or more articulated signs to represent part of the interaction context (*e.g.*, login, language, preferred_color) Araujo and Barbosa (2008). We submitted these proposals for evaluation. Initially, we incorporated the concept *articulated signs* but changed the element's name to **context features**. We grounded this definition on the linguistic concept of the same name, which refers to a formal discourse domain that limits the possible interpretations while also helping to understand the message context (Brown and Yule, 1983). Lists of

context features were elaborated by field specialists (Brown and Yule, 1983) to characterize communicative events by capturing aspects of the conversation context into categories (Brown and Yule, 1983).

3.13 Effect

This element represents that a conversation has consequences that affect the future user-system conversation. An *effect* represents a change in the context of the conversation as the result of the previous exchanges between the user and the designer. Besides influencing future conversations, such change is usually communicated to the user when it occurs. In Figure 3.27, when a user requests a search, the results remain available to be consulted whenever the user wants (*effect*: {session.search_results: true}) during the current session. In the figure, the user could consult the search results at any time via a topic shift (u: view search results), but only if the condition (*if*: {session.search_results == true}) is satisfied. In this example, whenever a search yields results, the system state regarding the current session is turned to *has active search results*. In this example, the context will change as the result of a *silent turn-yielding* that has the element **effect**, and the search results will remain active throughout that session (as indicated by the session.search_results context feature) or until the user performs a search that yields no results (as illustrated by the *designer's repair utterance* if/d: search has no results; effect: {session.search_results: false}).

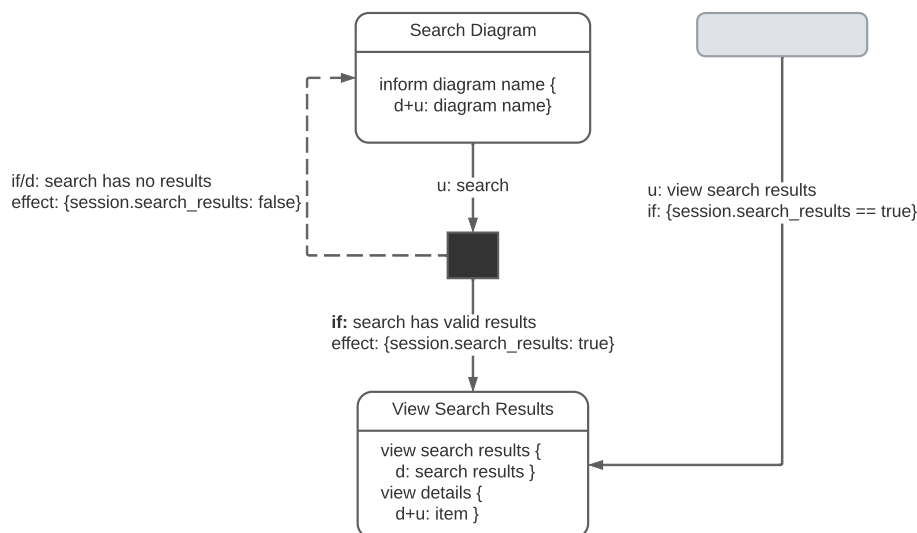


Figure 3.27: MoLIC diagram snippet representing an effect

Changelog - Effect This element was previously called **post-condition**

(post) by Silva and Barbosa (2005), and **perlocutionary effect(perl)** by Silva and Barbosa (2007). Araujo and Barbosa (2008) later proposed to change its representation to **perloc** to reinforce the concept of perlocutionary effect. As it is the assertion of a change in the system state, it fits Searle's classification of illocutionary acts as an *assertive* speech act (Searle, 1975; Leech, 1983). Linguistics adopts the term *effect*, and one of the most common words used in conversation to imply an effect is *then*. As we have *effect* and *then* as alternative terms that respect the interaction-as-conversation metaphor, we decided to assess MoLIC users' perception of either term.

Araujo and Barbosa (2008) suggested an addition to the concept, allowing the designer to make explicit or not the existence of the effect to the user. We accepted this suggestion, allowing either an explicit communication (using **if:**, **d:**, and **effect:**) or an implicit communication (using only **if:** and **effect:**), as explained in chapter 4.

3.14

Let

This element is a suggestion we made in this work. It allows the designer to explicitly define an initial value for a context feature (Figure 3.28). Context features defined with **let** may be changed during the conversation through **effect** elements. When the designer wants to define that the context starts with a specific value, they can use the new element **let**, as shown in Figure 3.28 (*e.g.*, `let: {default_zoom: 50%}` and `let: {default-printer: HP174}`); in the leftmost example, when the user emits the utterance `u: edit diagram X`, the `default_zoom` is automatically set to 50%. While in the rightmost example, when the user enters the scene `Set Default Printer`, the default printer is automatically assigned to the HP174 printer.

Changelog - Let According to Searle (1975)'s classification of illocutionary acts, declarations (or declarative speech acts) directly turn the spoken content into reality by changing the status of the specified object if the declaration is *successfully performed* (Leech, 1983). To represent this concept, we proposed the element **let** in MoLIC V4, which turns what was declared into reality by adding it to the conversation context.

3.15

External Interlocutor and Fork

When a conversation in the system leads to another system, the element to represent this external connection is the **external interlocutor** (a circle half white, half black). There may also be scenarios where the conversation not

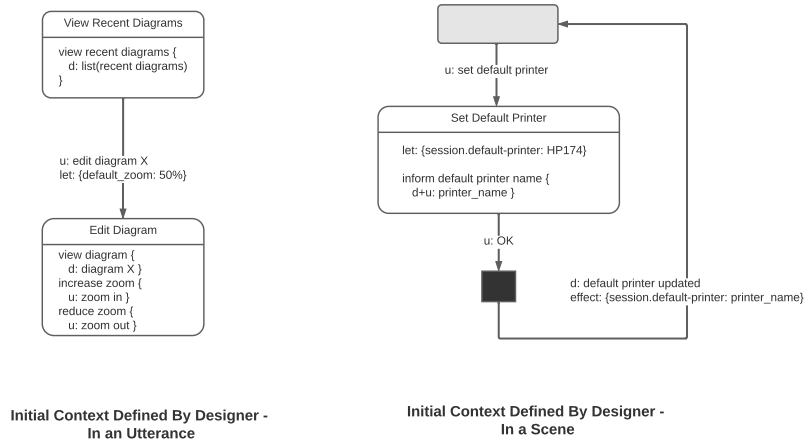


Figure 3.28: MoLIC diagram snippet representing the use of the **let** element

only triggers an external system but also, at the same time, leads the user to another scene inside the system. In this case, it is mandatory to use a ***fork***, a horizontal line that divides the unique flow of the application into two. An example is when the system sends an email and then shows a message to the user. The former is the ***external interlocutor*** and the latter is the conversation the user will be led to after the ***fork*** (Figure 3.29). This element was kept in MoLIC V4, unchanged.

3.16

Why

The element **why** is an annotation to the diagram that records the designer's reasons for some of their interaction design decisions represented in MoLIC. These decisions are based on all the knowledge acquired by the designer in the previous steps of the solution conception, *e.g.*, research about the user and the domain. By using this element, the designer is able to leave a trail of their decisions as exemplified in Figure 3.30, where the designer indicates that the first scene is **View Recent Diagrams** because they learned (*e.g.*, from interviews or user observations) that the user prefers to continue working in the recently edited diagrams.

An important note about the **why** element is that it does not affect the interaction in any way. The reason recorded by the designer may or may not correspond to reality.

Changelog - Why This element was defined by Araujo and Barbosa (2008) as *presup*, to be used on the designer's turn-yielding utterances. Later, Pessanha and Barbosa (2018) suggested changing it to *justify*. We included the element in this edition with two modifications: (1) we renamed it to **why**,

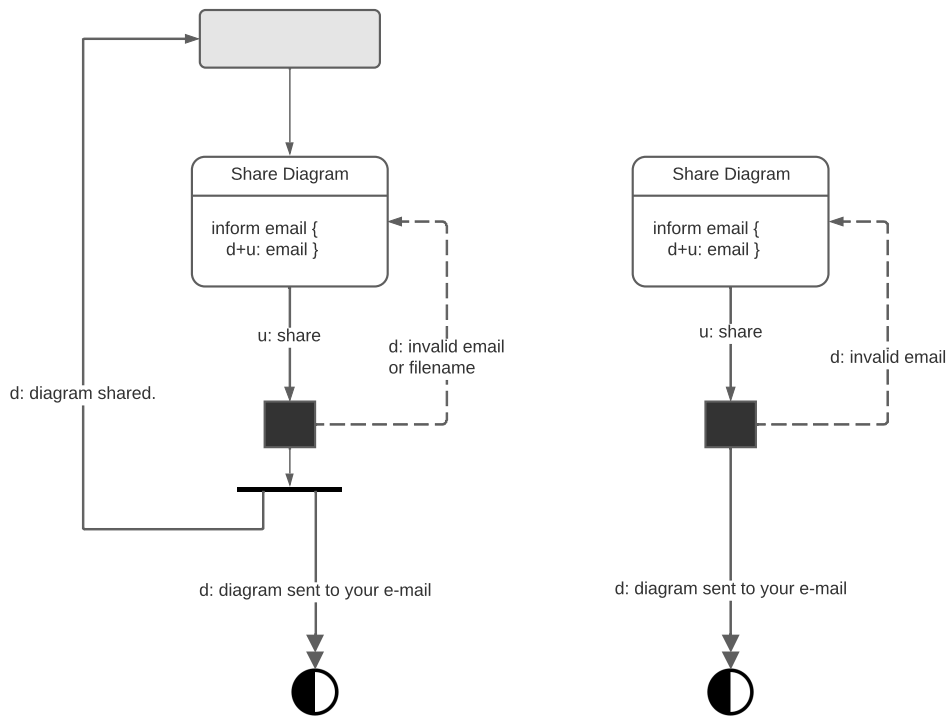


Figure 3.29: MoLIC diagram snippet representing External Interlocutor and Fork

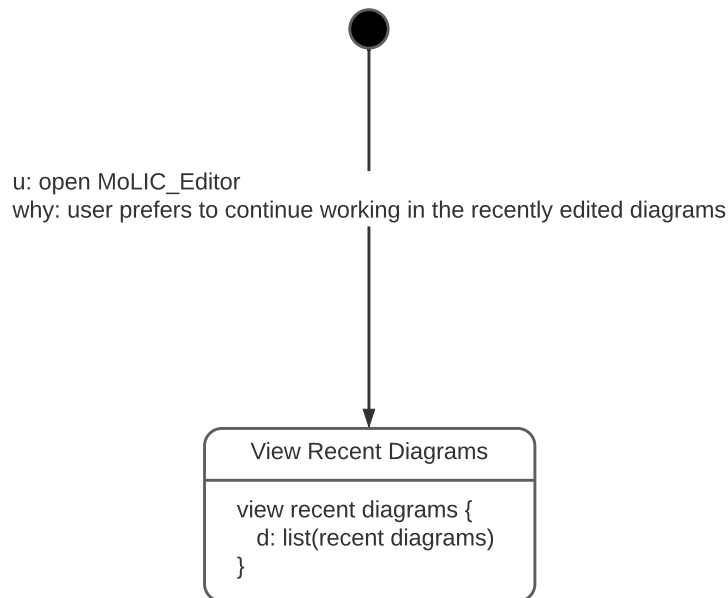


Figure 3.30: MoLIC diagram snippet representing a Why

as it records (in the diagram) the reasons for the design decisions made; and (2) we expanded the use of the element **why** to dialogues (as exemplified in Figure 3.31).

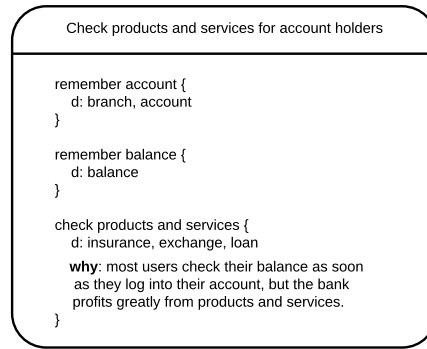


Figure 3.31: Element Why used in a diagram

3.17

Preferred Conversation

When the designer defines more than one way of achieving a goal, they may want to mark one specific conversation as the preferred one in the diagram, *i.e.*, the one they expect users to engage in at that moment. To do so, designers need only use a bolder user's turn-yielding utterance to mark the preferred conversations to achieve a goal, making that conversation stand out. At the user interface, a preferred conversation may be represented as a primary button to emphasize the preferred way to achieve a specific goal.

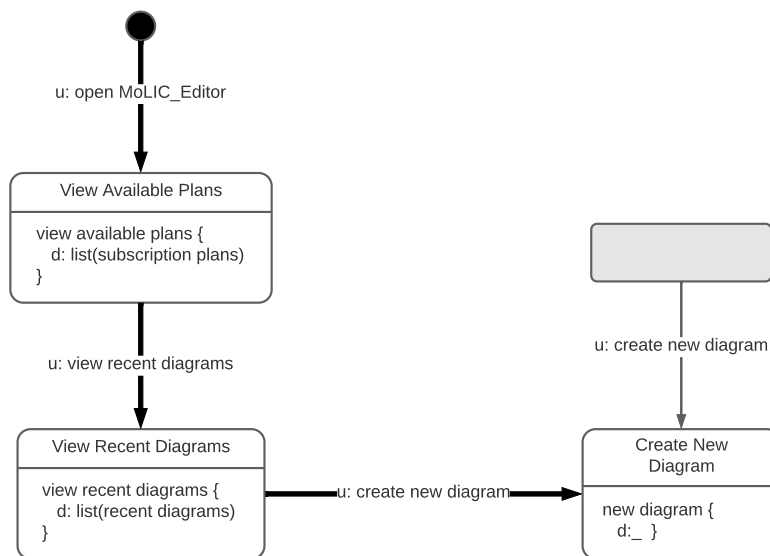


Figure 3.32: MoLIC diagram snippet representing a Preferred Conversation

Changelog - Preferred Conversation Suggested by Araujo and Barbosa (2008) in a different form, we included this element but modified it to avoid too much clutter in the diagram. Their proposal was to use a gcc (Goal-related Conversation Closing) element to mark only the final preferred

conversation to achieve the goal, which does not make clear the full preferred conversation nor is it visually clean.

The **preferred conversation** element is based on the linguist concept that a question may be built in a way that exposes the answer expected by its sender Sacks (2010), affecting the receiver so that the latter will tend to answer in a way aligned to the sender's expectations. Therefore, we allow the designer to record their preferences about how they expect the conversation to proceed (more often).

3.18 Main Scene

Araujo and Barbosa (2008) alerted for the lack of identification for a *main scene*, which is typically the starting point of most conversations (*e.g.*, often realized at the user interface as the main page on a web application) (Araujo and Barbosa, 2008). In order to differentiate the main scene from the others, the designer can use a thicker border in the scene, as exemplified in Figure 3.33 with the scene **Edit Diagram**.

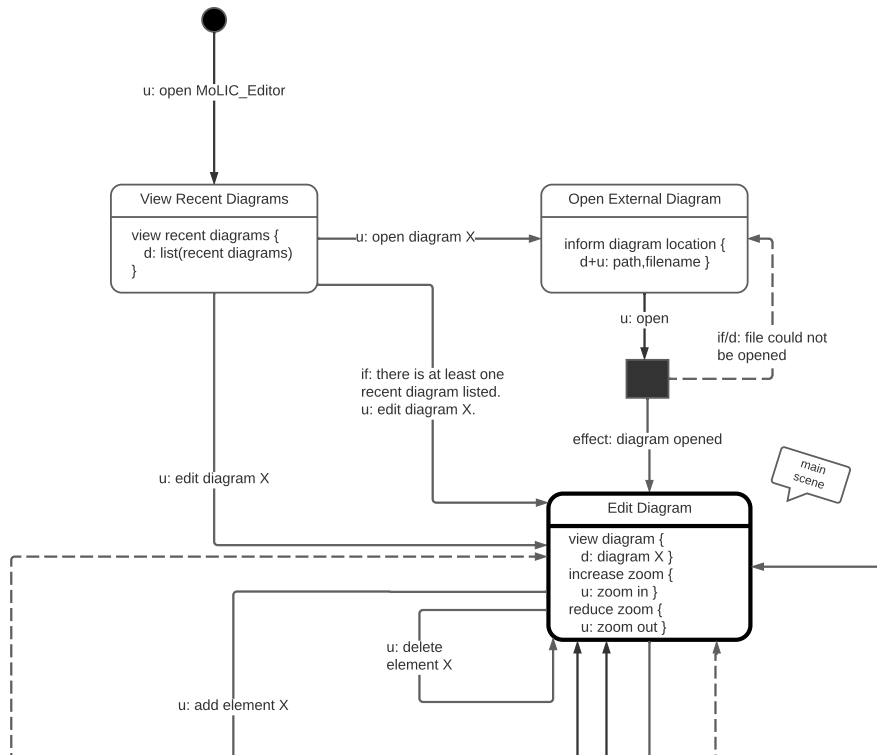


Figure 3.33: MoLIC diagram snippet representing the Main Scene

Changelog - Main Scene Suggested by Araujo and Barbosa (2008), we included this element, but once again in a different form to avoid diagram clutter. The original idea was to use a gco (Goal-related Conversation Opening)

element in all the user's turn-yielding utterances starting from the main scene, resulting in many added visual elements, which in turn pollute the diagram and reduce its readability.

3.19

Contact Points

In case there are multiple users interacting with the application (*e.g.*, playing more than one user role), MoLIC offers two different forms of modeling the interaction diagram. If the interaction of each user role with the system differs very little, it allows the designer to use only one diagram for all scenarios. In contrast, if the user roles differ substantially, creating one diagram for each role is better. In the latter scenario, if one of the roles influences the other's interaction with the system somehow, MoLIC offers ***contact points*** to represent such influences.

Contact points are represented by a circle with text (with the user role(s) associated), connected by a turn-yielding utterance to a scene or a process. Figure 3.34, adapted from Silva and Barbosa (2007), represents the perspective from both sides of the connection. Considering that the fictitious MoLIC Editor allows the user to receive revisions in their diagrams, the upper box of the example represents the **reviewer** role's diagram, in which a reviewer can request a list of adjustments to be made by the author. In the lower box, we have the **author** role's diagram, indicating the author will receive the request through the ***contact point*** and will be able to either accept or deny the request adjustments. The user can also send the result of their analysis to the reviewer through the ***Contact Point***.

Changelog - Contact Points This element was extracted from the work of Silva and Barbosa (2005) and left unchanged in MoLIC V4.

3.20

When

This element, proposed in MoLIC V4, allows the designer to define event triggers that will watch the conversation and interrupt the current user's utterance, redirecting the conversation to another scene. This definition occurs before interaction time and is triggered as soon as the predefined condition is met.

Changelog - When

Pessanha and Barbosa (2018) suggested a concept where the conversation topic would automatically change if the user emitted no utterances during a predefined period. They denominated this element as an *expiring scene*. We

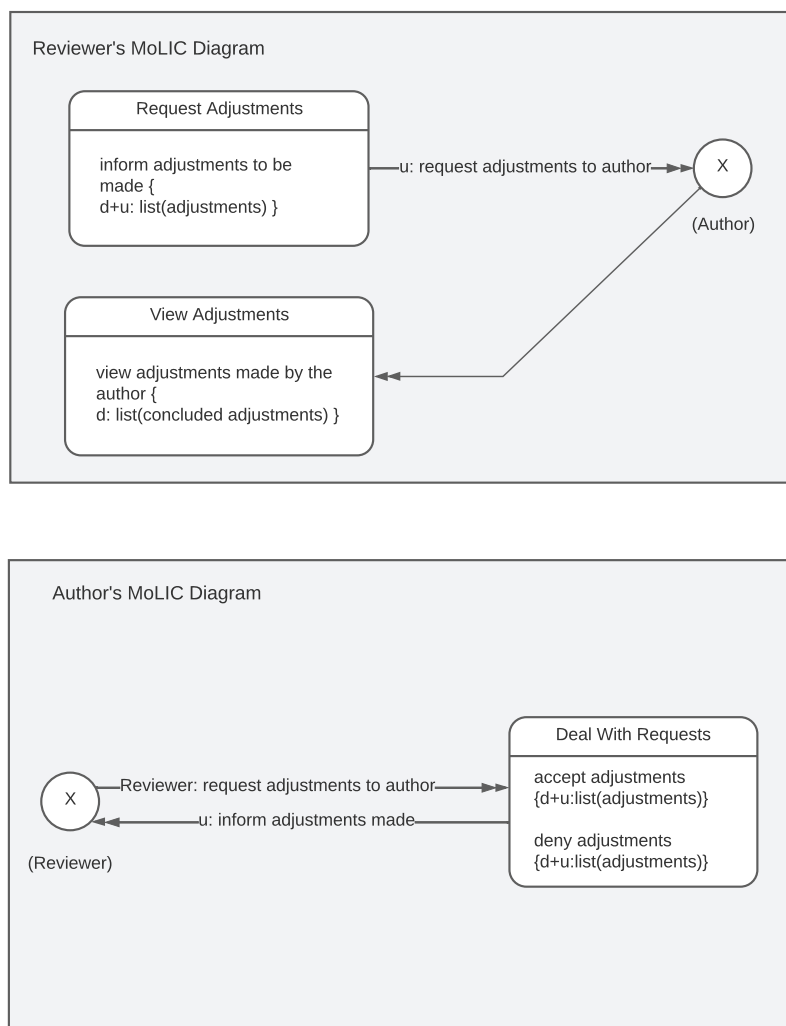


Figure 3.34: MoLIC diagram snippet representing the concept of Contact Points

decided to bring this concept to MoLIC V4 in a more generic form and with better alignment with the interaction-as-conversation metaphor. To do so, we identified in Linguistics the notion of *when-clauses*. Sentences such as “When I arrive, John will leave” (adapted from Declerck (1997)), are examples of *when-clauses* used as adverbial clauses (Declerck, 1997). As adverbial expressions may cause topic shifts in a conversation (Brown and Yule, 1983), we can then use the term **when** in cases where an external event affects the conversation.

3.21

Elements Removed from MoLIC V4

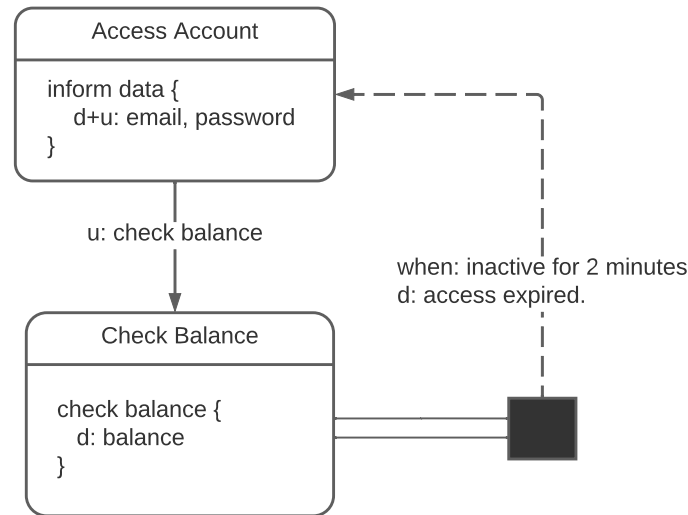


Figure 3.35: MoLIC diagram snippet representing the usage of the element *when*

3.21.1

Combined “Forward-Backward” Utterances

This element was conceived as an attempt to simplify the representation of the association of a user’s turn-yielding utterance with a user’s repair utterance (*e.g.*, going back to the previous scene). With this element, the designer would be able to merge both elements, as presented in Figure 3.36, ideally maintaining the same expressiveness with only one utterance. The conversation flowed initially in the direction of the black arrow, and the user would be able to return to the previous scene in the direction of the white arrow, meaning that the user reconsidered their decision (*e.g.*, as it occurs when the user presses a *back* button or link at the user interface).

This element was originally proposed by de Paula and Barbosa (2003), suggesting the inclusion of a white arrow in the source of a user’s turn-yielding utterance as a possibility to return to the previous scene. Later, Pessanha and Barbosa (2018) revisited this concept and proposed *Special Transition Utterances*, naming the original concept from MoLIC V1 as *utterance one-way* and proposing a new element (*utterance both ways*).

In MoLIC V4, we decided to exclude this element from the language since it creates ambiguity. To illustrate this issue, consider the example in Figure 3.36, which included the combined turn-yielding utterance starting from scenes A and B and leading to a system process. In this case, the use of the combined utterance would create confusion since there is no indication informing where to return to after the system process, or even whether the return would be possible at that point.

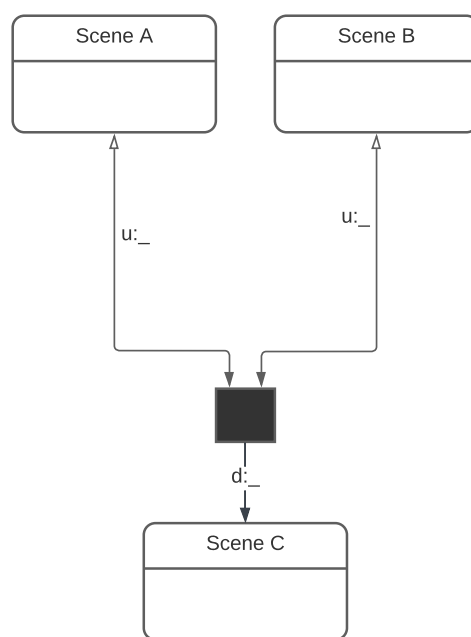


Figure 3.36: Combined Utterances - Scenario with Ambiguity

This same scenario would be correctly and clearly represented as shown in Figure 3.37 using separate turn-yielding utterances. In this case, the ambiguity is gone; the turn is returned to the user only after the system process. Then, the user can cancel (`u:cancel`) and repair the previous conversation, if necessary.

3.21.2 Stereotyped Scenes

The idea of the element **stereotyped scene** is to represent independently from the main diagram, a generic situation that occurs several times in a diagram (Araujo and Barbosa, 2008). Thus, simplifying the diagram by avoiding repetition.

Although this element, suggested as future work in de Paula and Barbosa (2003) and proposed as an element by Araujo and Barbosa (2008), may contribute to defining interaction design patterns by allowing the reuse of a combination of elements along the diagram, we decided not to cover it, as we deemed both their proposals as immature to compose the consolidated version of MoLIC, MoLIC V4.

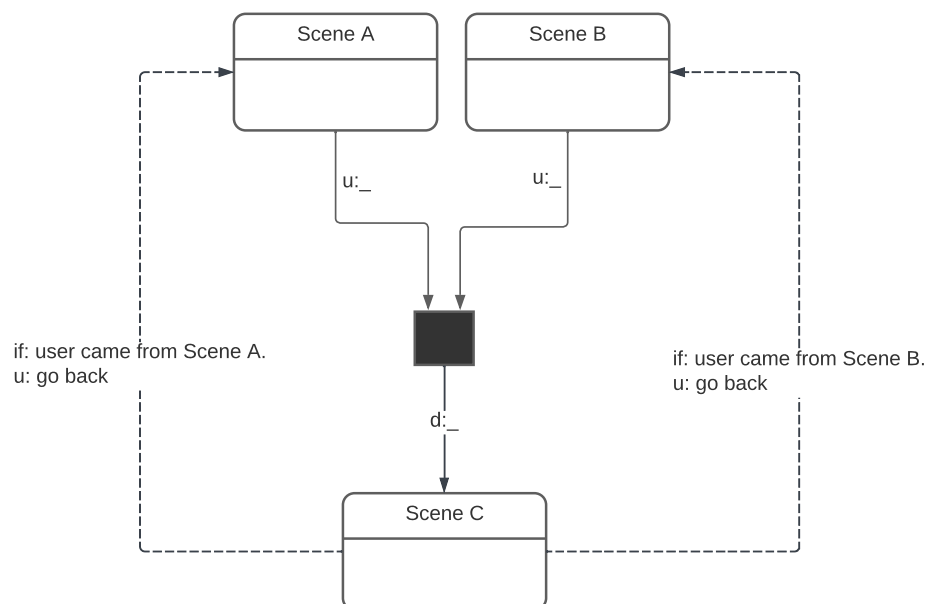


Figure 3.37: Separated Turn-Yielding Utterances - Scenario without Ambiguity

4

Usage Examples

This chapter presents usage examples of MoLIC V4 and suggests some practices to assist the modeling process. We will use as a running example the banking solution represented in Figure 4.1. The diagram in the figure models a banking system for the user role of an account holder. In section 4.1, we explain how to map user roles onto MoLIC diagram(s).

4.1

User Roles in MoLIC Diagrams

When designing an interactive solution through a MoLIC diagram, it is important to consider whether the user roles have a large or a small intersection regarding goals and tasks. If this intersection is large and the roles have little difference, the MoLIC diagrams for their interactions with the system would have many common elements and structures, and maintaining both diagrams might result in inconsistencies and overwork. To address these issues, the designer can create a single diagram and differentiate the roles using an **if** clause in scene dialogues and utterances specific to one of the roles. An option to ease the visualization of the different roles in a single diagram would be to color the common parts of the diagram in black and the parts specific to each role in a different color. In contrast, if the user roles have few similarities, we strongly recommend creating a separate diagram for each role to avoid creating a complex diagram with reduced readability. Note that this work does not include synchronous collaboration between multiple users. This was proposed by de Souza et al. (2015) in a variation called MoLICC. Its revision and consolidation lie outside the scope of this work.

4.2

Successive Refinements

MoLIC diagrams are usually built in successive refinement cycles. First, the designer defines all the topics available in the user-designer conversation and the utterances connecting these topics (Barbosa et al., 2021). In later stages, the designer should include the system processes and detail each scene's dialogues, utterances, and signs (Barbosa et al., 2021). The dialogues may be

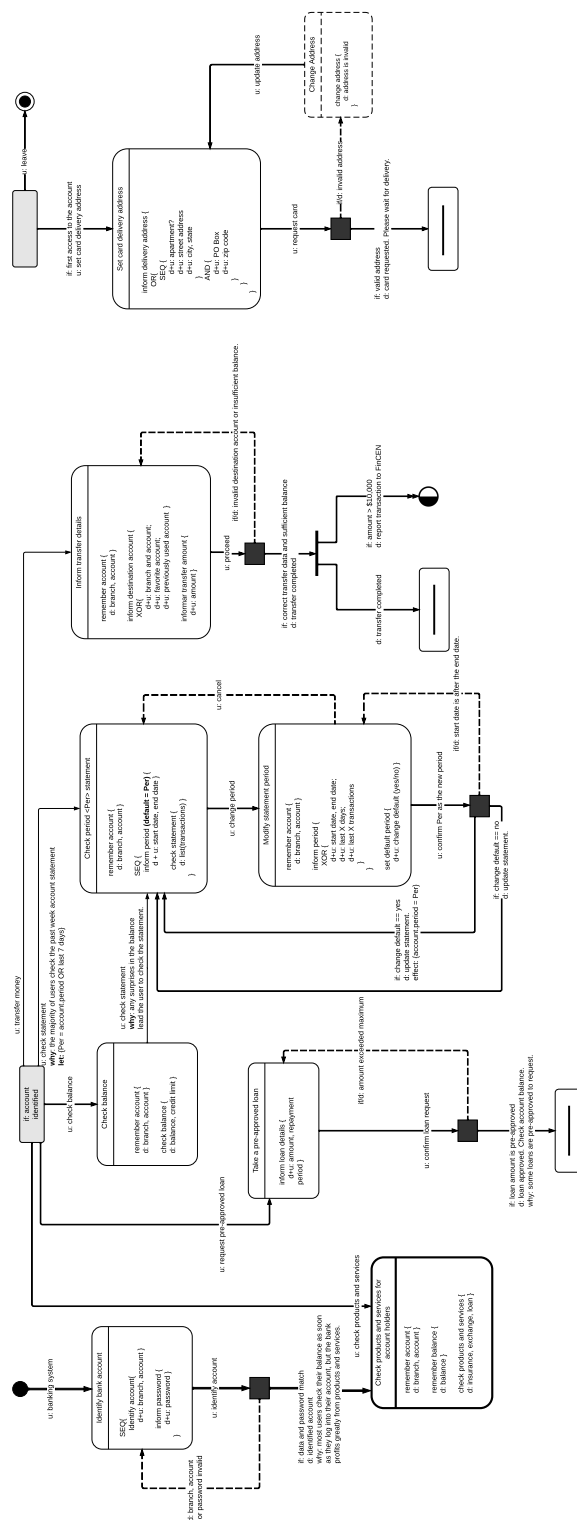


Figure 4.1: Banking example

grouped to create a sequential order or to make explicit whether only one or all dialogues should occur in a specific scene. It is important to note that it is up to the designer to specify (or not) these group operands during diagram creation and evolution. In intermediate stages, the scene dialogue relations

may be absent or implicit until the next refinement. Another aspect that must be considered when finishing a MoLIC diagram is that utterances and turn-yielding elements cannot be empty. Regarding the designer's utterances, if no message is spoken while changing the turn, at least an **if** element should be specified to indicate the corresponding condition (as is the case of the silent turn-yielding element). Regarding the user's utterances, the minimal form is **u: utterance content**.

4.3

Structuring Scenes

When modeling scenes, the designer may want to record in the diagram an intention to be considered when designing the user interface. In the examples of Figure 4.2, Figure 4.2A indicates that the designer thought of the **Set card delivery address** scene as a cohesive interaction. In contrast, by separating the conversation into two topics (in **Set card delivery address** and **Confirm delivery information**), as exemplified in Figure 4.2B and Figure 4.2C, each scene represents a different moment in the conversation, to draw attention to sub-objectives and their corresponding dialogues and information exchange. The main distinction between B and C lies in using the **topical conversation completion** element in C: it makes the end of the conversation clear because the element's representation is highly visually distinct from the scene without the topic used in B. However, one drawback of C is the requirement to introduce a system process solely for the purpose of announcing the designer's response, as all verifications had been done before, and there was only one result possible (*i.e.*, no need for branching out the outcome of the system process).

4.4

Representing Conditions

In this section, we explore the different conditional scenarios that we can represent in MoLIC. We explain each scenario depicted and marked in Figures 4.3 (with scenarios *a* to *d*) and 4.4 (with scenarios *e* to *g*).

Scenario *a*:

There is a condition for the turn exchange to occur after the system processing (if: **data and password match**), but the designer does not need to communicate explicitly about it (hence the absence of a **d:** utterance). In this case, the designer's next communication to the user will be to present the target scene of the **silent turn-yielding** element after successfully identifying the

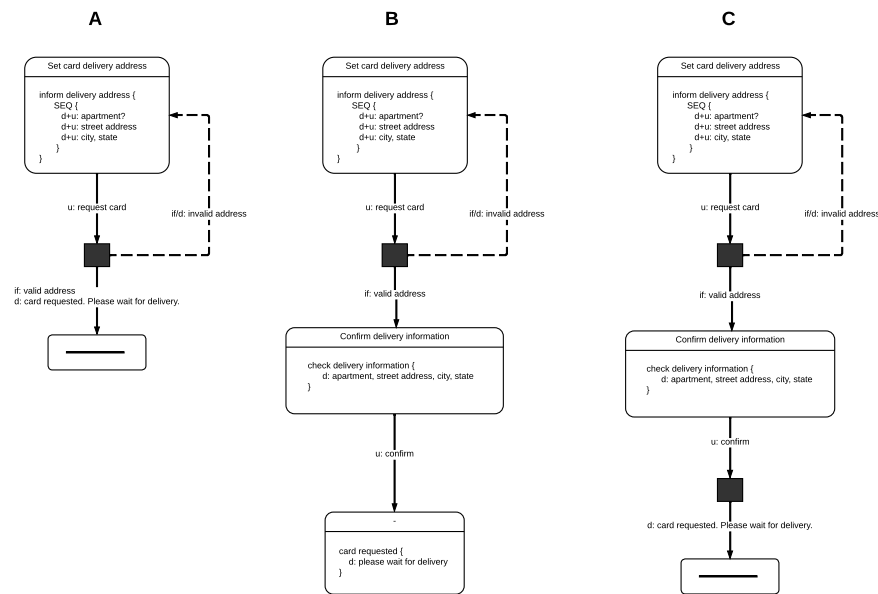


Figure 4.2: Scene Structuring Differences

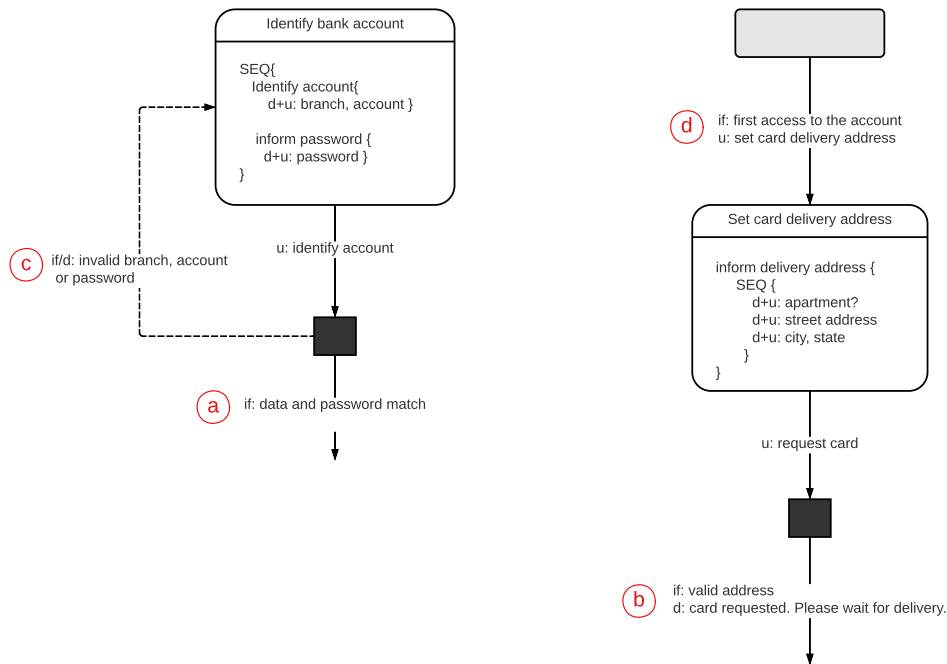


Figure 4.3: Representing Conditions - Scenarios a to d

bank account (left-hand diagram in Figure 4.3).

Scenario b:

There is a condition for the exchange of turns to occur after the system processing (if: **valid address**), and associated with the turn yielding, there will be a designer's utterance informing about the processing result (d: **card requested. Please wait for delivery.**; right-hand diagram in Figure 4.3).

Scenario c:

There is a condition for the turn exchange to occur after the system processing, which coincides with the designer's utterance about the processing result (if/d: **invalid branch, account or password**; left-hand diagram in Figure 4.3). **Note:** de Paula and Barbosa (2003) proposed for this scenario an overload of the designer's utterance by omitting the condition and representing only the emitter (d: **invalid branch, account or password**). We no longer advise this usage since it created conceptual confusion as explained later in the analysis, in subsection 5.7.1.

Scenario d:

There is a condition (if: **first access to the account**) for the user's utterance (u: **set card delivery address**) to be enunciated. This means that, at the user interface, the widget (*e.g.*, link or button) will be disabled or hidden from the user as long as the condition is unmet (right-hand diagram in Figure 4.3).

Scenario e:

Similar to scenario a, there is a condition for the silent turn yielding to occur (if: **change default == no**). The difference here is that there is an *effect* in the conversation context (effect: {**account.statement_period** = Per}; left-hand diagram in Figure 4.4).

Scenario f:

Similar to scenario b, there is a condition for the turn exchange after the system processing (if: **change default == yes**), and associated with the turn yielding, there will be a designer's utterance informing about the processing result (d: **your default period statement was updated**). The difference here is that there is an effect on the conversation context (effect: {**account.statement_period**

= Per}; left-hand diagram in Figure 4.4). While it is usually desirable for the designer to communicate the *effect* of a conversation to users, they can choose not to do so based on their knowledge of the users and the task being supported.

Scenario g:

Similar to scenario c, there is a condition for the turn exchange to occur after the system processing, which coincides with the designer's utterance about the processing result (if/d: available products). The difference here is that there is an effect on the conversation context (effect: {user.products = products available}); right-hand diagram in Figure 4.4).

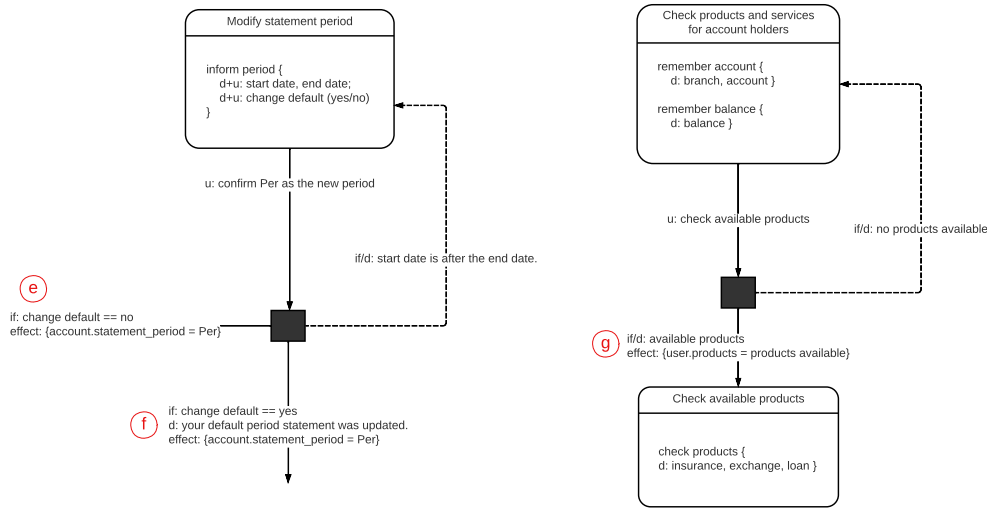


Figure 4.4: Representing Conditions - Scenarios e to g

4.5

Order of Elements in Turn-Yielding Utterances

When labeling the designer's utterance after a system processing, the elements **if**, the utterer (**d** for the designer or **u** for user), **effect**, and **why**, if all present, should preferably be organized in the order shown in Figure 4.5.

In the case of the user's utterance, the **let** element can be added to the turn-yielding element. We suggest placing **let** as the last one in the utterance.

4.6

Adding Conditions inside the Topic Shift Element

A *topic shift* may be associated with the condition that should be satisfied in order to enable any conversation topics available to the user. As

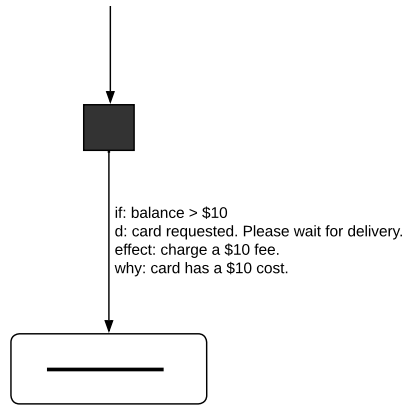


Figure 4.5: Elements order

suggested by Pessanha and Barbosa (2018), the designer can now add an **if** clause inside the element box. This was not included in MoLIC V4 as a new element but as syntactic sugar. Depicted in Figure 4.6, we have a simplified version of the banking diagram using the condition inside the topic shift element. If we did not use this syntactic sugar, all utterances outgoing a *topic shift* element would need to use the element **if**, as represented in Figure 4.7, potentially cluttering the diagram and hindering a quick understanding that all conditions are the same.

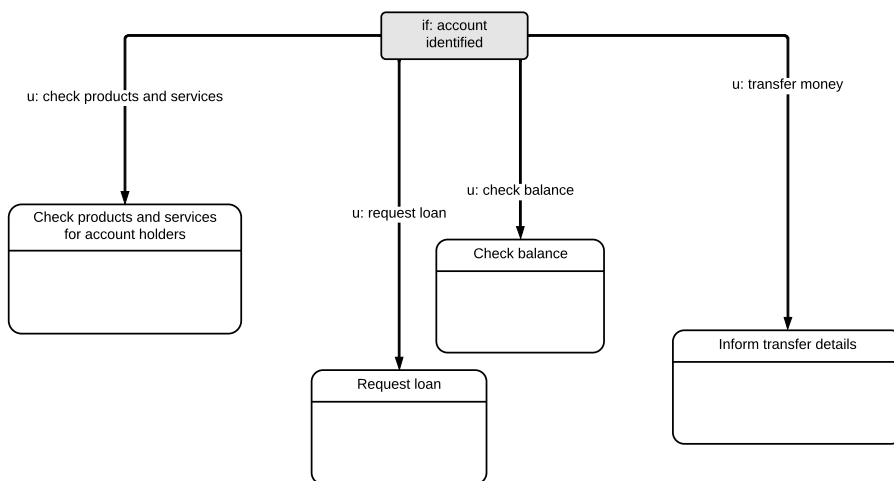


Figure 4.6: Topic shift With an embedded condition

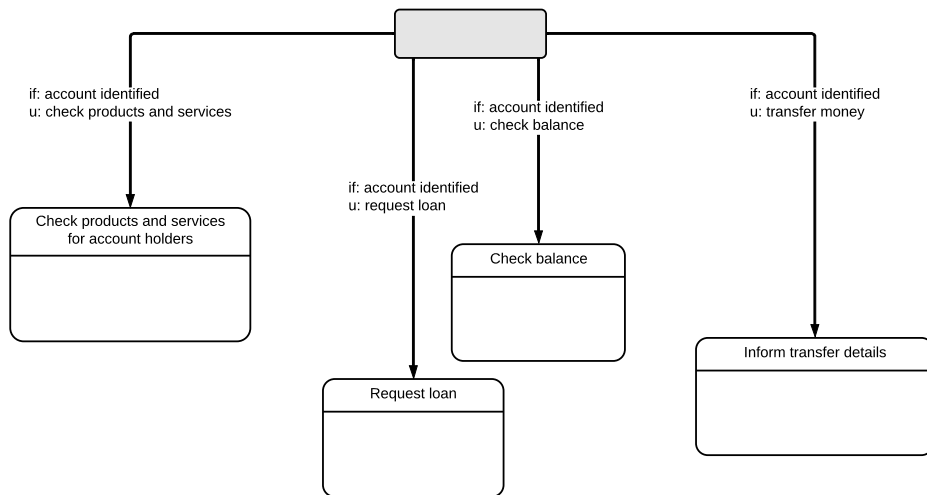


Figure 4.7: Topic shift Without an embedded condition

4.7

Representing Independent and Automatic Events

To represent the scenario where the designer is able to interrupt the user conversation, we created the element **Simultaneous Speech in the User's Turn**, which is exemplified by a system process coupled to a scene. Meaning that the designer will only be able to interrupt the user if they are in that specific scene where the trigger is placed. In the case where the designer wants to speak independently of which conversation is being taken by the user, we have to broaden the coverage of this scenario by using a different representation, which combines a **Global Topic Shift** with a **System Process**, as shown in Figure 4.8a. In this scenario, regardless of the conversation the user is holding, if they keep inactive for 2 minutes, they will be redirected to the login scene.

Another scenario that we cover with this combination is the automatic behaviors that happen seamlessly to the user, such as the auto-save of a document. In this case, the conversation does not deviate from the current topic. Thus, as this is not an interruption, the designer's speech associated with the **When** element, is not represented as a designer's repair utterance, but as a designer's utterance. As pictured in Figure 4.8b, the user will continue in whichever conversation they are in without any interruption.

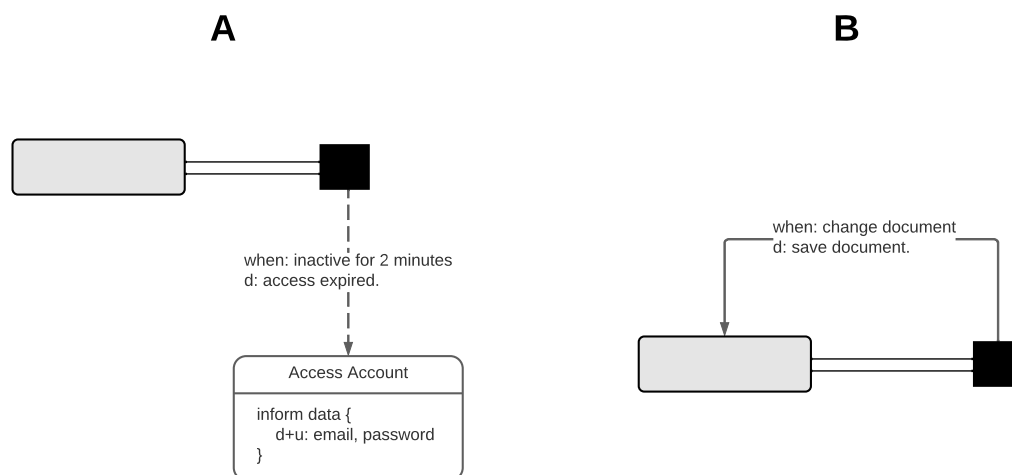


Figure 4.8: Representing Independent and Automatic Events

4.8 Using the Topical Conversation Completion

As stated in subsection 3.3.4, the completion of a goal may be represented by a **topical conversation completion**. However, this may result in poor feedback in cases when a summary of the transaction would be useful or desired by the user. Consider the following example pictured in Figure 4.9a: the user informed a delivery address to receive their card, submitted this information, and was informed to wait. What if the user informed the wrong apartment number?! They have no way of verifying the information given previously. If this verification is deemed important, it would be better if, after submitting the address to receive their card, the designer presented a scene to notify which address was informed, giving the user a chance to check where the card will be sent and make adjustments if necessary, as exemplified in Figure 4.9b. One could say that this feedback could simply be added to the utterance **d: card requested. Please wait for delivery**, detailing the address and other relevant information. It surely could, but it would result in an overload of the designer's turn-yielding utterance. Thus, using a regular scene may be considered in cases where the user would benefit from more detailed and informative feedback. In that case, the notion of *successful completion* could also be represented clearly by a dash in place of the topic as in Figure 4.9c.

4.9 Sequencing System Processes

In some cases, the designer may want to refine the results of a system process to avoid overloading the **if** clauses and the designer's (**d:**) utterances. An example is pictured in Figure 4.10: to request a card, the user must

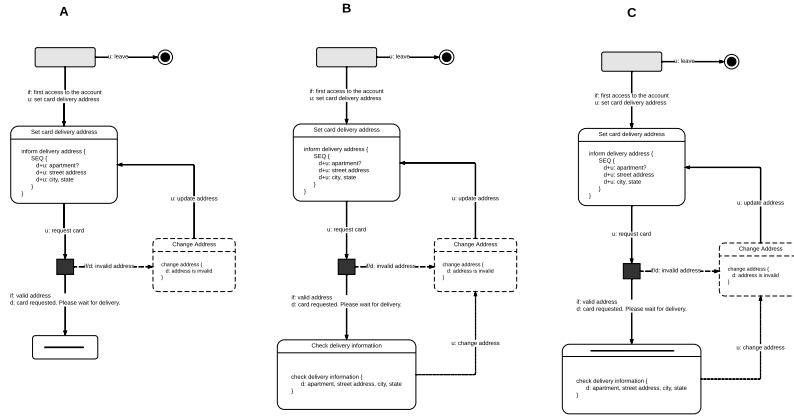


Figure 4.9: Feedback with Topical Conversation Completion *versus* Regular Scene

inform the correct address and have at least R\$10 in the account. Each condition is associated with specific feedback to make clear what went wrong in that conversation. Multiple system processes can be sequenced, allowing the designer to combine each processing result with finer granularity. This combination was proposed by Pessanha and Barbosa (2018) as *Sequential System Processes*. In MoLIC V4, we included it not as an element, but as a usage practice. Note that this concatenation of system processes is valid when the designer wants to communicate about each step.

4.10 Omitting Scenes and Dialogues

Pessanha and Barbosa (2018) suggested creating a representation of a scene that could be omitted from the user in the diagram. This means the user could be able to personalize the system by hiding some functionalities. However, this behavior can be represented using the elements MoLIC V4 already provides. A scene is inaccessible to the user whenever the conditions of an **if** clause are not met. Analyzing Figure 4.7, if the account is not identified, access to the corresponding scenes will be hidden (or disabled) in the system, meaning that none of these topics will be available in the conversation.

4.11 Using Active and Passive Prevention

In subsection 3.7, we described the different breakdown prevention mechanisms. In this section, we exemplify how the active and passive prevention mechanisms can be mapped onto the user interface. As shown in Figure 4.11, the passive prevention allows the user to move forward in the conversation (*e.g.*, the **Proceed** button is enabled) even if the user provides information that

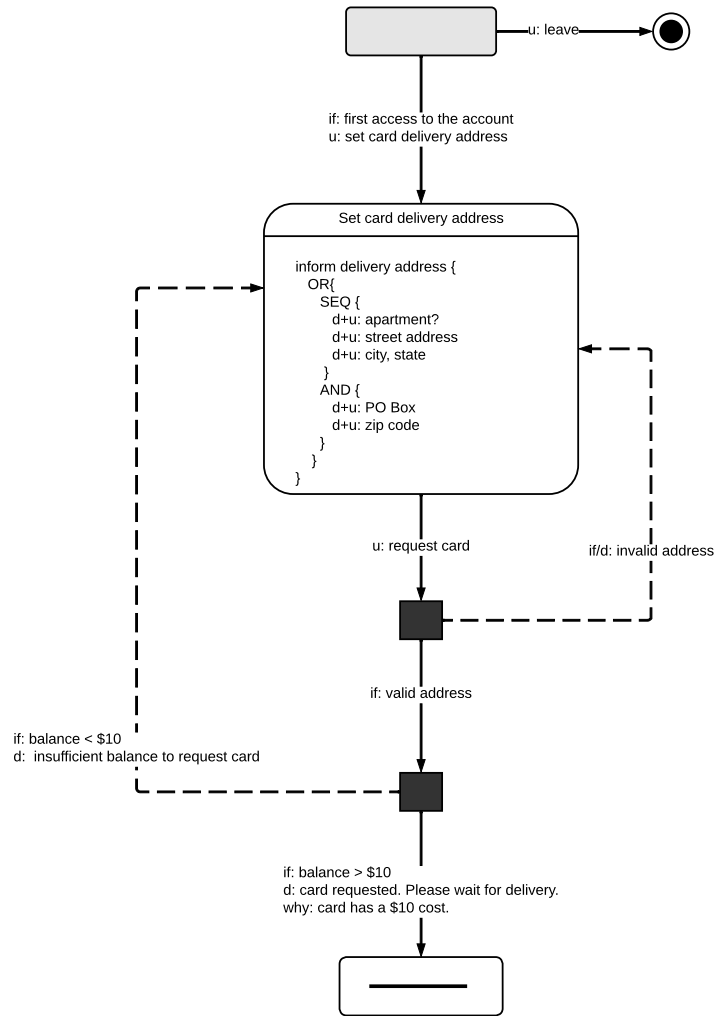


Figure 4.10: MoLIC Diagram Snippet representing sequential system processes

violates the passive prevention (**PP**) condition (*e.g.*, the account information does not meet the required format), which will then result in a breakdown. In the active prevention case, as pictured in Figure 4.12, the user will not be allowed to proceed with the task (*i.e.*, say **proceed**) until they correct the informed sign. At the user interface, this may be reflected as disabling the button **Proceed** until the user informs an account in the correct format.

In most cases, active prevention conditions should be accompanied by a corresponding passive prevention condition (*i.e.*, some communication about the necessary conditions for the conversation to proceed). Otherwise, the user may not know what the active prevention condition is, why they cannot proceed with the conversation, or what they can do to satisfy that condition.

Passive Prevention

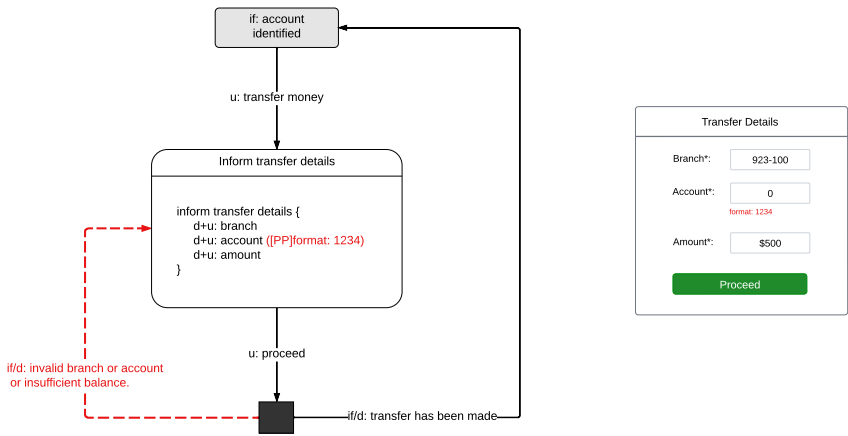


Figure 4.11: Passive Prevention - MoLIC and Mock-up Example

Active Prevention

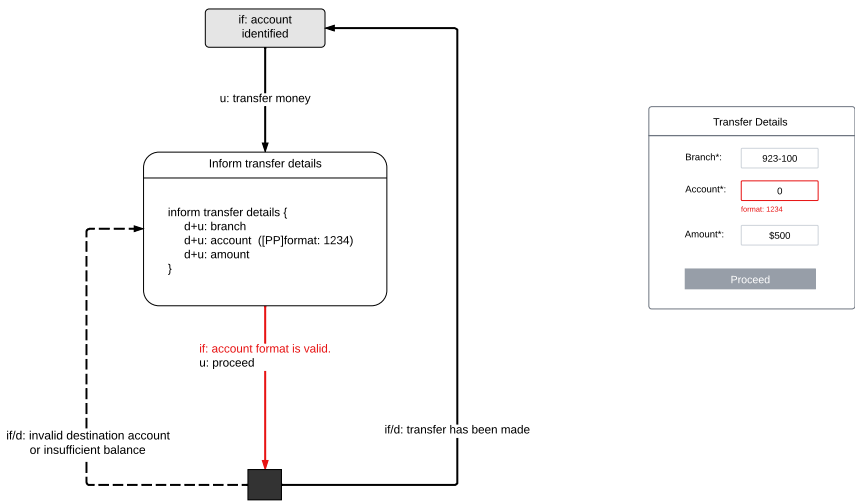


Figure 4.12: Active Prevention - MoLIC and Mock-up Example

4.12

Mapping the Diagram onto the User Interface

One point to highlight when creating the interface based on a MoLIC diagram is the importance of considering not only the scene content and outgoing utterances but also the content of the utterance that leads to the scene. In Figure 4.13, we explore a more detailed mapping of the previous example (Figure 4.12). Note that, in this case, we use the content of the utterance that leads to the scene as an alert message at the user interface.

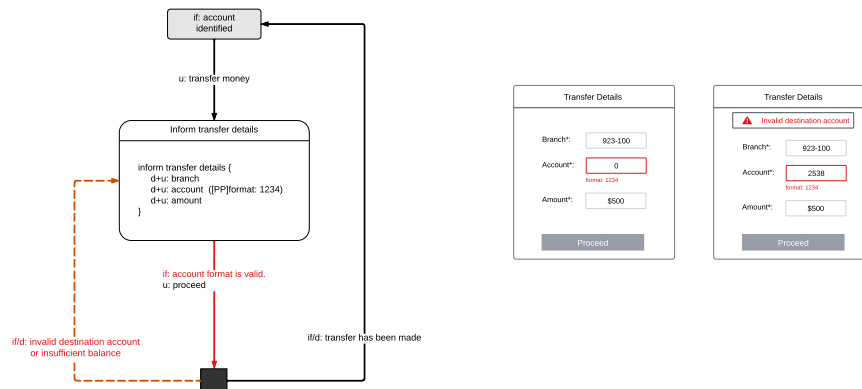


Figure 4.13: Active Prevention - MoLIC and Mock-up Example

4.13

Considerations About MoLIC's Epistemic Character

According to Hoover et al. (1991), the use of models in the design process helps develop a design solution. Design representations such as scenarios and task models do not provide a clear view of the relations between goals or the big picture of the interaction designed (Silva and Barbosa, 2007).

MoLIC interaction diagrams provide an overview of the whole and ease the visualization not only of related goals through the connections between scenes but also of similar and different interaction structures, promoting a consistent design. By encouraging the designer to think about alternative paths and breakdown occurrences and how to recover from them (Silva and Barbosa, 2007), MoLIC also reinforces its epistemic character.

Those differences allow us to understand that different design approaches and representations explore distinct aspects of problem understanding and solutions conceived.

Analyzing related work, we could find citations about MoLIC's epistemic character. de Carvalho et al. (2019) showed that MoLIC allows the designer

to explore alternative solutions to the same problem, thus contributing to the design of a better solution. de Paula et al. (2005) affirms that MoLIC allows professionals from diverse backgrounds to collectively work on a shared understanding of the solution to the problem in question, thus enabling deeper collaboration and reflection.

Also in de Paula et al. (2005), there is an understanding of how the clear visualization of all the interaction possibilities provided in the diagram helps a design or development professional to perform better. From this, we understand that once concluded, the MoLIC diagram still has an epistemic factor, since it eases the comprehension of both the problem and the solution through its representation.

de Souza and Leitão (2009) define an epistemic tool as having two characteristics: (1) it helps the understanding of the problem being solved, and (2) it helps to conceive alternative solutions. Therefore, the epistemic character of MoLIC can be analyzed from those two perspectives. By analyzing an existing diagram or designing and modeling a solution, the reader may understand both the solution and the problem underlying the solution represented in the diagram. Understanding the epistemic character of MoLIC means identifying the occurrences of characteristics (1) and (2) considering both perspectives: analysis of an existing artifact and the creation (or extension) of an artifact during design.

In a brief consideration of the epistemic character of MoLIC, we can identify that some elements help designers understand more about the problem being solved, prompting them to think about the interaction from different perspectives. One example are the *breakdown repair utterances* (subsection 3.4.2), which encourage designers to investigate and reflect on the problems that could occur in the conversation (Silva and Barbosa, 2007), then devise corresponding solutions and repairs. Another element, the *simultaneous speech in the designer's turn* (subsection 3.9.1), encourages the designer to reflect more upon the processes that will occur. By having an entire scene to design, this element influences the designer to think about what could be communicated during that turn. Finally, the *why* element (section 3.16) builds a bridge between the understanding of the problem space and of the solution represented in MoLIC, making it a prime element to support the epistemic nature of the language.

The epistemic character of MoLIC deserves an entire research project to be thoughtfully studied and understood. It involves capturing how each element contributes to the understanding of the problem being solved and provides insights into alternate solutions from both perspectives: conceiving

the diagram and reading the artifact. Ideally, this research should be done through a longitudinal study with a project that tracks the design of a solution over time with various groups of participants.

5 Evaluation of MoLIC Proposals

5.1 Research Procedure

This work has the main goal of realigning MoLIC with the interaction-as-conversation metaphor while consolidating concurrent proposals. With this, we aim to expand the language’s expressiveness without increasing complexity.

After the steps of consolidating concurrent proposals and aligning the elements with the linguistics concepts, we evaluated our proposals to answer the **RQ3: how do users of previous MoLIC versions perceive the changes** and, in cases where we proposed multiple element names deemed interchangeable, consider renaming the element based on the preferred one. In contrast, the elements with a unique name grounded in linguistic concepts were not intended for validation; in such cases, our objective was solely to gather the perceptions of MoLIC users regarding the changes, aiming to clarify the need for future strategies that ensure widespread acceptance of the language’s new version.

We conducted a questionnaire to answer this research question and collected the community’s impressions about the changes proposed to MoLIC. In the next sections, we describe the questionnaire and the analysis procedure. The full questionnaire is available in Appendix D.

Informed Consent We present at the beginning of the form ([see section 1](#)) the informed consent terms indicating that, if the respondent completes the survey, they agree with the terms. The full term can be found in Appendix C.

Respondents’ Profiles In [section 2](#), we gathered information about the respondent’s profiles. We collected the following data:

- Level of knowledge of MoLIC
- Current occupation (student, professor, researcher, or industry professional)
- Field of study
- Level of knowledge of interaction design
- Perception of MoLIC’s usefulness, *i.e.*, how much they think MoLIC contributes to interaction design

- Usage frequency, *i.e.*, with which frequency they use MoLIC in practice or in research

Evaluation of Change Proposals This section comprises 26 sections, the bulk of the questionnaire. It starts with a self-assessment (see section 3) of their knowledge of MoLIC, collecting their level of confidence in reading/interpreting and constructing MoLIC diagrams.

From section 4 onward, the questions are divided into three main groups: renamed elements, newly proposed elements, and modifications beyond the elements' names. In most of the questions, we provided a 7-point Likert scale for respondents to score how much they agreed or disagreed with the change proposal and an open text field for the respondents to justify their score and let us know if they had another idea for the element's proposed name or representation.

In addition, we asked (see section 27) how comfortable the respondents felt about the different uses of the conditions, associated with effects, user's utterances, and designer's utterances.

We also added a question (see section 28) with the intention of analyzing their overall perception about the possible omission of the operand AND, which means that all dialogs of the group must be held (Silva and Barbosa, 2005).

At the end of the questionnaire (see section 29), we asked whether the respondent would be interested in sharing their contact information to participate in future interview studies about MoLIC.

Target Audience In order to take part in the study, respondents should meet two criteria: they should possess at least a basic understanding of both MoLIC and interaction design.

Pilot Test We conducted a pilot test involving two respondents to identify any issues with the questionnaire. The changes we made based on the pilot test were the following:

- We discovered an incorrect answer option in the respondent's field in the questionnaire.
- One of the pilot respondents took an extended period of approximately 40 minutes to complete the questionnaire. To reduce the time without losing very valuable information, we eliminated the questions that inquired about their familiarity with each of the 24 distinct elements illustrated in the diagram presented alongside the question. Instead, we decided to maintain solely the full diagram and two questions. The first question aimed to collect respondents' self-reported familiarity with reading and

interpreting MoLIC diagrams, while the second question focused on their familiarity with creating MoLIC diagrams.

- The removal of the word “transition” from the utterance elements created discomfort in one of the pilot respondents. We returned to the theoretical foundations and proposed including the term “turn-yielding” to replace the term “transition”, which is more aligned with the conversation paradigm. To gather a broader range of perceptions, we proposed two options for consideration: (1) Remove the term “transition” from the utterances, and (2) Replace the term “transition” with “turn-yielding”.

Duration The questionnaire has been open for responses since May 30. On Jun 11, summing up 12 days, we downloaded the responses analyzed in this dissertation.

Disclaimer It is important to clarify that this evaluation method through an online questionnaire has limitations and is not intended to evaluate MoLIC V4, but the perceptions of the MoLIC users of the changes. As stated before, most changes were well grounded in linguistics concepts; thus, not open to informal debate. Our major intent was to understand how the users would perceive these changes to clarify which elements would require more attention in future works based on MoLIC V4.

5.2

Analysis Procedure

In the questionnaire, respondents were prompted with the concept of an element and asked how well (on a Likert scale from 1 (very poorly) to 7 (very well)) that element illustrated the corresponding, concept considering that the element should contribute to the expressiveness of the language without turning the language excessively complex while being aligned with the interaction-as-conversation metaphor. They were also encouraged to explain their score and, if they did not agree with the name or element, provide an alternative (name or element). We analyzed their preferences and highlighted their written responses to assess how well a proposal was well-received.

When we proposed multiple elements or names for a concept, we prioritized the ones grounded in the interaction-as-conversation metaphor. We evaluated the questionnaire’s responses according to the following criteria: (1) if severe problems were identified by the respondents, or by ourselves based on their comments, we discarded the proposal; (2) if the change was grounded in linguistic concepts and/or in the interaction-as-conversation metaphor and did not present any major issue, it was selected to be included in the consolidated

MoLIC V4; and (3) if all or none of the proposals were grounded in linguistic concepts, the comments (and, to a lesser extent, the scores) drove the decision.

5.3 Respondents

The study counted 16 respondents. Of these, 6 were professors and/or researchers, 2 were Doctorate students, 4 were Master's students, 3 were undergraduate students, and 1 was a professional from the industry.

On a Likert scale, from 1 to 7, three respondents reported having little knowledge about MoLIC (ranging from 1 to 3). Five respondents selected the neutral option (4), and eight reported feeling comfortable (with scores ranging from 5 to 6).

Thirteen respondents are from Computer Science. The remaining 3 are from Design.

In terms of interaction design knowledge, thirteen respondents self-reported as in a comfortable range (from 5 to 7), while 2 respondents self-reported as less comfortable.

The respondents were asked about how much they believed MoLIC contributes to interaction design. Thirteen reported scores above 5; two respondents scored low importance of MoLIC for interaction design; and one scored neutral.

Half of the respondents do not use MoLIC in projects (either from work or research). One respondent uses MoLIC in more than 75% of their projects. Two respondents reported using MoLIC in half of their projects. And the other five respondents reported using MoLIC in a maximum of 25% of their projects.

In the auto-evaluation section of the questionnaire, there were two questions: the first asked how respondents felt comfortable reading and interpreting MoLIC diagrams: 13 respondents reported feeling at least comfortable (scores from 5 to 7); two respondents scored neutral; and one reported feeling a little uncomfortable (score 3). In the second question of this section, we asked how they would feel about constructing MoLIC diagrams: eight respondents reported feeling at least comfortable (scores from 5 to 7); four were at least a little uncomfortable (scores 2 and 3); and four respondents scored neutral.

When comparing the respondent's profiles, we observed that there were no significant differences in terms of preference for the previous versions or resistance to changes. Therefore, in this chapter, we will consider the respondent's group as a whole, without making any distinctions.

In the next sections, we present the questionnaire results, including a summary of the answers collected in the questionnaire, an analysis of those

answers, and a discussion of the implications to MoLIC V4.

5.4

Scene-Related Questions

5.4.1

Alert Scene

The *Alert Scene* (see section 5), pictured in Figure 5.1 was added in the textbook by Barbosa and Silva (2010) but it had not been evaluated, so it was not part of an official MoLIC version. Most respondents gave this element positive scores, with 13 (5–7). One respondent gave it a neutral score, and two gave it a negative score (3). Diverse alternative forms were suggested for the element, including a red border, an exclamation mark appended to the element, a double dashed line, or even a completely new element. Three respondents indicated that the dashed border could cause an interpretation of optionality (one respondent correlated it to a task model they were used to). One respondent questioned the concept of the alert, in an attempt to differentiate a blocking alert from a notification alert. Another question raised about the utility of the element was that, if the designer treated the cause of the error before (in the example, verified whether the user had enough balance to withdraw), then no breakdown would occur. Five respondents indicated that the name *alert scene* best fits the element. In contrast, one respondent suggested the term *attention scene*, affirming that an alert could mean something negative or worrisome.

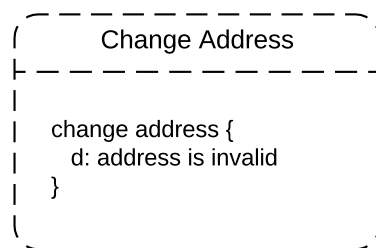


Figure 5.1: Alert Scene

Analysis: The first consideration to avoid confusion is to make clear that this element is not completely new, it already existed in the textbook by Barbosa and Silva (2010). In this work, we added the element to MoLIC V4 and included it in the evaluation, as it had not been evaluated before. This element was well received by the questionnaire respondents. But in its uses, it will be important to ensure that users familiar with the mentioned task model will not confuse the elements (since in that model, the dashed border indicates

an optional task). It is important to clarify that this element is for all kinds of alerts, meaning it is not exclusive to negative alerts that could lead to errors. For instance, an **Alert Scene** can be used to ask for confirmation regarding critical transactions (*e.g.*, transfer of large sums of money), even when that will not lead to an error. As capturing an error before it happens may not always be possible, the **Alert Scene** is also necessary in order to notify the user that a breakdown or unintended consequence could occur if they decide to move forward in the conversation.

5.4.2

End of a Conversation Related to a Specific Goal

To indicate the conclusion of a conversation related to a specific goal (see [section 6](#)), three elements were tested: (i) the designer's monologue; (ii) an empty scene associated with the designer's message in the turn-yielding utterance; and (iii) a scene without topic with dialogs containing only the designer's utterances, indicating the result of the system process. Analyzing the approval range (scores from 5 to 7), the scene without a topic was preferred, with 15 positive scores. The monologue element received 9 positive scores and the empty scene received only 3 positive scores. One respondent argued that the empty scene would stand out from the other scenes in a high-level analysis. One respondent alleged (incorrectly) that the user could only know what the designer says if their utterance is within a scene. Two respondents questioned whether the scene without a topic could have a topic indicating the specific conversation closure. When asked about name suggestions for the concept of conversation closure, the names topical conversation completion and successful completion both received 5 positive scores. When explaining their name choices, many respondents indicated that *successful completion* conveyed an idea of success, which is not always the case.

Analysis: The concurrent proposals involving **Empty Scene**, **Monologue**, and a **Scene without topic** showed that some respondents are still attached to the monologue element. As proposed by Araujo and Barbosa (2008), we will replace the monologue with the empty scene since the same utterance divided between the monologue and the designer's utterance could be represented only in the designer's utterance, simplifying the representation while still standing out in the diagram. Since the **Empty Scene** received ten negative scores (scores ranging from 1 to 3), there will be a need to promote this element as a substitute for the monologue in order to ensure its adequate usage.

Considering the great acceptance of the **Scene without topic** and

because it reuses an existing element without creating more complexity, we are adding this concept to MoLIC. The *scene without topic* is represented as in Figure 5.2. Note that this is not an entirely new element, it is a *Scene* with a dash for a topic, to indicate that the topic did not change from the previous conversational exchange. Thus, it is up to the designer to decide which element fits best in the conceived solution. Regarding the new element proposed (as depicted in Figure 5.3), three name proposals were evaluated: *empty scene*, *successful completion*, and *topical conversation completion*. Although the term *successful completion* received the most positive scores, the respondents strongly related their selection to the fact that it indicates a conversation that ended successfully, which should not be assumed. If we consider the scenario where a file to be loaded is corrupted, for instance, the idea that “there is nothing more the user can do toward their goal” still applies, even though it is not a successful case. Thus, to avoid any relation to success, and considering the respondent’s preferences, who also preferred the term *topical conversation completion*, we decided to adopt it instead of the term *empty scene*, which received no positive scores.

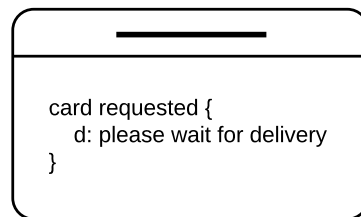


Figure 5.2: Scene without topic

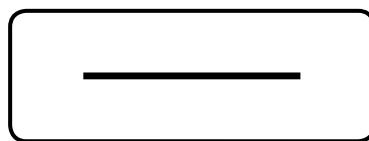


Figure 5.3: Topical Conversation Completion

5.4.3

Required vs Optional Signs

Pessanha and Barbosa (2018) suggested indicating only the optional signs using a question mark. We could extract from the questionnaire (see [section 7](#)) a bigger preference for using asterisks to indicate only the required signs (with 16 responses ranging in scores between 6 and 7), while the alternative to

indicate only the optional signs received only 3 positive scores and 12 negative scores. The option of not indicating required or optional signs received 14 negative scores. The majority of respondents defended the asterisks either because it was used before in MoLIC or because it is most widely used at the user interface. Another respondent argued that it is more important to mark what is required instead of what is optional.

Analysis: When deciding between representing only the **optional signs** or only the **required signs** of dialogue, the vast majority of the questionnaire’s respondents preferred to keep indicating only the required signs with asterisks (either for the tradition of old versions or familiarity with forms). To defend this change, we will reinforce the fact that usually the diagrams are elaborated without this indication, then, it is better to assume that all signs are required and mark only the optional with the question mark as depicted in Figure 5.4. We cannot allow both representations to coexist in MoLIC due to the ambiguity that would be created if nothing was indicated in the dialogue: would it mean that all signs are mandatory or optional?

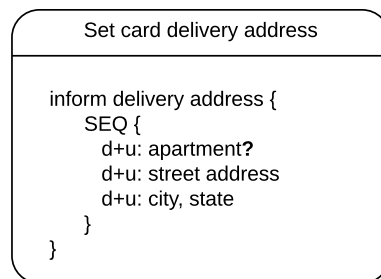


Figure 5.4: Marking only the Optional Signs

5.4.4 Dialog Format

We extracted from the questionnaire ([see section 8](#)) that the alternative with the emitter prefix at the level of signs as exemplified in Figure 5.5 was best received by the questionnaire’s respondents. This alternative received 11 positive scores and 2 negative scores, while the concurrent proposal, to use the prefix at the dialogue’s level, received fewer positive scores (9) and more negative scores (5). When asked to explain their decisions, one respondent argued that the emitter prefix should be at the dialogue level since the conversation is at a higher level, meaning it is about the dialogue, not the signs. Four respondents claimed that using the emitter prefix at the level of signs is better because it allows the designer to separate what sign is emitted by themselves and what sign is expected from the user.

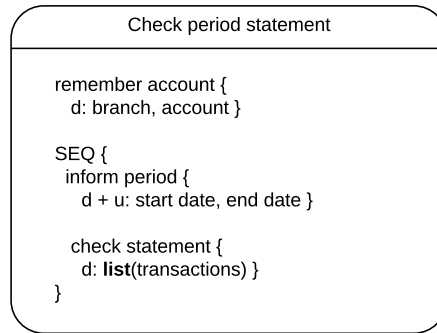


Figure 5.5: Emitters next to the dialogue’s signs and the list element

Analysis: In terms of dialog format, there were two alternative proposals: indicating the emitter in the level of the dialogue or in the sign. Even though a respondent pointed out that the conversation is about the dialogue to defend the prefix being placed with the dialog, the ability to differentiate what sign is emitted only by the designer or by the user when asked by the designer, is an argument solid enough to decide for indicating the emitter at the sign level.

5.4.5

Sign with Multiple Values

Respondents were asked (see section 9) about the element they preferred to represent a range of values that could be associated with a sign. The term “list” was preferred, with 12 positive scores and only 1 negative score. The alternative “options” received 7 positive scores and 4 negative scores. Finally, the alternative that suggested the use of both “set” and “list” received 4 positive scores and 7 negative scores. When asked to explain their decisions, the respondents declared that it is better to have only one element’s name and allow defining the ordering when desired. One respondent defended the idea of keeping MoLIC simple by not defining the type of the sign (*i.e.*, a string, a list, a number, etc). Another respondent raised a concern that, with more different denominations, the representation would be more complex and the language comprehensibility might be harmed.

Analysis: The ability to represent signs that can be associated with multiple values has existed since the first version of MoLIC. Araujo and Barbosa (2008), intending to complement the interaction diagram, proposed to include in each dialog the signs and their corresponding types (plain text, single choice, multiple choice, etc). Thus, this element does make MoLIC more complex, since it is not new. When deciding which name is better to

represent this concept and element, the respondents preferred the term “list”. Even though this term is not aligned with the interaction-as-conversation metaphor, we decided to use the name *list*. The main reason, identified after the questionnaire was executed, is because the term “response options” is only adequate to inform about signs that are in the form **d+u:** (since it is used to represent the user’s response to the designer’s offered options). However, there is also a need to represent lists of signs that will be emitted by only the designer (**d:**), without expecting a response, as represented in Figure 5.5 with the dialogue check statement `{d: list(transactions)}` and further specified in subsection 3.3.2.

5.4.6

Selection of a Sign with Multiple Values

The element **pick** (see section 10) was proposed to allow the designer to indicate the number of elements that can be selected from a sign that has a range of values. From the responses, we noticed a good reception of the element, with 11 positive scores and only 1 negative score. Two respondents were concerned about adding user interface elements to MoLIC, preferring to keep it more abstract and fluid. One suggestion was to add this as a textual complement to the representation. The term “select” was proposed twice to replace the term “pick”.

Analysis: Pessanha and Barbosa (2018) suggested the inclusion of cardinality, through the prefix **pick**, to indicate the number of values that should be selected in a sign that has response options. In accordance with the idea to keep MoLIC more abstract and fluid, we understood that this term is not well aligned with the conversation metaphor and could bring more complexity to MoLIC. Despite the term *pick* limitations, in response to the expressed desire identified in the questionnaire to incorporate this concept, we decided to add it with the term *choose*, as represented in Figure 5.6. It is a commonly used term in informal conversations, hence better aligned with the interaction-as-conversation metaphor than the term *pick*. Since this element was renamed after all the steps of grounding in the theory and questionnaire study, we strongly recommend submitting it for evaluation in future works.

5.4.7

Interpretation of a Scene Dialog without Grouping Operand

The respondents were presented (see section 28) with a scene containing three dialogues with no explicit grouping operand. Three possible interpretations were presented and the respondents were expected to score how much they

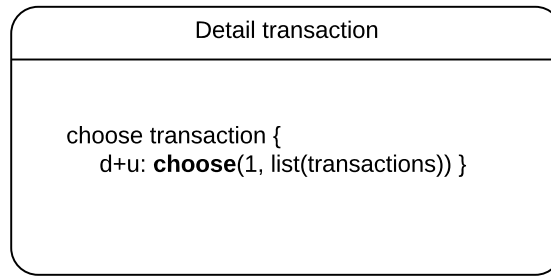


Figure 5.6: Choose among a list of signs

agreed with each. The first option, “all dialogues must be undertaken”, received 6 positive scores and 4 negative scores. The second option, “no dialogue must be undertaken” received 10 negative scores and 1 positive score. The last option, “some dialogues should be undertaken”, received 4 positive scores and 5 negative scores. In the written responses, one respondent indicated that the dialogue emitted solely by the designer was not required. Another respondent expressed that if one of the dialogues were not to be undertaken, it would have to be associated with a condition. In a similar approach, another response suggested indicating with an asterisk the required dialogues. Another respondent affirmed that all dialogues should occur in sequence. One respondent indicated that this interpretation would depend on the level of detail that the designer intends to give and also who is consuming the model and what their objective is.

Analysis: To understand the respondents’ awareness of the grouping operand **AND** and the possibility of hiding it as proposed by Silva and Barbosa (2005) and depicted in Figure 5.7, we presented a scene with three dialogues and asked for their agreement with three different interpretations. Two of those alternatives received the most positive scores: “All dialogs must occur” and “One of the dialogues must occur”, showing there is a need for better specification of the grouping operands. To reinforce our perception, one respondent described that all dialogues should occur in sequence, which is represented by another grouping operand: **SEQ**. To allow the designer to indicate that a dialogue of an explicit or implicit group (marked with AND) is optional, we are extending the concept of optional signs to a dialogue. Thus, the dialogue **save contact for future transactions** in a money transaction scene, could be represented as **save contact for future transactions?** d+u: confirm to indicate it is optional, for instance.

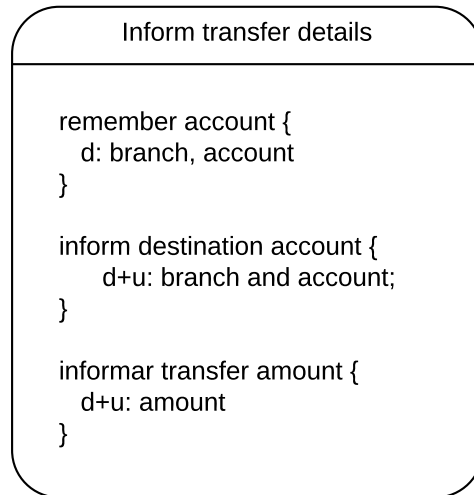


Figure 5.7: Implicit form of the grouping operand AND

5.5

Questions Related to Turn-Yielding Elements

The analysis of the turn-yielding elements will be made jointly at the end of this section.

5.5.1

Rename - User's Transition Utterance

To realign the element depicted in Figure 5.8 with the interaction-as-conversation metaphor, it was necessary to eliminate the term “transition”. We suggested (see section 14) two alternative new names: *user's utterance* and *user's turn-yielding utterance*. The name *user's utterance* received 10 positive and 4 negative scores, while the alternative name *user's turn-yielding utterance* received 4 positive and 7 negative scores. In the textual responses, one concern 3 respondents shared was the confusion that could occur between the utterances of a scene and the utterances of an exchange of turns when removing the word “transition”. Two respondents complained about the element's name size.



Figure 5.8: User's (Turn-Yielding) Utterance

5.5.2

Rename - Designer's Transition Utterance

The utterance emitted by the designer after a successful system process (depicted in Figure 5.9) was previously called *Designer's Transition Utterance*. To better align this element (see section 20) with the interaction-as-conversation metaphor, we proposed two new names: *designer's utterance* and *designer's turn-yielding utterance*. The first option, a simplified version of the original name, received 10 positive scores and 2 negative scores, whereas the version that adds the term *turn-yielding* to the name was mostly rejected (9 negative scores and 4 positive scores). The same concern as with the user's utterance was raised here: the lack of differentiation between the utterance within a scene and the utterance associated with a turn shift. No alternative names were suggested.

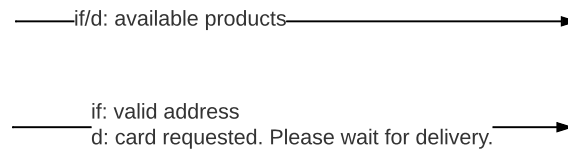


Figure 5.9: Designer's (Turn-Yielding) Utterance

5.5.3

Name Proposals - User's Repair Utterance

This element (see section 15), depicted in Figure 5.10, has existed since the first version of MoLIC but did not have a name defined for it. Since this utterance is related to a repair in the conversation, two names were proposed: *user's repair utterance* and *user's (turn-yielding) repair utterance*. The simplified version of the name received 12 positive scores and 2 negative scores. In contrast, the name with the *turn-yielding* term received 5 positive and 9 negative scores. In the written responses, one respondent alleged that the term *repair* would make sense if something went wrong. Five respondents found that the term *turn-yielding* would add too much complexity to the element's name.

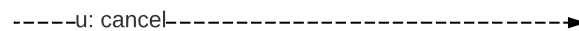


Figure 5.10: User's Repair Utterance

5.5.4

Rename - Breakdown Transition Repair Utterance

This element, depicted in Figure 5.11 (see section 19), is intended to represent breakdown repairs that may occur at interaction time. Since the term “transition” was not closely related to the metaphor of interaction as a conversation, the following names were proposed: *Designer’s (Breakdown) Repair Utterance* and *Designer’s (Turn-Yielding) (Breakdown) Repair Utterance*. The first option received 12 positive scores and 2 negative scores, against only 3 positive scores and 9 negative scores to the addition of the *turn-yielding* term. Two respondents indicated that *turn-yielding* would add more complexity. Three variations of the name were proposed in the written responses: *designer’s repair utterance*, *designer’s recovery utterance*, and *breakdown repair utterance*.

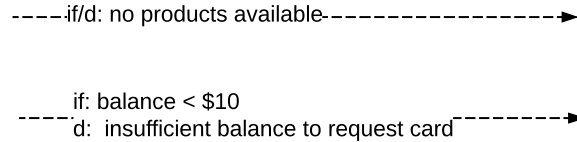


Figure 5.11: Designer’s (Turn-Yielding) (Breakdown) Repair Utterance

5.5.5

Silent Turn-Yielding

To address the possibility of the designer simply giving the turn to the user without any related utterance, we proposed (see section 21) to turn this representation into an element called *Silent Turn-Yielding* as represented in Figure 5.12 with only a condition that must be satisfied to proceed with the conversation. The element was well received by 10 respondents, while 4 did not agree with the inclusion. In the written responses, 2 respondents praised the element’s addition. One respondent raised a question as to whether this element was really silent since it would take the user to “another system page”. An alternative name proposed was *successful processing*.

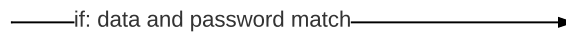


Figure 5.12: Silent Turn Yielding

Joint Analysis of Turn-Yielding Elements: When addressing the utterances, it is important to clarify that the term “transition” was not

appropriate, since it is associated by Hutchby and Wooffitt (1988) as something that occurs between turns. In Wooffitt (2005), the transition is seen as part of a turn that leads the conversation to an exchange of turns. Thus the word transition cannot be used to represent the whole turn. An example is: after stating something, the emitter asks the receiver a question in an attempt to exchange the turn: I would like a glass of water, and you? The excerpt “and you?” is the transition part of a turn. The equivalent representation in MoLIC might be viewed as the *arrowhead* of the arrow. Furthermore, as explained in section 3.4, the term “turn-yielding” is used in Linguistics to represent the cases where the speaker gives the turn spontaneously and, in some cases, may even appeal for a response (Stenström, 1994). Thus, from MoLIC’s perspective, a turn-yielding occurs when an exchange of turns occurs between the user and designer and vice-versa. To address the removal of the word “transition” from the element’s name, respondents were prompted with two options: (1) remove “transition” and (2) replace “transition” with “turn-yielding”. The term “turn-yielding” received several criticisms; we associated this resistance with the fact that it is an uncommon word for non-native English speakers (as was the questionnaire’s participants’ case). Thus we decided to keep this term since it allows differentiating utterances emitted inside a scene from utterances emitted between scenes and other MoLIC elements (system process, etc.). One thing to elucidate is that this change has no effect on the visual representation in the diagram, only on the concept’s name. To promote the usage of this term, it will be necessary to carefully revise MoLIC’s instructional material to deepen learners’ understanding and acceptance of terms that are more aligned with the interaction-as-conversation metaphor, even if unconventional or unfamiliar to them.

Before discussing the term *repair*, it is important to explain the modification made to the element that allows the user to amend their statements. This element has existed since the first version of MoLIC, but it was treated simply as *User’s Utterance*, albeit represented by a dashed line pattern. However, it has a very specific use: to return to a previous conversation to amend it. Thus, to better represent this concept, we decided to give it a name: ***User’s Repair Utterance***. With two elements related to adjustments in the conversation, we decided to standardize the terms by renaming the element ***Breakdown Transition Recovery Utterance*** to ***Designer’s (Turn-Yielding) (Breakdown) Repair Utterance***. With this change, it is clearer that the emitter is the designer, and that they are repairing a breakdown. The breakdown term differentiates it further from the ***User’s Repair Utterance***, which represents a self-repair (to their own previous statements), not a recovery from

a breakdown. To clarify a question raised during the questionnaire, the term repair is not exclusive to “when things go wrong”. A repair may be initiated from a scenario without apparent errors (Sacks, 2010). As stated in Schegloff et al. (1977), “...we will refer to ‘repair’ rather than ‘correction’ in order to capture the more general domain of occurrences...In view of the point about repair being initiated with no apparent error, it appears that nothing is, in principle, excludable from the class ‘repairable’”. Thus, the term *repair* can also be used to represent a reconsideration.

Differently from the previously presented utterances, the term turn-yielding is not adequate for the element ***User’s Repair Utterance***, since the turn remains with the user after this utterance is emitted; thus, there is no exchange of turns caused by this element.

In cases where the designer wants to give their turn to the user without any utterance, this would be represented by an arrow containing only a condition that, when met, allows proceeding with the conversation. We named this element ***Silent Turn Yielding*** with the intent of shedding light on this interaction possibility. It is not a completely new element, we just named what could already be represented in MoLIC to legitimate it and call attention to the possibilities it brings. In response to questioning the fact that the element is effectively silent since the user could be taken to “another system page”, it is important to clarify that the “another system page” is represented by the target scene, whose topic will be presented when the user “arrives” there.

Thus, the final terms for these utterances are: ***User’s (Turn-Yielding) Utterance***, ***Designer’s (Turn-Yielding) Utterance***, ***User’s Repair Utterance***, ***Designer’s (Turn-Yielding) (Breakdown) Repair Utterance***, and ***Silent Turn Yielding***. If the designer understands that the omission of the terms *turn-yielding* and *breakdown* represented here within parentheses will not cause ambiguity or misunderstanding, they may use the reduced version of this expression.

5.6

Questions Related to Specific Elements

5.6.1

Different Ways to Represent the Status of an Ongoing System Process

Two alternative elements were presented in the questionnaire (see section 11): (a) a scene coupled to a system process; and (b) a system process with a white rectangle with a single sign inside. Alternative (a) received 10 positive scores and 3 negative scores, while alternative (b) received 8 positive scores and 4

negative scores. There was a third option, (c), proposing the coexistence of both elements, receiving 7 positive scores and 6 negative scores. When asked to explain their scores, the respondents pointed out the following: alternative (a) is more expressive and allows talking about one sign or more (meaning (b) would not be needed); alternative (b) could be maintained for the shorthand case. One respondent questioned whether there could be parallel conversations from the coupled scene about other topics.

Analysis: To represent the progress of a system process, two concurrent proposals were evaluated: a scene with a system process, which uses two elements that already exist, adding only a connection between them; and a structure created with the limited purpose of representing a single sign related to the system processing status. In our understanding, this element has two major problems: first, what it represents can already be represented by the connection of existing elements; and second, when it was proposed in MoLIC V2 Silva and Barbosa (2005), its creation was based on the idea that “no dialogue about the ongoing process would occur between user and designer’s deputy”, which was not very precise because, as the user is able to cancel the processing at any moment, we can conclude that there is indeed a dialogue. Therefore, we maintain our original idea of removing this element and keeping only the more expressive and better-aligned element: ***Simultaneous Speech in the Designer’s Turn***, represented in Figure 5.13.

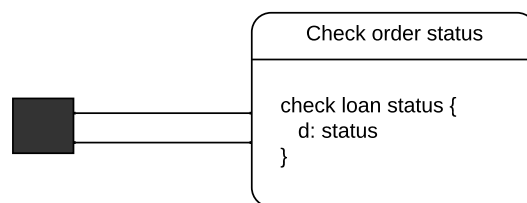


Figure 5.13: Simultaneous Speech in the Designer’s Turn

5.6.2

Name Proposals - User Interruption in the Designer’s Turn

The element presented in the previous question – system process connected to a full-fledged scene – was proposed in MoLIC V2 but had not received an official name. It was treated as a scene coupling or synchronous scene. To choose a name for this element, we proposed two alternatives (see section 12) aligned with the conversational perspective: *Overlap in the Designer’s Turn*, which received 4 positive scores and 6 negative scores, and *Simultaneous Speech*

in the *Designer's Turn*, which received 6 positive scores and 4 negative scores. When asked to explain their scores, two respondents indicated that the term overlap gives an idea of interruption, while simultaneous speech associates with both speaking simultaneously. One respondent associated the term overlap with something that went wrong and another complained about the element's name size. Regarding alternative names suggested by the respondents, seven different names were proposed, including interrupt designer's turn, *user speech in designer's turn*, and *speech overlap*.

Analysis Even though the term *overlap* can be used interchangeably with the term *simultaneous speech* Laskowski et al. (2012), overlap seems to have a negative connotation of interruption. Since the purpose of this element is to allow both the designer and user to *talk*, we would rather avoid a biased perception of interruption. Thus, we will assign the name ***Simultaneous Speech in the Designer's Turn*** to the element represented by a scene connected to a system process, as depicted in Figure 5.13.

5.6.3

Designer Interruption in the User's Turn

This element (see section 13) was proposed in MoLIC V4 to allow the designer to interrupt the user when a predefined event occurred during the conversation. It received 13 positive scores and 2 negative scores, which indicates significant approval. When choosing the name, both options received the same number of positive scores (5) and almost the same number of negative scores for *Overlap in the User's Turn* (6) and *Simultaneous Speech in the User's Turn* (5), respectively. In the textual feedback, two respondents commented on the element but not the name: one respondent said this element would be convenient and another asked for a more explicit difference between the two overlap cases (user in the designer's turn and designer in the user's turn). Regarding the name, one respondent complained about the element's name size. Other names proposed by the respondents were: *designer override*, *user speech overlap*, and *design speech in the user's turn*.

Analysis: As previously explained, to address the event of the designer interrupting the user's turn, we created a new concept that allows the association of a designer action with the occurrence of an event. From the survey results, we consolidated our choice to name it ***Simultaneous Speech in the User's Turn***, as depicted in Figure 5.14.

Note that, as the diagram makes clear whether it is the user's turn or the designer's turn, we can adopt a collective name *Simultaneous Speech*, having in mind that the representations are different and the designer will only be able

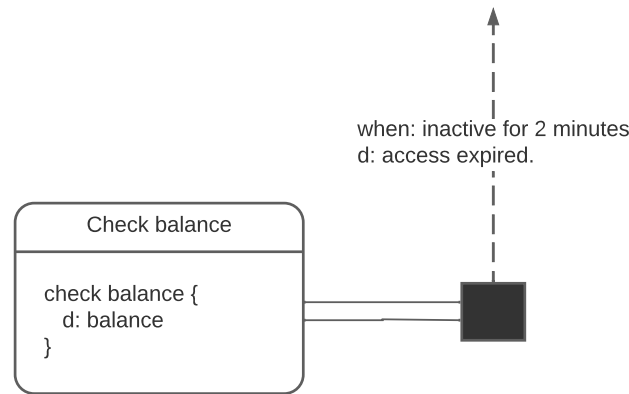


Figure 5.14: Simultaneous Speech in the User's Turn

to interrupt the user via the utterances previously assigned with the element *when*.

5.6.4

Rename - Ubiquitous Access

To realign MoLIC with the metaphor of interaction as a conversation, the element *ubiquitous access* (see section 18) had to be renamed. The proposed name was *topic shift*. This proposal received 9 positive scores and 4 negative scores. Four respondents also expressed textually their agreement with the new name, while two respondents were against this rename arguing that it might be viewed as the end of a conversation, not necessarily a change in the conversation topic. Another respondent compared the turn yielding between scenes as also being a topic shift and reinforced that the term *ubiquitous* expresses that “at any moment” the indicated topics could be initiated. The alternative names proposed were: global topic shift, anywhere topic shift, ubiquitous topic, ubiquitous point, and ubiquitous shift.

Analysis: The term *Ubiquitous Access* was proposed in MoLIC V1. Pessanha and Barbosa (2018) raised an important question about this terminology: from a conversation perspective, the act of changing from one scene to another indicates a change in the *topic* of the conversation. Moreover, they explained that because this change can be started at any moment, a more suitable name would be *Global Topic Change*. When aligning the term suggestion to the conversation metaphor, we identified the concept of *topic shift* (explained in section 3.10). After the study, we noticed some respondents seemed attached to the term *ubiquitous*, most likely because of their previous experiences with MoLIC. Against the expression *topic shift*, as raised by a questionnaire’s re-

spondent, is that we cannot differentiate a turn-yielding utterance that causes a topic shift (*i.e.*, an utterance from one scene to another scene) from a topic shift that occurs through the specific topic shift element. Thus, we decided to rename the element ubiquitous access to ***Global Topic Shift***, a minor adjustment from Pessanha and Barbosa (2018)’s proposal (from *change* to *shift*), yet better aligned with the corresponding linguistic concept. Its representation was kept unchanged, as depicted in Figure 5.15.

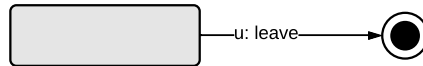


Figure 5.15: Global Topic Shift

5.6.5

Rename - Conditional in a Conversation

Three names were investigated in this question (see section 22): *pre*, *if*, and *precond*. The term *pre* received 6 positive scores and 3 negative scores; the term *precond* received 9 positive scores and 2 negative scores; the term *if* received more positive scores (10) and more negative scores (4). In the textual remarks, three respondents reinforced their preference for the term *if*, considering a stronger familiarity with the term.

Analysis: The element used to express a condition that has to be met in order to proceed with the conversation was proposed in MoLIC V1 (Barbosa and de Paula, 2003a) as a precondition, along with its minimal expression form: *pre*. Araujo and Barbosa (2008) proposed changing this element to *precond* to avoid confusion, because they introduced an element named *presup*, which shared the *pre* prefix. Another change proposal was made by Pessanha and Barbosa (2018), suggesting the use of the well-known term *if*. Returning to the linguistics concepts, we could relate the term *if* with conditional clauses, as explained in section 3.11, but not exactly *preconditions*, which is a term more strongly related to computing. In the questionnaire, the term *if* received positive comments and was considered more familiar. One respondent alleged the preference for the previous form because it was already in use, even though they were explicitly asked to consider only the fit with the interaction-as-conversation metaphor and leave their familiarity with the previous versions of MoLIC aside. As our main concern here is to achieve better alignment

of MoLIC V4 with the interaction-as-conversation metaphor, we decided to rename the element to *if* (depicted in Figure 5.16). As with all changes in concept nomenclature and diagram notation, a robust instructional material will be needed to assist future usage of MoLIC V4.

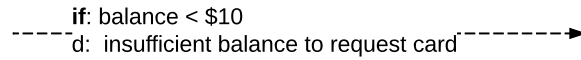


Figure 5.16: If

5.6.6

Rename - Consequence in a Conversation

To represent the consequences of a conversation (see section 23), three names were compared: *then*, *perl*, and *effect*. The term *then* received the most positive scores (12); *effect* received 6 positive scores and 6 negative scores; and *perl* received 1 positive score and 10 negative scores. In the written responses, four respondents defended the pair “if-then”. One respondent indicated that *perl* refers to a difficult concept to understand. No alternative names were proposed.

Analysis: To represent that an utterance has an effect in the conversation context, several names were proposed over the years: *post-conditions*, *perl*, and *perloc*. Although the expression *perlocutionary effect* is an appropriate linguistic concept, the qualifier *perlocutionary* is unfamiliar to most MoLIC users. This prompted us to adopt the term *effect*. As the term *then* is used in conversation to express effects, we also included it in the questionnaire. From the responses, we noticed a strong agreement with the term *then* because of its strong association with the term *if*, both in computer science and everyday conversations. The problem raised with this pair is exemplified by responses such as “if the condition is expressed by *if*, then the effect has to be expressed by *then*. However, in MoLIC, an effect is not always related to a condition; it may simply occur as a result of a user’s utterance, for instance. To prevent misunderstandings stemming from this undue association, which may lead to the assumption that these elements should always be used as a pair, we decided to rename the conditional element to **effect** (as represented in Figure 5.17), which is both simple to understand and well-aligned with the conversation metaphor.

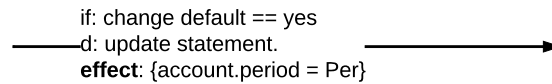


Figure 5.17: Effect

5.6.7

Let - Context Initial Value

We suggested (see section 26) the inclusion of a new element that allows the designer to specify initial values for a context feature. The conceptual element received 7 positive scores and 2 negative scores. The respondents evaluated the proposed name *let* with 6 positive scores and 4 negative scores. In the written responses, one respondent agreed with the term’s name, since it is associated with computer science. Another respondent asked whether it would be *d* or *d+u*. One respondent questioned the use of brackets in the usage of the term and asked whether it was related to the definition of multiple default values. The following alternative names were proposed: *set*, *preset*, *define*, and *default*. However, none of these names was well aligned with the conversation metaphor. Therefore, the notation **let** was adopted in MoLIC V4, as exemplified in Figure 5.18.

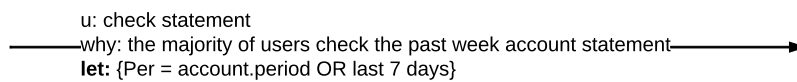


Figure 5.18: Let

Analysis: Continuing on the changes related to the conversation context, the element ***Let*** was proposed to allow the specification of an initial value for a ***Context Feature***. With this element, we expect to allow the designer to define values for an interaction, either by default or considering a previous change to the context by means of an ***effect*** element. After analyzing the questionnaire responses, we saw this could be a source of confusion that requires clarification when presenting the element.

To accommodate a default value to be overridden by the interaction, we suggest the following use: `let: {zoom = user.zoom OR default.zoom}`. This means that, at first, the zoom level will be the default (as the `user.zoom` value would be null). If the user changes the zoom level, thereby assigning a value to `user.zoom`, future interactions will consider their preference over the default value. Note that the attachment of the zoom level to the user is a design decision. The designer could have just as well decided on attaching the zoom

level to the current session, using a `session.zoom` sign instead. In this case, when the user enters the system next time, the zoom level would be back to the default.

5.6.8

Definition of the Conversational Context

We proposed (see section 25) to incorporate in MoLIC V4 a concept suggested by Araujo and Barbosa (2008) that allows the designer to define the conversation context with a second level of granularity. The element received 8 positive scores and 4 negative scores. In terms of alternative names, the options were: *articulated sign* (as proposed by Araujo and Barbosa (2008)), which received 2 positive scores and 4 negative scores, and *context feature* (better aligned with the conversation metaphor), with 7 positive scores and 3 negative scores. In the open comments, three respondents expressed concerns about defining what they called “variables” in MoLIC and questioned the necessity of this definition in the diagram. Another respondent said this addition could add more complexity to the model and questioned the difference between the *context sign* and this new element. Regarding alternative names, the following were proposed: element change, variable change, and subtopic.

Analysis: We evaluated in the questionnaire a concept proposed by Araujo and Barbosa (2008) to specify the conversation context with finer granularity. Araujo and Barbosa (2008) proposed this element with the name *Articulated Signs* and, based on the literature, we suggested the alternative name *Context Features*. In the questionnaire responses, four respondents gave negative responses, either questioning whether we were starting to define variables in MoLIC or the real need for this level of specification. After critical analysis of the questionnaire’s responses, we understood that adding a new concept just to differentiate a context definition of one or two levels was adding unnecessary complexity to the language. Thus, we decided to unify the concepts and name them *context features* while keeping the idea of defining as many levels as the designer wants (since it allows a more flexible specification of contexts and business rules that may affect the interaction) without defining a new element. These elements allow the designer to associate a number of concepts with the conversation context. Examples of context features are `user`, `account`, `(account+user).statement-period`, `document`, `document.template`, `network`, `network.default-printer`, `network.default-printer.default-color-scheme`. The only changes we propose are renaming the *context sign* to ***Context Feature*** to align it with the corresponding linguistic concept better, and the ability to refine the defined context.

Examples of context features:

- account
- account.withdrawal-limit
- (account+user).statement-period

5.6.9**Rename - Presup**

The term *presup* (short for *presupposition*) (see section 24) was used in MoLIC to indicate the reasoning associated with a design decision. With this element, the designer can leave a trace in the diagram of the decisions taken before. In MoLIC V4 we propose to rename it to *why*. Eight respondents agreed with the change, while four disagreed and two were neutral. Three respondents added in a written response that they preferred the term *why* for its simplicity. In contrast, three respondents indicated that the element *presup* was used to indicate a non-confirmed hypothesis while the term *why* could imply something more certain as a confirmed reason. An alternative name proposed was *because*.

Analysis: Araujo and Barbosa (2008) proposed the element ***presup*** to record in the diagram the reasons associated with the proposal of a specific turn-yielding utterance when conceiving certain parts of the solution. Later, Pessanha and Barbosa (2018) proposed to rename it to *justify* since the element's purpose was to justify the designer's deputy's purpose related to the existence of a turn-yielding element. Since it represents a trace of the design rationale by referencing it and recording (in the diagram) why certain elements were put in the diagram, we proposed to rename this element to *why*. Three respondents wrote that they preferred the term *why* for its simplicity. In response to a counterargument raised by one respondent, the element ***why*** is not only limited to hypotheses but can also be used to record well-grounded findings from the user and domain research and well-argued decisions typically recorded in a design rationale documentation. Another change we proposed is to allow in MoLIC V4 designers to also record the reasons or motivation for other elements besides turn-yielding elements (*e.g.*, dialogues). These two different uses of the element *why* are depicted in Figures 5.19, and 5.20.

5.6.10**Preferred Conversation**

This element (see section 16) was proposed by Araujo and Barbosa (2008) but had not been evaluated before, so it was not part of MoLIC's current version. Based on the questionnaire's responses, the concept was well received: 13

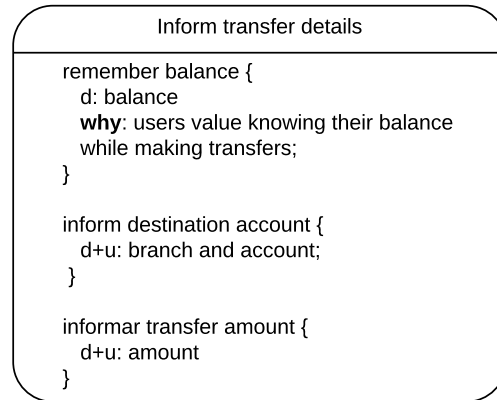


Figure 5.19: Why in a Dialogue

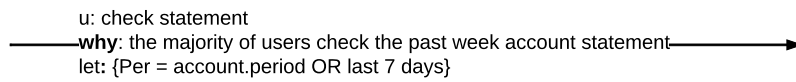


Figure 5.20: Why in a Turn-Yielding Element

positive scores and 2 neutral. Regarding the element's name, 13 respondents approved with positive scores, while 2 were neutral. In the written responses, three respondents further praised the addition of this element. Three respondents considered that the representation of a thicker line might not be the best option. One respondent alleged that MoLIC should not represent differences in terms of visual emphasis, since it should not consider user interface elements. One respondent expressed concerns about the potential increase in language complexity resulting from the inclusion of the new elements.

Analysis: Based on the questionnaire, the addition of the element was approved by most respondents. Concern about the element's visual representation of the *Main Scene* was raised because, depending on the line thickness, this differentiation could be made visually difficult to discern. To address the concern of referring to the user interface, it is important to clarify that this element represents the designer's intention to prioritize one conversation over another when there are multiple ways to achieve the same goal, thus not primarily related to the interface. The concrete example given in the questionnaire was unfortunate in this regard because it referred to user interface elements, which may have caused the misunderstanding. In this analysis, we provide a new example, depicted in Figure 5.21

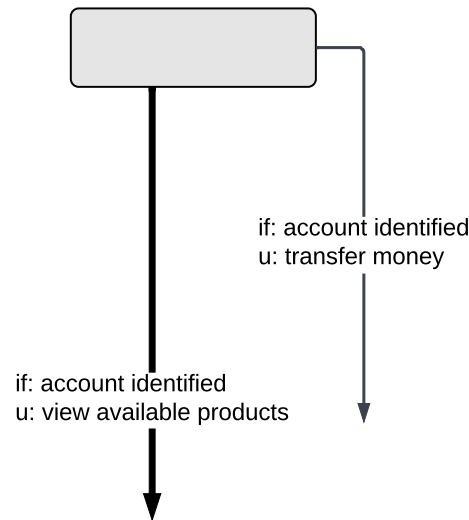


Figure 5.21: Preferred Conversation

5.6.11 Main Scene

This element (see section 4) was proposed by Araujo and Barbosa (2008) and added to MoLIC V4. Most respondents agreed with the element: 15 positive scores. Only one respondent gave a lower score and explained that, depending on the border thickness, it could be hard to differentiate the main scene from the other scenes. This respondent suggested exploring other forms for the element. Other respondents reinforced that, while the approach was clean and efficient, the border should be thick enough to clearly stand out from the other scenes. When asked to suggest alternative names, main page, initial scene, and opening scene were suggested.

Analysis: From the questionnaire, we can extract that the element *Main Scene* (depicted in Figure 5.22) was well received. One point of attention in future uses will be the border thickness that has to be easily differentiated from the other scenes. The suggested names were not considered adequate because they either referred to a specific platform (*page*) or because they implied it would always be the initial point of the interaction (*initial*; *opening*), which may not be the case.

5.6.12 When - Action Associated with an Event

The *when* element (see section 17) was proposed to allow the designer to make predefined interruptions on the user's turn in association with the element *Simultaneous Speech in the User's Turn*. The element received 12 positive scores for its notation (**when:** clauses) and 13 positive scores for its name.

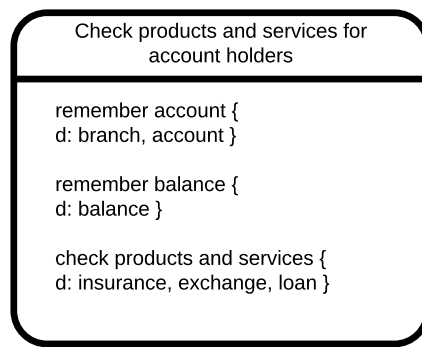


Figure 5.22: Main Scene

One respondent stated that this element was poorly differentiated from the *Simultaneous Speech in the User's Turn*. Three respondents praised the addition of the element, while one alleged that the term is unclear. An alternative name was proposed: *conditional event*.

Analysis: It is not unusual during user-system interactions for events to cause deviations from the current conversation flow. A clear example is a banking system where the user must log in again after a timeout period (*i.e.*, period of inactivity). To represent this scenario, MoLIC should allow the representation of a “turn-grabbing” utterance from the designer’s deputy. To develop this element, we had to consider that the designer should be able to interrupt the user’s conversation at specific moments. Thus, we proposed the element **When**, as depicted in Figure 5.23 to “watch” the conversation and affect it at the specified moment (*i.e.*, as soon as the specified event occurs). The overall perception of the respondents about the new element was positive.

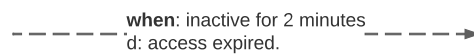


Figure 5.23: When

One respondent was concerned that the **when** element might be confused with the *Simultaneous Speech in the Designer's Turn* (where it also occurs). To address that, future works and instructional material should present usage scenarios of the element to make it more familiar to MoLIC users. For instance, a **when** element can go from one scene to another. Imagine a poll scenario where the user is in a scene to choose one of the options and has only a limited amount of time to do so. After the specified period elapses, the **when** element deviates the conversation to a scene for the user to see the poll results, regardless of whether the user had submitted their choice.

5.7

General Questions about Usage Scenarios

5.7.1

Usage Scenarios of Conditions, Consequences (Effects), and Utterances in Turn-Yielding Elements

To understand the respondents' perceptions of the different meanings of the elements **if**, **u**, **d**, and **effect**, we presented (see section 27) four diagram fragments that covered seven usage variations:

- (a) silent turn yielding (*e.g.*, **if**: valid account info and matching user and password): 11 positive scores;
- (b) a designer's utterance describing the result of a system processing, given that a certain condition was satisfied (*e.g.*, **if**: valid address; **d**: card requested, please wait for delivery): 11 positive scores;
- (c) a condition combined with designer's (repair) utterance, given that the description of the condition is exactly what the designer wants to communicate to the user (*e.g.*, **if/d**: invalid account info and/or user password): 9 positive and 2 neutral scores; one respondent had mixed feelings about this combination, and another said that the **if** was unnecessary because it was a breakdown;
- (d) a user's utterance to shift or advance a topic, enabled by the satisfaction of a certain condition (*e.g.*, **if**: user is logged in; **u**: request card): 10 positive and 1 negative score;
- (e) silent turn yielding with a resulting effect (*e.g.*, **if**: change default account statement period == no; **then**: {session.statement-period = Per}): 9 positive and 2 neutral scores;
- (f) a designer's utterance describing the result of a system processing, given that a certain condition was satisfied, with a resulting effect (*e.g.*, **if**: change default account statement period == yes; **then**: {account.statement-period = Per, session.statement-period = null}): 9 positive scores;
- (g) a condition combined with designer's utterance, given that the description of the condition is exactly what the designer wants to communicate to the user (*e.g.*, **if/d**: N products found; **then**: {session.search-result = found products}): 10 positive scores.

Analysis: Overall, the usage scenarios of conditions, consequences, and turn-yielding elements were very well accepted in the questionnaire, raising only a few concerns. The combination *if/d* raised mixed feelings by one respondent and was considered unnecessary by another. We suggested this format to avoid cases such as *if: invalid email; d: invalid email*. Previously in MoLIC, it was common to use the designer’s turn-yielding utterance as both the communication and the conditional (the same applied to the designer’s repair utterance). This misunderstanding may be due to a proposed syntactic sugar in MoLIC V1, which considered that, when the **pre:** and **d:** clauses were the same, only **d** needed to be specified in the diagram. However, this overload of the **d:** element to represent both a condition and a designer’s utterance created some conceptual confusion, so it is no longer advised. To avoid this, MoLIC V4 demands that, if a condition must be met for the conversation to move forward in that direction, it should be explicitly represented through the element **if** (or the combined elements **if/d**).

5.7.2

Declaration of Interest

Nine respondents expressed interest in participating in a future interview about MoLIC ([see section 29](#)).

5.7.3

Clarification

Along with the questionnaire answers, we identified a misunderstanding that we would like to clarify here and emphasize the need for better explanations of the corresponding elements in teaching materials on MoLIC:

One respondent incorrectly alleged that the user could only know what the designer says or allows the user to say if the utterance is within a scene. Therefore, it is important to clarify that the user should always receive the designer’s message, whether it is defined in the utterance that leads to a scene or inside the scene.

5.7.4

Limitations of the Study

When analyzing the questionnaire answers we noticed a drop in justifications starting from section 19 onwards (out of a total of 29 sections). We found that nearly half of the respondents stopped justifying their scores. This limits the analysis, as the absence of explanations in these cases reduces a comprehensive understanding of the scores. It is important to acknowledge that this limitation

does not invalidate the study, since most changes made in this work were not subjected to approval, as they were grounded in the linguistics concepts. The primary objective of this questionnaire was to understand the participants' perceptions regarding the changes made and to identify elements that would require further attention in future works.

6

Conclusion

In this chapter, we present the main contributions of this study and propose topics to be addressed in future works.

6.1

Main Contributions

In this section, we describe how we responded to each research question.

RQ1: How to realign MoLIC with the interaction-as-conversation metaphor? By revisiting the elements of MoLIC V2nd collecting extension proposals made in the works of Araujo and Barbosa (2008) and Pessanha and Barbosa (2018), we were able to identify which elements deviated from the interaction-as-conversation metaphor. We then searched in the Linguistics literature concepts corresponding to the same conversational situations the elements intended to represent. When we could not relate to linguistic concepts, we relied on terms used in informal conversation. From this, we renamed some elements included in MoLIC V4. As a result, we brought the language closer to its original interaction-as-conversation metaphor, grounded in semiotic engineering (Barbosa and de Paula, 2003b).

RQ2: How to expand the language without increasing its complexity? An important step to avoid increasing complexity was the revision of only the elements that were, to the best of our knowledge, widely used, or those proposed by Araujo and Barbosa (2008) and Pessanha and Barbosa (2018) with sound argumentation (even though not always in line with the interaction-as-conversation metaphor). Another important pillar to keep the simplicity was collecting the perceptions of the changes in the questionnaire so that, when multiple concepts or element names were equally well-aligned with the conversation metaphor, we could choose the one that respondents felt most comfortable with. The elements in this situation were: (1) empty scene, successful completion, and topical conversation completion; (2) overlap in (user's/designer's) turn, and simultaneous speech in (user's/designer's) turn; (3) (user's/designer's) utterance, and (user's/designer's) turn-yielding utterance; (4) user's repair utterance, and user's turn-yielding repair utterance; (5) designer's breakdown repair utterance, and designer's turn-yielding

breakdown repair utterance; (6) then, and effect.

RQ3: How do users of previous MoLIC versions perceive the changes? In an attempt to anticipate how the MoLIC users, in general, would perceive the changes proposed in this work, our questionnaire also collected the respondents' previous experiences with and perception of MoLIC. Experienced users of MoLIC welcomed most changes since they addressed scenarios that could not be unambiguously or consistently represented in MoLIC before. Examples of praised elements were the *main scene*, *preferred conversation*, *when*, and *simultaneous speech in designer's turn*.

Some respondents complained about renaming certain elements because they were already accustomed to the existing ones. However, they did not argue about their alignment with the metaphor, which was the reason for the changes. Therefore, although their feedback did not change our argumentation and design choices, we will take it to mean that, going forward, the instructional material on MoLIC V4 should argue for the benefits of the changes.

In a few cases, respondents questioned one of the multiple change proposals to an element that did not differ in relation to their alignment with the interaction-as-conversation. In this case, their responses largely drove the element definitions and representations adopted in MoLIC V4. This occurred with the positioning of the emitter prefix (**d:** or **d+u:**), which was previously (in MoLIC V2 associated to dialogues and is now associated with the signs inside dialogues).

It is important to clarify our purpose with the questionnaire study. As some of the proposed changes were well grounded in linguistic concepts, they were not intended for validation but rather for analyzing MoLIC users' perceptions of the changes. In this case, we aimed to understand which elements may require more careful communication and more detailed instructional material so as to reduce the respondents' efforts to incorporate the changes in their usage of MoLIC V4.

6.2

Future Work

For future work, we strongly suggest different strategies to evaluate MoLIC usage and occasional proposals for modification or extension. The questionnaire we developed was long and required extensive interpretation of concepts, usage scenarios, and diagrams, which had to be read, understood, and assessed. As such, we recommend adopting strategies such as dividing the questionnaire into multiple smaller questionnaires (for larger audiences and quicker assessments) or, preferably, conducting an in-depth qualitative study

with user observation and interviews, instead of either one long or multiple short online questionnaires.

With MoLIC V4, a consolidated version better aligned with the interaction-as-conversation metaphor and linguistic concepts, a new version of the practical guide Silva and Barbosa (2007) would greatly contribute to the community of MoLIC users. Along with this, we suggest the revision of MoLIC, which is an inspection technique to ensure that MoLIC diagrams are “complete, consistent, unambiguous, and contain few or no defects” (Lopes et al., 2015).

Another opportunity from this new version is to expand the scope of the systems covered by the language, for instance, by revisiting works related to collaborative systems (de Souza et al., 2015) and AI (Fernandes et al., 2021).

When analyzing the questionnaire responses about the participant’s self-assessment of their confidence in *creating* diagrams, we noticed a drop in the scores compared to their comfort with *reading and interpreting* diagrams. A study on the language’s users’ most frequent doubts should be conducted so that practical instructional material for learning MoLIC is able to present a rich catalog of examples and usage situations to clarify those doubts. Such material would not only benefit new MoLIC users but also reduce the effort of existing MoLIC users to learn and adopt the new version.

Finally, we recommend an in-depth, longitudinal study of the epistemic character of MoLIC, to investigate what MoLIC diagrams can reveal about the underlying problem, the represented solution, and how the two are related.

Bibliography

- Araujo, A. C. I. C. d. and Barbosa, S. D. J. (2008). Apoio ao design e à interpretação de modelos de interação humano-computador representados em molic.
- Barbosa, S. and Silva, B. (2010). *Interação Humano-Computador*. Elsevier Brasil.
- Barbosa, S. D. J. and de Paula, M. G. (2003a). Designing and evaluating interaction as conversation: A modeling language based on semiotic engineering. In Jorge, J. A., Jardim Nunes, N., and Falcão e Cunha, J., editors, *Interactive Systems. Design, Specification, and Verification*, pages 16–33, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Barbosa, S. D. J. and de Paula, M. G. (2003b). Designing and Evaluating Interaction as Conversation: A Modeling Language Based on Semiotic Engineering. In Jorge, J. A., Jardim Nunes, N., and Falcão e Cunha, J., editors, *Interactive Systems. Design, Specification, and Verification*, pages 16–33, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Barbosa, S. D. J., Silva, B. S. d., Silveira, M. S., Gasparini, I., Darin, T., and Barbosa, G. D. J. (2021). *Interação Humano-Computador e Experiência do Usuário*. Autopublicação.
- Brown, G. and Yule, G. (1983). *Discourse Analysis*. Cambridge Textbooks in Linguistics. Cambridge University Press.
- de Carvalho, A. P., Pereira, F. H. S., Assunção, O. B., Pereira, A. F., and Prates, R. O. (2019). An analysis of molic’s consolidation. In *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems*, IHC ’19, New York, NY, USA. Association for Computing Machinery.
- de Paula, M. G. and Barbosa, S. D. J. (2003). Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica: Construção de um modelo de interação.

- de Paula, M. G., da Silva, B. S., and Barbosa, S. D. J. (2005). Using an interaction model as a resource for communication in design. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, page 1713–1716, New York, NY, USA. Association for Computing Machinery.
- de Souza, C. S. (2005a). Semiotic engineering: bringing designers and users together at interaction time. *Interacting with Computers*, 17(3):317–341.
- de Souza, C. S. (2005b). *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press.
- de Souza, C. S. and Leitão, C. F. (2009). Semiotic Engineering Methods for Scientific Research in HCI. *Synthesis Lectures on Human-Centered Informatics*, 2(1):1–122.
- de Souza, L. G., Barbosa, S. D. J., and Conte, T. (2015). Evaluating a molic extension for collaborative systems design. In *Proceedings of the 14th Brazilian Symposium on Human Factors in Computing Systems*, IHC '15, New York, NY, USA. Association for Computing Machinery.
- Declerck, R. H. C. (1997). *When-Clauses and Temporal Structure*. Routledge, London.
- Egins, S. and of Sydney, U. (1990). *Keeping the Conversation Going: A Systemic-functional Analysis of Conversational Structure in Casual Sustained Talk*. University of Sydney.
- Faltaous, S., Gruenefeld, U., and Schneegass, S. (2021). Towards a universal human-computer interaction model for multimodal interactions. In *Proceedings of Mensch Und Computer 2021*, MuC '21, page 59–63, New York, NY, USA. Association for Computing Machinery.
- Fernandes, U. d. S., Prates, R. O., Chagas, B. A., and Barbosa, G. A. R. (2021). Analyzing molic’s applicability to model the interaction of conversational agents: A case study on ana chatbot. In *Proceedings of the XX Brazilian Symposium on Human Factors in Computing Systems*, IHC '21, New York, NY, USA. Association for Computing Machinery.
- Ferreira, J. J., Fucs, A., and Segura, V. (2019). Modeling People-AI Interaction: A Case Discussion with Using an Interaction Design Language. In Marcus, A. and Wang, W., editors, *Design, User Experience, and Usability. User Experience in Advanced Technological Environments*, Lecture

- Notes in Computer Science, pages 379–388, Cham. Springer International Publishing.
- Hoover, S. P., Rinderle, J. R., and Finger, S. (1991). Models and abstractions in design. *Design Studies*, 12(4):237–245.
- Hutchby, I. and Wooffitt, R. (1988). *Conversation Analysis*. Blackwell Publishing Inc.
- Karttunen, L. (1974). Presupposition and linguistic context. *Theoretical Linguistics*, 1:181–194.
- Laskowski, K., Heldner, M., and Edlund, J. (2012). On the dynamics of overlap in multi-party conversation. In *Proc. Interspeech 2012*, pages 847–850.
- Leech, G. (1983). *Principles of Pragmatics*. Longman.
- Lopes, A., Marques, A. B., Barbosa, S. D. J., and Conte, T. U. (2015). Molveric: An inspection technique for molic diagrams. In *International Conference on Software Engineering and Knowledge Engineering*.
- Pessanha, A. V. and Barbosa, S. D. J. (2018). Análise e evolução da molic para especificação da interação humano-computador.
- Prates, R. O., de Souza, C. S., and Barbosa, S. D. J. (2000). Methods and tools: A method for evaluating the communicability of user interfaces. *Interactions*, 7(1):31–38.
- Preece, J., Rogers, Y., and Sharp, H. (2002). *Interaction Design*. John Wiley & Sons, Inc., USA, 1st edition.
- Roger, D., Bull, P., and Smith, S. (1988). The development of a comprehensive system for classifying interruptions. *Journal of Language and Social Psychology - J LANG SOC PSYCHOL*, 7:27–34.
- Sacks, H. (2010). On the Preferences for Agreement and Contiguity in Sequences in Conversation* 1. In *Language in Use*. Routledge. Num Pages: 15.
- Sacks, H., Schegloff, E. A., and Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.
- Schegloff, E. A., Jefferson, G., and Sacks, H. (1977). The preference for self-correction in the organization of repair in conversation. *Language*, 53(2):361–382.

- Searle, J. R. (1975). A taxonomy of illocutionary acts. In Gunderson, K., editor, *Language, Mind and Knowledge*, pages 344–369. University of Minnesota Press.
- Silva, B. S. D. and Barbosa, S. D. J. (2005). Molic segunda edição: revisão de uma linguagem para modelagem da interação humano-computador.
- Silva, B. S. D. and Barbosa, S. D. J. (2007). Designing human-computer interaction with molic diagrams – a practical guide.
- Silveira, M., Barbosa, S., and de Souza, C. (2004). Designing online help systems for reflective users. *J. Braz. Comp. Soc.*, 9:25–38.
- Stenström, A. (1994). *An Introduction to Spoken Interaction*. Learning about language. Longman.
- Stokoe, E., Humă, B., Sikveland, R. O., and Kevoe-Feldman, H. (2020). When delayed responses are productive: Being persuaded following resistance in conversation. *Journal of Pragmatics*, 155:70–82.
- Wooffitt, R. (2005). Conversation Analysis and Discourse Analysis.

A

History of Changes - MoLIC Elements

In Table A.1, we list all the elements present in the MoLIC 4th version, detailing in which version each element was proposed and all changes made to it over the years in the selected works. The table legends are the following: P, for proposed; C, for changed; U, for used as proposed, and R, for removed.

Table A.1: Elements' versions

Element	V1	V2	V3 (not revised)	Barbosa and Silva (2010)	Pessanha and Barbosa (2018)	4th Version
Alert Scene				P	C (unified representation of alert scene)	U (as proposed in the HCI Text Book)
Combined Utterances	P (unnamed)	not cited	U	not cited	C (utterance one way and both ways)	R (removed element due to ambiguity)
Contact Points		P	U	not cited	C (differentiate sync and async interactions)	U (as proposed in the V2)
Context Features	P (as control signs)	U	C (proposed context-related articulated signs)	not cited	not cited	U (unified V3 proposals and renamed)
Conversation Closing Point		P	U	U	U	U
Conversation Opening Point		P	U	U	U	U
Designer's (Turn-Yielding) (Breakdown) Repair Utterance	P	C (changed representation to dashed line)	U	U	U	C (replaced transition with turn yielding, changed recovery to repair, and added the emitter)
Designer's Turn-Yielding Utterance	P	U	U	U	U	C (replaced transition with turn yielding)
Dialog Grouping Operands		P	U	U	C (change operands representation)	U (as defined in the V2)
Dialogue	P	C (added dialog grouping operands)	C (define sign emitter in the dialogue instead of in each sign)	U	U	U

Table A.1: Elements' versions (continued from previous page)

Element	V1	V2	V3 (not revised)	Barbosa and Silva (2010)	Pessanha and Barbosa (2018)	4th Version
Effect	P (post-conditions)	C (renamed to perl)	C (renamed to perl) effect and added the differentiation of implicit and explicit effect) U	not cited	not cited	C (used as defined in the V3 and renamed to then)
External Interlocutor and Fork		P	U	not cited	C (differentiate sync and async interactions)	U (as defined in the V2)
Global Topic Shift	P (ubiquitous access)	U	U	U	C (renamed to global topic change and proposed condition inside the element)	C (renamed to Global Topic Shift and added Pessanha's condition inside element as a usage variant)
If	P (pre)	U	C (renamed to precond)	U (as defined in the V3)	C (renamed to if)	U (as proposed by Pessanha)
Let						P
List		P (set)	U	C (list)	C (differentiate set and list and add cardinality)	C (use only the term list to ordered and unordered sign options. Added cardinality)
Main Scene			P	not cited	C (topic between asterisks to identify main scene)	C (used as defined in the V3 and changed the representation to a thicker border)
Preferred Conversation			P	not cited	not cited	C (used as defined in the V3 and changed the representation to a bolder utterance)

Table A.1: Elements' versions (continued from previous page)

Element	V1	V2	V3 (not revised)	Barbosa and Silva (2010)	Pessanha and Barbosa (2018)	4th Version
Scene	P	U	C (proposed main scene)	U (as defined in the V2)	C (topic between asterisks to identify main scene)	U (as defined in the V3)
Signs	P	C (required signs inside the dialogue)	C (proposed context-related articulated signs)	U (as V2)	C (indicate only optional signs)	U (as proposed by Pessanha)
Silent Turn-Yielding				P	U	U (as proposed in the HCI Text Book)
Simultaneous Speech in the Designer's Turn		P (ongoing process and scene coupling)	U	U	U	C (renamed scene coupling and removed ongoing process)
Simultaneous Speech in the User's Turn						P
System Process	P	U	U	U	U	U
Topical Conversation Completion	P (as special scene)	C (replaced special scene with monologue)	C (replaced monologue with empty scene)	U (as defined in the V3)	not cited	C (renamed V3 element)
User's Repair Utterance	P (same visual representation as user's utterance)	U	C (changed representation to dashed line)	U (as defined in the V3)	U (as defined in the V3)	C (named the element)
User's Turn-Yielding Utterance	P	U	U	U	U	C (replaced transition with turn yielding)
When					P	P
Why			P (presup)	not cited	C (renamed to justify)	C (used as defined in the V3; renamed to why; expanded the use to dialogs)

B

Elements Translation to Portuguese

In Table B.1, we present the translation to Portuguese for each element in MoLIC V4.

Table B.1: Elements' translation to Portuguese

Element	V1
Alert Scene	Cena de Alerta
Contact Point	Ponto de Contato
Context Feature	Característica do Contexto
Conversation Closing Point	Ponto de Encerramento
Conversation Opening Point	Ponto de Abertura
Designer's (Turn-Yielding) (Breakdown) Repair Utterance	Fala de Reparação (de Ruptura e Passagem de Turno) do Designer
Designer's (Turn-Yielding) Utterance	Fala (de Passagem de Turno) do Designer
Dialog Grouping Operands	Operador de Agrupamento de Diálogos
Dialogue	Diálogo
Effect	Efeito
External Interlocutor and Fork	Interlocutor Externo e Bifurcação
Global Topic Shift	Mudança Global de Tópico
If	Se
Let	Definição
List	Lista
Main Scene	Cena Principal
Preferred Conversation	Conversa Preferencial
Scene	Cena
Signs	Signos
Silent Turn-Yielding	Passagem de Turno Silenciosa
Simultaneous Speech in the Designer's Turn	Fala Simultânea no Turno do Designer
Simultaneous Speech in the User's Turn	Fala Simultânea no Turno do Usuário
System Process	Processamento do Sistema
Topical Conversation Completion	Conclusão da Conversa sobre um Tópico
User's Repair Utterance	Fala de Reparação do Usuário
User's (Turn-Yielding) Utterance	Fala (de Passagem de Turno) do Usuário
When	Quando
Why	Porquê

C

Informed Consent - Questionnaire (in Portuguese)

Título do Projeto: Realigning MoLIC to the Interaction-as-Conversation Metaphor Professora Responsável: Simone D.J. Barbosa Aluno responsável: Caroline Loppi Instituição a que pertencem a professora e o aluno responsável: PUC-Rio E-mail: loppi.clg@gmail.com

Você está sendo convidado a participar da pesquisa para o projeto “Realigning MoLIC to the Interaction-as-Conversation Metaphor”, de responsabilidade da aluna Caroline Loppi. Este estudo tem o objetivo de avaliar propostas de alteração da linguagem MoLIC – Modeling Language for Interaction as Conversation – que foram feitas ao longo dos anos. O público-alvo são pessoas com noções de design de interação e com conhecimento de MoLIC (do básico ao avançado).

Todas as perguntas e todos os assuntos tratados no questionário levarão em conta sua experiência com o tema. O estudo terá seus dados coletados estatisticamente para os fins da pesquisa.

A participação neste estudo é voluntária. Você pode pular as perguntas que não queira responder e interromper o questionário a qualquer momento. Ao final do questionário, você terá uma opção de consentir para uma entrevista sobre o mesmo tema.

As informações que você compartilhar conosco serão mantidas confidenciais. Os dados serão mantidos em pasta protegida por senha e ninguém fora da equipe de pesquisa terá acesso às suas respostas ou mesmo saberá que você participou deste estudo. Em qualquer apresentação dos resultados, seu anonimato e privacidade serão preservados.

Você deve ter ao menos 18 anos de idade para participar deste estudo.

Caso você tenha alguma dúvida sobre o estudo, contacte Caroline Loppi, loppi.clg@gmail.com.

Caso você tenha dúvidas ou preocupações com relação aos seus direitos como participante do estudo, você pode contactar a Câmara de Ética em Pesquisa da PUC-Rio à Rua Marquês de São Vicente, 225, Prédio Kennedy, 2o andar – Gávea – RJ. Parecer 101/2020.

Ao completar este questionário, você estará consentindo em participar do estudo. Por favor imprima uma cópia deste formulário para seus registros.

O questionário online, disponível até 03/06/2023 pode ser encontrado no seguinte link: <https://forms.gle/zSHhvag4e6UpVTEPA>

D

Questionnaire to Evaluate MoLIC Changes in V4

Questionário sobre a MoLIC

Título do Projeto: Realigning MoLIC to the Interaction-as-Conversation Metaphor

Professora Responsável: Simone D.J. Barbosa

Aluno responsável: Caroline Loppi

Instituição a que pertencem a professora e o aluno responsável: PUC-Rio

E-mail: loppi.clg@gmail.com

Você está sendo convidado a participar da pesquisa para o projeto "Realigning MoLIC to the Interaction-as-Conversation Metaphor", de responsabilidade da aluna Caroline Loppi. Este estudo tem o objetivo de avaliar propostas de alteração da linguagem MoLIC -- Modeling Language for Interaction as Conversation -- que foram feitas ao longo dos anos. O público-alvo são pessoas com noções de design de interação e com conhecimento de MoLIC (do básico ao avançado).

Todas as perguntas e todos os assuntos tratados no questionário levarão em conta sua experiência com o tema. O estudo terá seus dados coletados estatisticamente para os fins da pesquisa.

A participação neste estudo é voluntária. Você pode pular as perguntas que não queira responder e interromper o questionário a qualquer momento. Ao final do questionário, você terá uma opção de consentir para uma entrevista sobre o mesmo tema.

As informações que você compartilhar conosco serão mantidas confidenciais. Os dados serão mantidos em pasta protegida por senha e ninguém fora da equipe de pesquisa terá acesso às suas respostas ou mesmo saberá que você participou deste estudo. Em qualquer apresentação dos resultados, seu anonimato e privacidade serão preservados.

Você deve ter ao menos 18 anos de idade para participar deste estudo.

Caso você tenha alguma dúvida sobre o estudo, contacte Caroline Loppi, loppi.clg@gmail.com.

Caso você tenha dúvidas ou preocupações com relação aos seus direitos como participante do estudo, você pode contactar a Câmara de Ética em Pesquisa da PUC-Rio à Rua Marquês de São Vicente, 225, Prédio Kennedy, 2o andar – Gávea – RJ. Parecer 101/2020.

Ao completar este questionário, você estará consentindo em participar do estudo. Por favor imprima uma cópia deste formulário para seus registros.

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

* Indica uma pergunta obrigatória

Questionário sobre a MoLIC

Observação: Ao responder o questionário, tenha em mente as pessoas que ainda irão conhecer a linguagem. Portanto, evite justificativas como “prefiro, pois já estou acostumado(a)”.

Qual o seu grau de conhecimento da MoLIC? *

1 2 3 4 5 6 7

Conheço muito pouco

☒ ☐ ☐ ☐ ☐ ☐ ☐

Sou especialista

Em quais perfis você se encaixa?

- ☐ Aluno(a) de graduação
- ☐ Aluno(a) de pós-graduação lato sensu / especialização
- ☐ Aluno(a) de mestrado acadêmico
- ☐ Aluno(a) de doutorado
- ☐ Professor(a)
- ☐ Pesquisador(a)
- ☐ Profissional



Qual é sua principal área de formação?

☐ Computação

☐ Design

☐ Engenharia

☐ Psicologia

☐ Outro: _____

Qual o seu grau de conhecimento sobre **design de interação**?

1 2 3 4 5 6 7

Conheço muito pouco

☐ ☐ ☐ ☐ ☐ ☐ ☐

Sou especialista

O quanto a **MoLIC** contribui para o design de interação?

1 2 3 4 5 6 7

Nem um pouco

☐ ☐ ☐ ☐ ☐ ☐ ☐

É essencial

Com que frequência você usa MoLIC no seu dia-a-dia de pesquisa ou trabalho?

☐ Nenhum projeto

☐ Em até 25% dos projetos

☐ Em até 50% dos projetos

☐ Em até 75% dos projetos

☐ Em mais de 75% dos projetos

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)



Questionário sobre a MoLIC

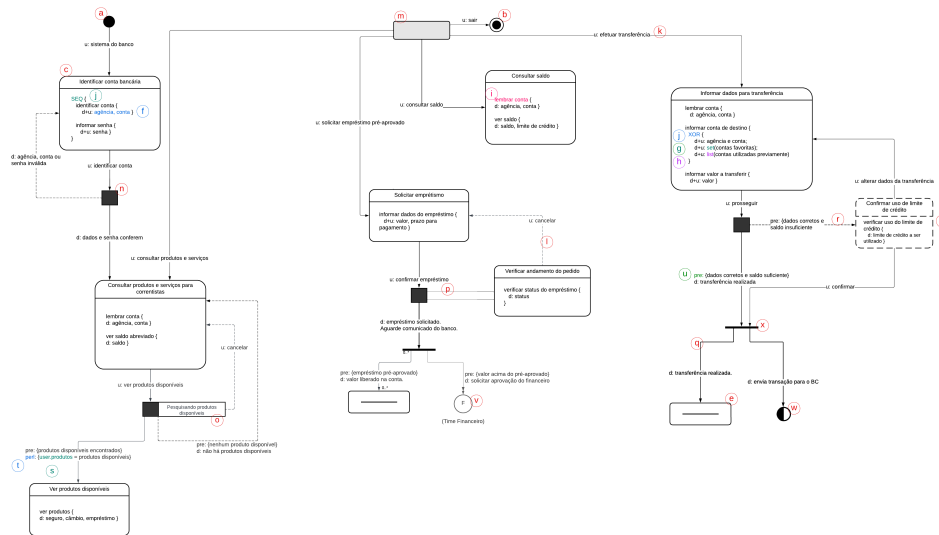
loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Autoavaliação do seu Conhecimento sobre a MoLIC

Considere o diagrama a seguir. Por questões de legibilidade, recomendamos fortemente que você baixe o arquivo com o [diagrama em formato vetorial \(PDF\)](#).



Considerando o diagrama acima, o quão confortável você se sente, de uma forma geral, **lendo/interpretando** diagramas MoLIC?

1 2 3 4 5 6 7

Pouco confortável



Muito confortável



Considerando o diagrama acima, o quão confortável você se sentiria **construindo** outros diagramas MoLIC?

	1	2	3	4	5	6	7	
Pouco confortável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito confortável

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Proposto

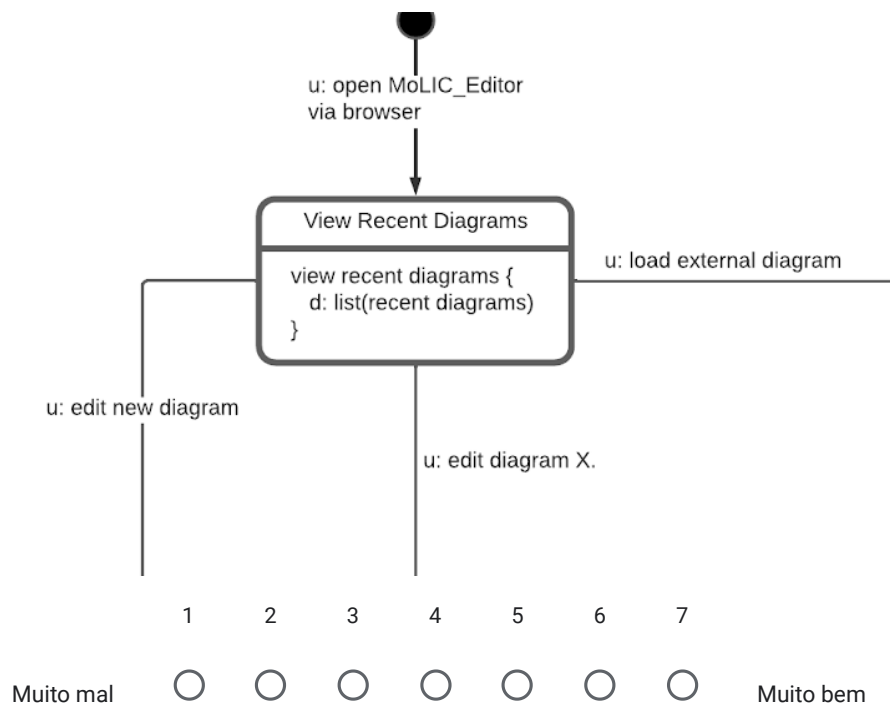
Para representar o conceito de **cena principal**, foi proposto o seguinte elemento:
Main Scene.

Conceito: em muitos sistemas, uma cena pode ser considerada principal, ou seja, é a partir dela que a maior parte das conversas são iniciadas. Um exemplo é a cena associada à página principal de uma aplicação.

Representação: cena com a borda mais grossa.

Exemplo: página inicial da aplicação de edição de diagramas a partir de onde é possível navegar para diversas funcionalidades do sistema, tais como: editar um novo diagrama, carregar um diagrama externo e editar um diagrama existente.

Considerando que este elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você não concorde com o nome desse elemento (Main Scene), **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Proposto

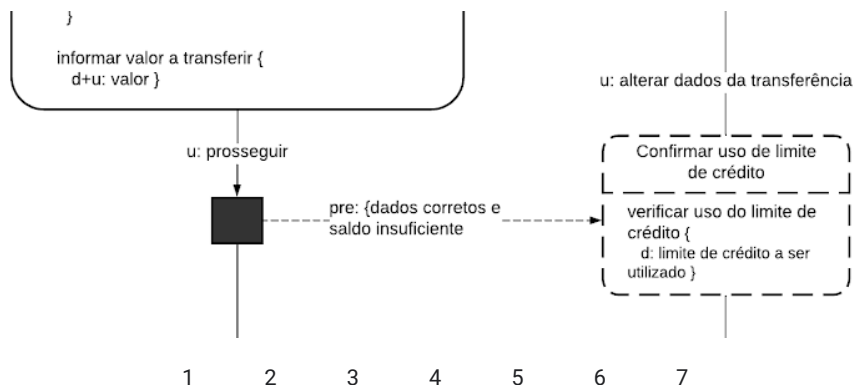
Para representar o conceito de **cena de alerta**, foi proposto o seguinte elemento:
Alert Scene.

Conceito do Elemento: cena usada para alertar o usuário sobre uma possível causa de falha de comunicação, caso ele prossiga com a conversa.

Representação: cena com a borda tracejada

Exemplo: durante uma transferência, após confirmar os dados, o usuário é informado que ele não possui saldo suficiente e que, caso confirme a transação, entrará no seu limite de crédito.

Considerando que este elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



Muito mal 1 2 3 4 5 6 7 Muito bem



Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você não concorde com o nome desse elemento (Alert Scene), **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Modificado

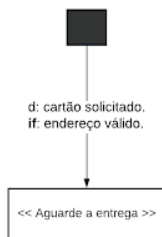
Para representar o conceito abaixo, foram propostos diversos elementos.

Conceito: indicar a conclusão de uma conversa relativa a um determinado objetivo.

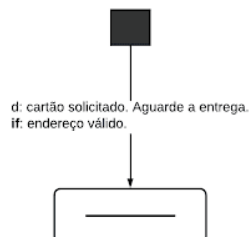
Exemplo: após realizar a solicitação de um novo cartão com sucesso, o usuário será apenas informado sobre o resultado da solicitação, a partir de onde ele precisará mudar de assunto para alcançar algum outro objetivo.

O quão bem cada alternativa ilustra o conceito indicado, considerando que ela deve estar alinhada à metáfora de interação como uma conversa?

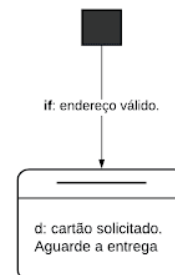
Alternativa A



Alternativa B



Alternativa C



Alternativa A: Monólogo do designer.

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem



Alternativa B: Cena vazia com a mensagem do designer na fala de troca de turno.

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Alternativa C: Cena sem tópico contendo apenas a fala do designer, indicando conclusão bem (ou mal) sucedida do processamento.

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Considerando os elementos acima, se você tivesse que escolher entre os nomes a seguir, qual estaria mais alinhado à metáfora de interação como conversa?

- ☐ Empty Scene
- ☐ Successful Completion
- ☐ Topical Conversation Completion
- ☐ Outro: _____

Explique por que o nome escolhido é mais adequado.

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Modificado

Para representar o conceito abaixo, mais de um elemento foi proposto.

Conceito: Diferenciar os signos obrigatórios dos opcionais em um diálogo.

Exemplo: um formulário tem os campos nome, CPF e RG, onde RG, é opcional.

O quão bem cada alternativa ilustra o conceito indicado, considerando que ela deve estar alinhada à metáfora de interação como uma conversa?

Alternativa A

Criar Conta

informar dados {
 d+u: nome, CPF, RG?
}

Alternativa B

Criar Conta

informar dados {
 d+u: nome*, CPF*, RG
}

Alternativa A: Indicar apenas signos opcionais com interrogação: nome, CPF, RG?

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Alternativa B: Indicar apenas signos obrigatórios com asterisco: nome*, CPF*, RG

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem



Alternativa C: Não indicar na MoLIC signos obrigatórios ou opcionais: nome, CPF, RG.

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você ache que nenhuma dessas representações é adequada, **que representação você sugere**, alinhada à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Modificado

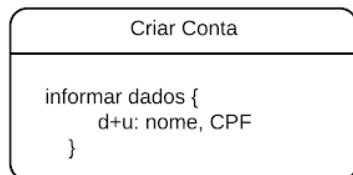
Para representar o conceito abaixo, mais de um elemento foi proposto.

Conceito: Formatação dos diálogos e signos de uma cena.

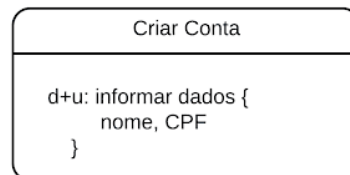
Exemplo: **diálogo** *informar dados*, com **signos** *nome* e *CPF*.

O quão bem cada alternativa ilustra o conceito indicado, considerando que ela deve estar alinhada à metáfora de interação como uma conversa?

Alternativa A



Alternativa B



Alternativa A

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Alternativa B

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem



Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você ache que nenhuma dessas representações é adequada, **que representação você sugere**, alinhada à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Modificado

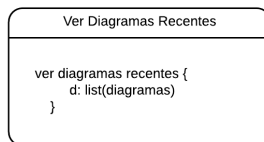
Para representar o conceito abaixo, mais de um elemento foi proposto.

Conceito: indicar um conjunto de opções ou valores associados a um signo.

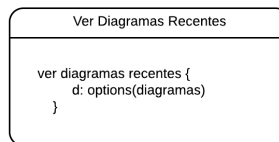
Exemplo: o sistema irá apresentar uma lista de diagramas recentes para o usuário visualizar.

O quão bem cada alternativa ilustra o conceito indicado, considerando que o elemento deve estar alinhado à metáfora de interação como uma conversa?

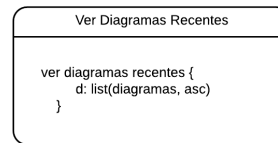
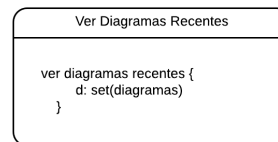
Alternativa A



Alternativa B



Alternativa C



Alternativa A: List - usada para indicar conjuntos de signos, ordenados ou não.

list(diagramas)

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem



Alternativa B: Response Options - usado para qualquer tipo de lista, podendo ou não adicionar ordenação na exibição dos elementos.

options(diagramas) ou options(diagramas, mais recente)

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Alternativa C: Set - usado apenas para listas sem ordenação e List - usado apenas para listas ordenadas.

set(diagramas) e list(diagramas, mais recente)

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você ache que nenhuma desses elementos é adequado, **que elemento você sugere**, alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

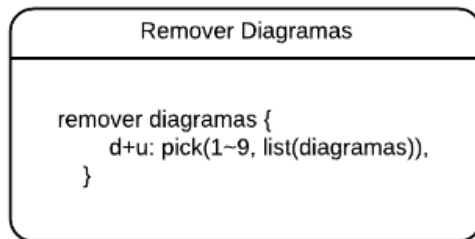
Elemento Proposto

Para representar o conceito de **seleção**, foi proposto o seguinte elemento: **pick**

Conceito: o elemento pick permite ao designer indicar a quantidade de elementos a serem selecionados no conjunto de signos.

Exemplo: selecionar de 1 a 9 diagramas para remoção dentre os diagramas do projeto.

Considerando que este novo elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta



Caso você não concorde com o nome desse elemento, **que nome você sugere**, que seja mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Modificado

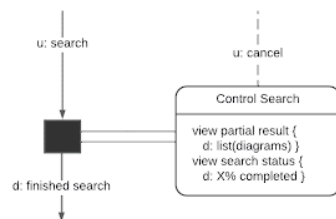
Para representar o conceito abaixo, mais de um elemento foi proposto.

Conceito: representar o progresso de um processamento do sistema e conceder controle do processamento ao usuário.

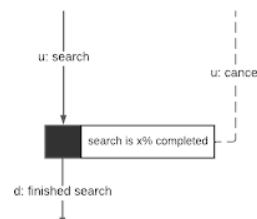
Exemplo: durante o processamento de uma busca, o usuário pode, por exemplo, verificar o tempo restante para conclusão, interromper o processo ou ainda verificar os resultados parciais de uma busca em andamento.

O quão bem cada alternativa ilustra o conceito indicado, considerando que o elemento deve estar alinhado à metáfora de interação como uma conversa?

Alternativa A



Alternativa B



Alternativa A: permite informar sobre múltiplos signos e concede maior controle ao usuário. Também permite cancelar o processamento. Esta alternativa incentiva o designer a fornecer mais controle ao usuário em relação ao processamento (ex: interromper e retomar processo, verificar status da busca e ver resultados parciais de uma busca em andamento).

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem



Alternativa B: permite informar sobre apenas um signo, além de oferecer possibilidade de cancelar o processamento.

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Alternativa C: manter os elementos das alternativas A e B simultaneamente na MoLIC.

	1	2	3	4	5	6	7	
Muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito bem

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você ache que nenhum desses elementos é adequado, **que elemento você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

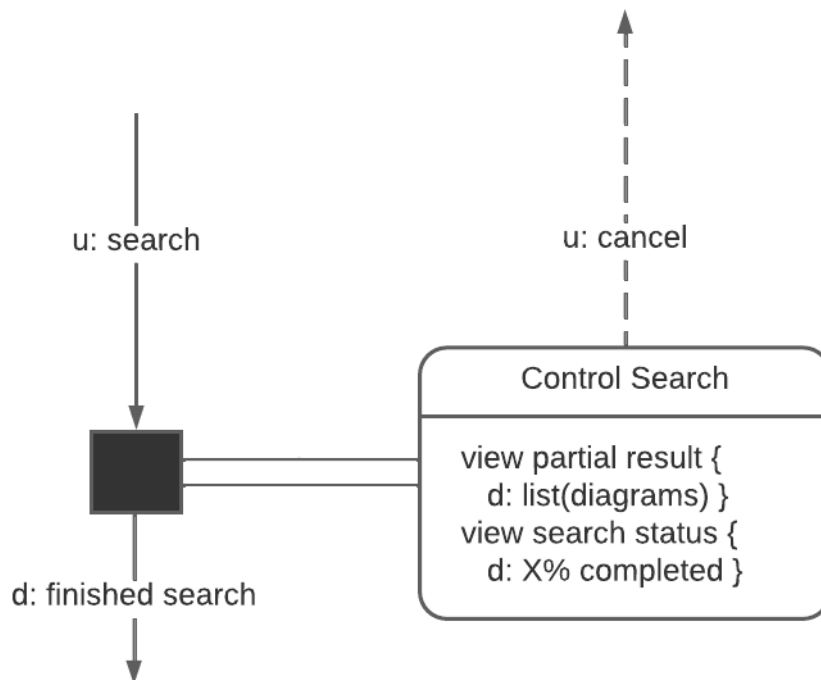
Nome de Elemento

Na nova versão da MoLIC, deu-se um nome para o elemento a seguir, com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito: representar o progresso de um processamento do sistema e, embora o turno seja do designer, o usuário pode interromper a conversa e alterar o seu rumo.

Exemplo: durante o processamento de uma busca no sistema, o usuário pode verificar o resultado parcial retornado, o status da busca ou ainda cancelá-la.

O quanto você concorda com o alinhamento de cada sugestão de nome para o elemento com a metáfora de interação como conversa?



Nome 1: Overlap in Designer's Turn

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Nome 2: Simultaneous Speech in Designer's Turn

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você ache que nenhum desses nomes é adequado, **que nome você sugere**, que estaria mais adequado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



✉ Não compartilhado

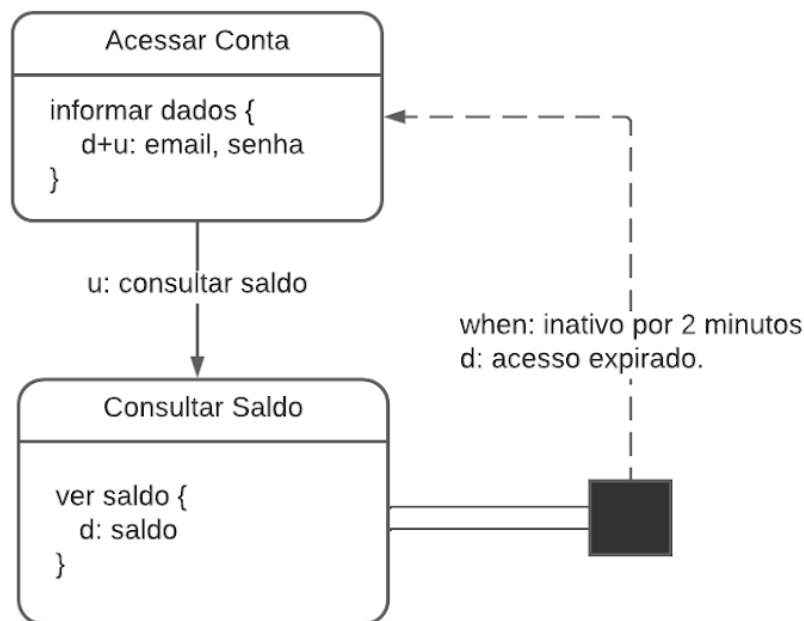
Elemento Proposto

Para representar o conceito de **interrupção na fala do usuário (pelo designer)**, foi proposto o elemento a seguir:

Conceito: permite ao designer definir previamente uma interrupção ao turno do usuário, quando um determinado evento for disparado (que pode ser causado por ou causar uma mudança no contexto da conversa). O evento deve ser definido utilizando o elemento **when**.

Exemplo: enquanto o usuário estiver na tela de consulta de saldo, se ele ficar inativo por mais de 2 minutos, o sistema informará que a sessão expirou e o redirecionará para fazer login novamente.

Considerando que este elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



1 2 3 4 5 6 7

Muito mal

☐☐☐☐☐☐☐

Muito bem



O quanto você concorda com o alinhamento de cada sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome 1: Overlap in User's Turn

1 2 3 4 5 6 7

Discordo totalmente ☐ ☐ ☐ ☐ ☐ ☐ ☐ Concordo totalmente

Nome 2: Simultaneous Speech in User's Turn

1 2 3 4 5 6 7

Discordo totalmente ☐ ☐ ☐ ☐ ☐ ☐ ☐ Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você não concorde com o nome desse elemento, **que nome você sugere**, que estaria mais adequado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



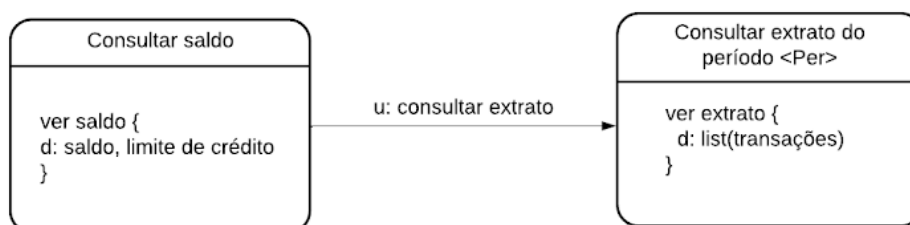
Não compartilhado

Elemento Renomeado

Na nova versão da MoLIC, o elemento a seguir teve seu nome alterado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito do Elemento: indicação de uma oportunidade de mudança de tópico que pode ocorrer na conversa por decisão do usuário.

Exemplo: em uma aplicação de banco, o usuário está visualizando seu saldo e deseja agora visualizar o extrato da conta.



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome Original: User's Transition Utterance

Nome 1: User's Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome Original: User's Transition Utterance

Nome 2: User's Turn-Yielding Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que nenhum desses nomes é adequado à metáfora de interação como conversa, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

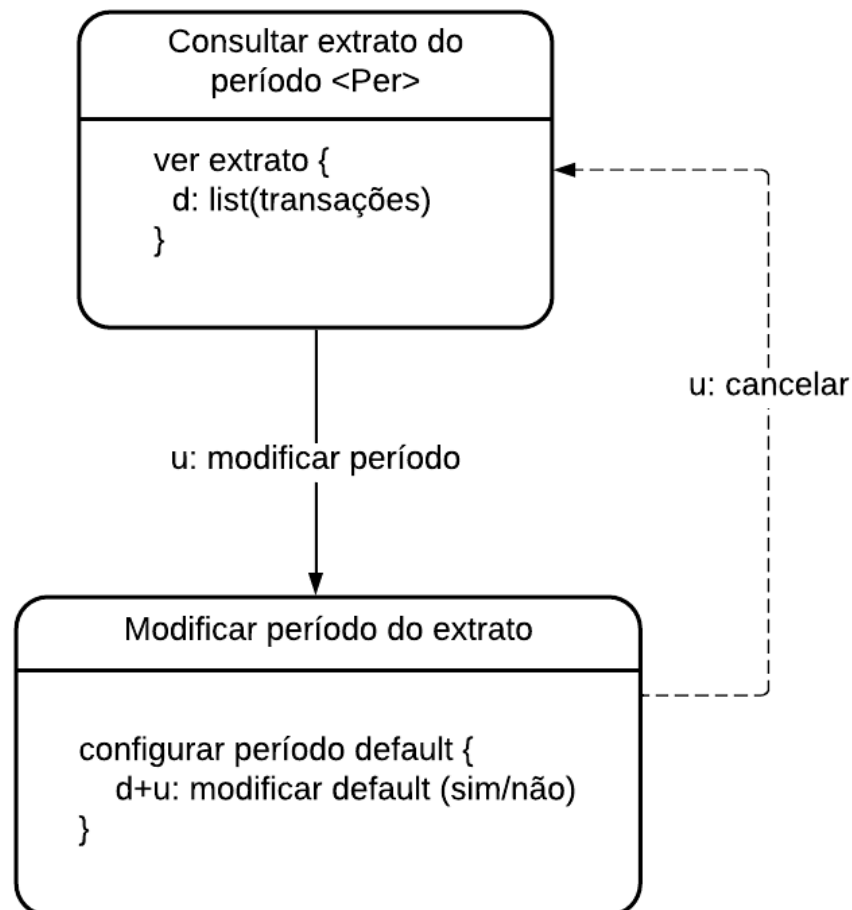
Elemento Nomeado

Na nova versão da MoLIC, o elemento a seguir foi nomeado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito: indicação de que o usuário pode desistir de uma conversa e retornar para a conversa de origem.

Representação: seta tracejada, com fala do usuário prefixada por u:

Exemplo: usuário está modificando o período de visualização do extrato do banco e deseja desistir dessa alteração.



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa

Nome sugerido 1: User's Repair Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa

Nome sugerido 2: User's (Turn-Yielding) Repair Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que o nome proposto não é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



✉ Não compartilhado

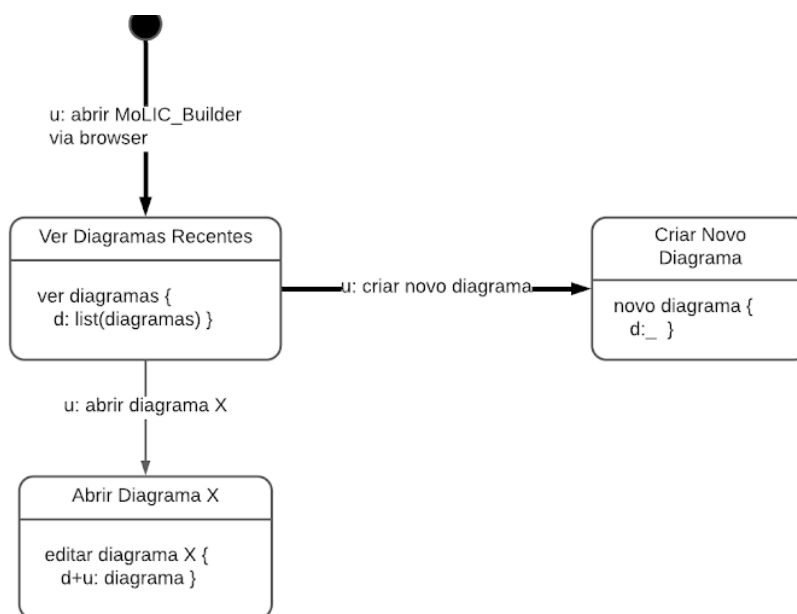
Elemento Proposto

Para representar o conceito de **conversa preferencial**, foi proposto o elemento a seguir:

Conceito: usado para indicar que uma determinada conversa é preferencial em relação às outras saindo de uma mesma cena.

Exemplo: atendendo ao desejo dos stakeholders, será apresentada com mais destaque na tela a oportunidade de fala de troca de turno para criar um novo diagrama (em vez de editar um antigo) pois converte mais usuários para a conta premium devido ao limite de arquivos editáveis na conta gratuita.

Considerando que este novo elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



Muito mal 1 2 3 4 5 6 7 Muito bem



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome sugerido: Preferred Conversation

1 2 3 4 5 6 7

Discordo totalmente



Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você não concorde com o nome desse elemento, **que nome você sugere?**
Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



✉ Não compartilhado

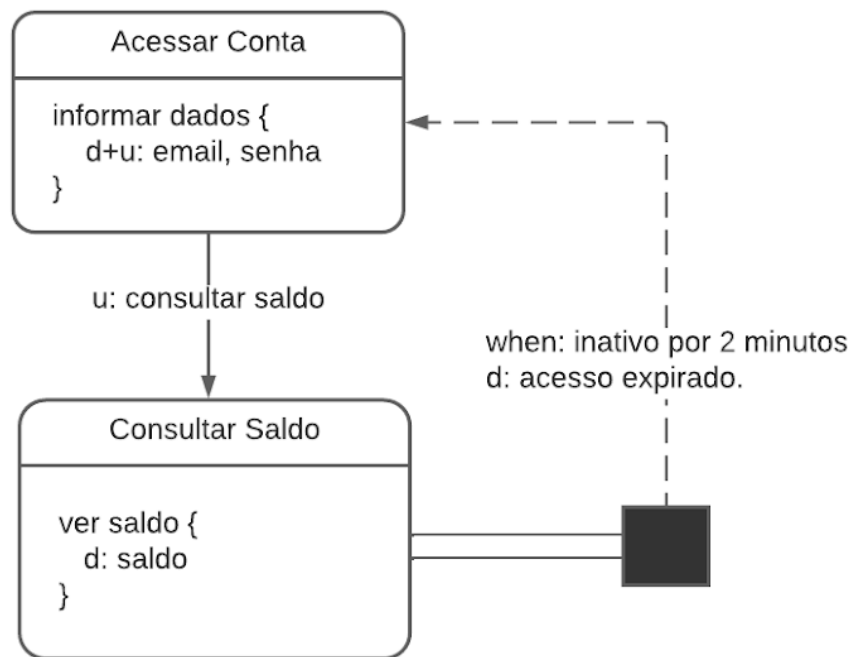
Elemento Proposto

Para representar o conceito de **ação associada a evento**, foi proposto o elemento a seguir:

Conceito: permite ao designer definir uma ação associada a um evento que ocorra na interação.

Exemplo: quando o usuário permanecer mais de 2 minutos na tela de consulta de saldo, ele será direcionado de forma automática para a tela de login.

Considerando que este novo elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



1 2 3 4 5 6 7

Muito mal

☐☐☐☐☐☐☐

Muito bem



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome sugerido: When

1 2 3 4 5 6 7

Discordo totalmente



Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você não concorde com o nome desse elemento, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



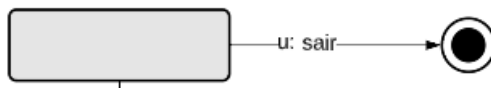
Não compartilhado

Elemento Renomeado

Na nova versão da MoLIC, o elemento a seguir teve seu nome alterado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito do Elemento: permite ao usuário mudar de tópico a qualquer momento da interação.

Exemplo: usuário está realizando uma transferência e, na mesma tela, há uma opção para sair do sistema.



O quanto você concorda com essa mudança de nome?

Nome Original: Ubiquitous Access

Novo Nome: Topic Shift

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta



Caso você ache que nenhum desses nomes é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



✉ Não compartilhado

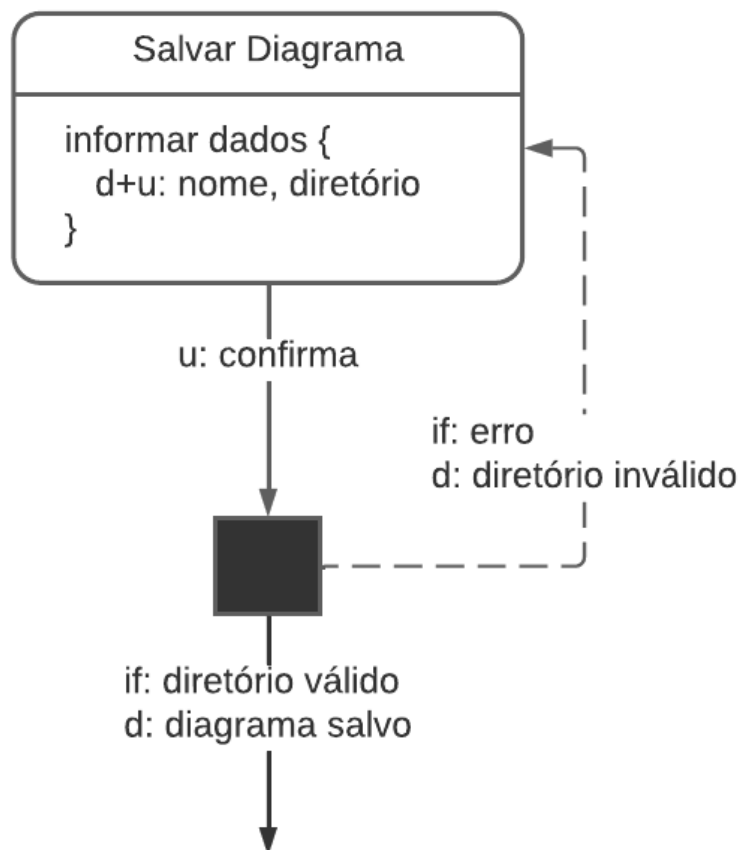
Elemento Renomeado

Na nova versão da MoLIC, o elemento a seguir teve seu nome alterado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito do Elemento: fala de troca de turno emitida pelo designer para guiar a conversa após uma ruptura na comunicação ocorrida durante um processamento do sistema.

Representação: seta tracejada com a fala do designer prefixada por d:

Exemplo: ao salvar um arquivo, o sistema retorna que o diretório selecionado não existe. Então, o designer irá alertar sobre a falha e direcionar o usuário para tentar novamente informando um novo diretório.



O quanto você concorda com com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome Original: Breakdown Transition Recovery Utterance

Novo Nome 1: Designer's (Breakdown) Repair Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

O quanto você concorda com com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome Original: Breakdown Transition Recovery Utterance

Novo Nome 2: Designer's (Turn-Yielding) (Breakdown) Repair Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que nenhum desses nomes é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



✉ Não compartilhado

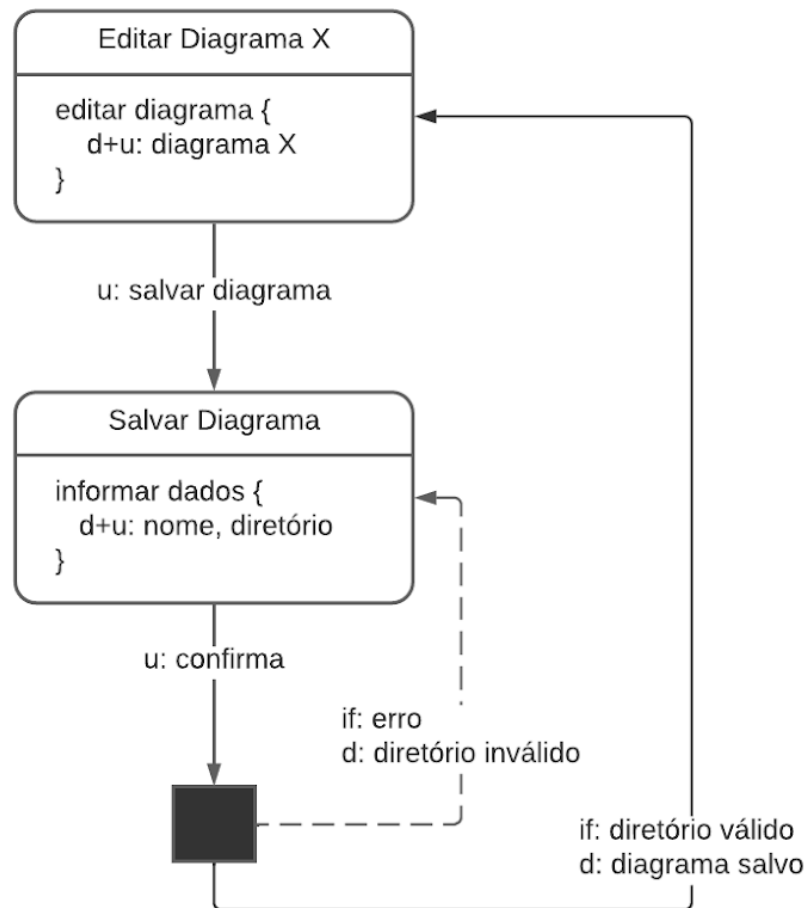
Elemento Renomeado

Na nova versão da MoLIC, o elemento a seguir teve seu nome alterado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito do Elemento: fala do designer para guiar a conversa após um processamento do sistema concluído com sucesso.

Representação: seta sólida com a fala do designer prefixada por d:

Exemplo: após salvar um arquivo com sucesso, o sistema (designer) irá informar a conclusão do salvamento e então redirecionar o usuário para outro ponto da aplicação: por exemplo, para a página de edição do diagrama salvo.



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome Original: Designer's Transition Utterance

Novo Nome 1: Designer's Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome Original: Designer's Transition Utterance

Novo Nome 2: Designer's Turn-Yielding Utterance

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que nenhum desses nomes é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Explique por que seria mais adequado.

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

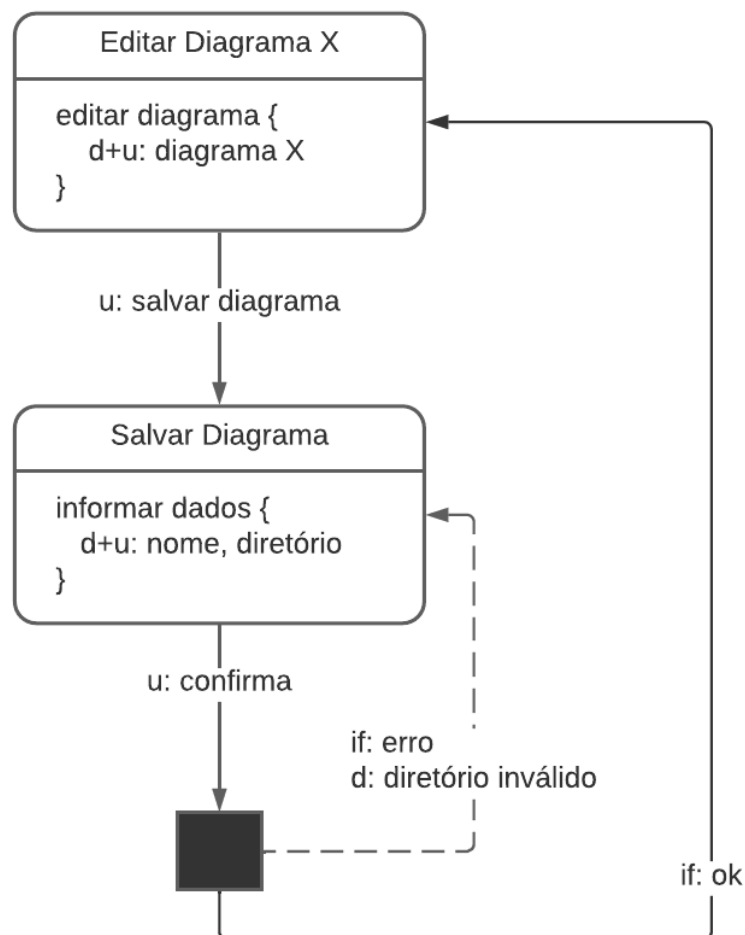
Elemento Nomeado

Na nova versão da MoLIC, o elemento a seguir foi nomeado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito do Elemento: troca de turno sem fala do designer associada que guia de forma silenciosa a conversa após um processamento do sistema concluído com sucesso.

Representação: seta sólida sem fala do designer; apenas a condição para a troca de turno.

Exemplo: Após uma transação concluída com sucesso, o sistema não irá apresentar mensagem de sucesso, apenas redirecionará o usuário para outro ponto da aplicação, por exemplo a página inicial.



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome sugerido: Silent Turn-Yielding

1 2 3 4 5 6 7

Discordo totalmente



Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que o nome proposto não é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Explique por que seria mais adequado.

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



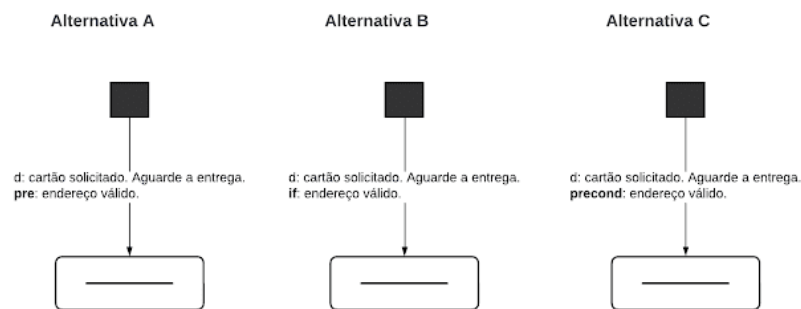
Não compartilhado

Elemento Renomeado

Na nova versão da MoLIC, o elemento a seguir teve seu nome alterado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

Conceito: usado para informar uma condição que deve ser atingida para que a conversa prossiga independente do emissor (usuário ou designer).

Exemplo: o cartão só será solicitado efetivamente se o endereço informado pelo usuário for válido.



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Alternativa A: pre

Discordo totalmente ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 Concordo totalmente



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Alternativa B: if

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Alternativa C: precond

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que nenhum desses nomes é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Elemento Renomeado

Na nova versão da MoLIC, o elemento a seguir teve seu nome alterado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

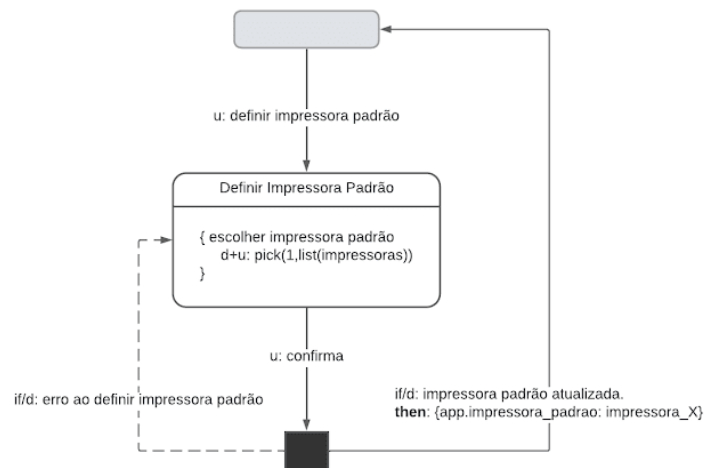
Conceito: indica consequências de uma conversa (podendo ser emitida pelo usuário ou designer) no estado do sistema (a nível de signo de contexto).

Exemplo: o usuário define, na tela de configurações, a impressora padrão a ser usada em todas as impressões futuras.

O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Alternativa A: Then

Alternativa A



1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

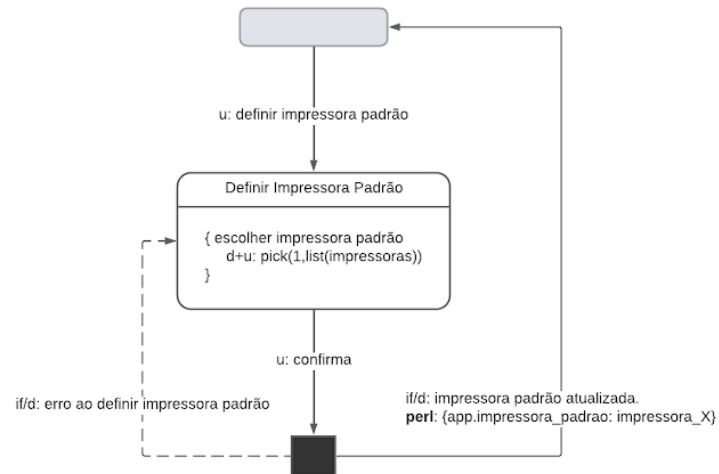
Concordo totalmente



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Alternativa B: Perl

Alternativa B



1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

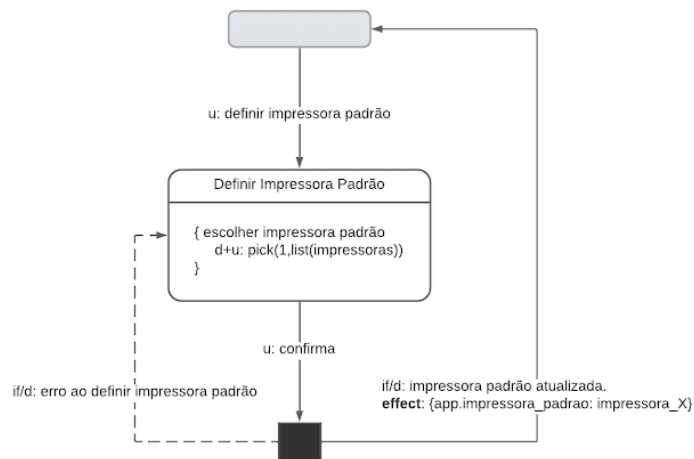
Concordo totalmente



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Alternativa C: Effect

Alternativa C



1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que nenhum desses nomes é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

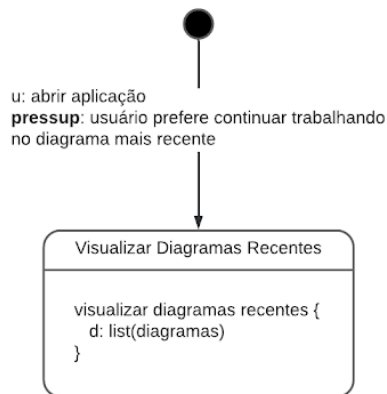
Elemento Renomeado

Na nova versão da MoLIC, o elemento a seguir teve seu nome alterado com o objetivo de alinhar a denominação do elemento com a metáfora de interação como conversa.

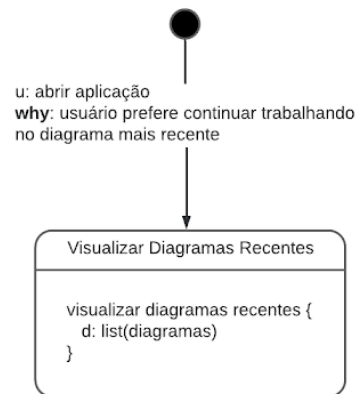
Conceito do Elemento: usado para registrar no diagrama o raciocínio por trás de uma decisão de design associada a um elemento do diagrama.

Exemplo: ao abrir o sistema de edição de diagramas, a primeira tela a ser apresentada é a de visualizar diagramas recentes, uma vez que o time de designers identificou que os usuários preferem continuar trabalhando no diagrama mais recentemente atualizado.

Alternativa A



Alternativa B



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome Original: Pressup (presupposition)

Novo Nome: Why

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente



Se puder, **fale um pouco mais** sobre a sua concordância/discordância.

Sua resposta

Caso você ache que nenhum desses nomes é adequado, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



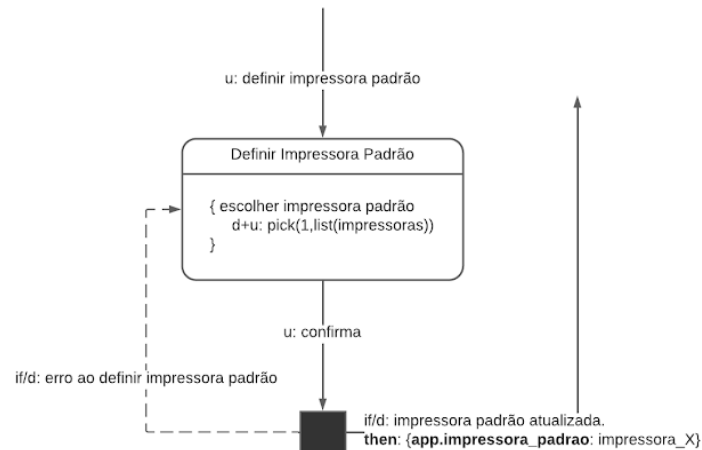
Não compartilhado

Elemento Proposto

Para representar o conceito de **especificar o contexto da conversa com granularidade mais fina**, foi proposto o elemento a seguir:

Exemplo: usuário define, na tela de configurações, a impressora associada para ser usada por padrão em toda a aplicação.

Considerando que este novo elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



Muito mal 1 2 3 4 5 6 7 Muito bem

☐ ☐ ☐ ☐ ☐ ☐ ☐

Os seguintes nomes foram propostos para este elemento: Articulated Sign e Context Feature. O quanto você concorda com o alinhamento de cada sugestão de nome para o elemento com a metáfora de interação como conversa?



Alternativa A: Articulated Sign

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Alternativa B: Context Feature

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você não concorde com o nome desse elemento, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



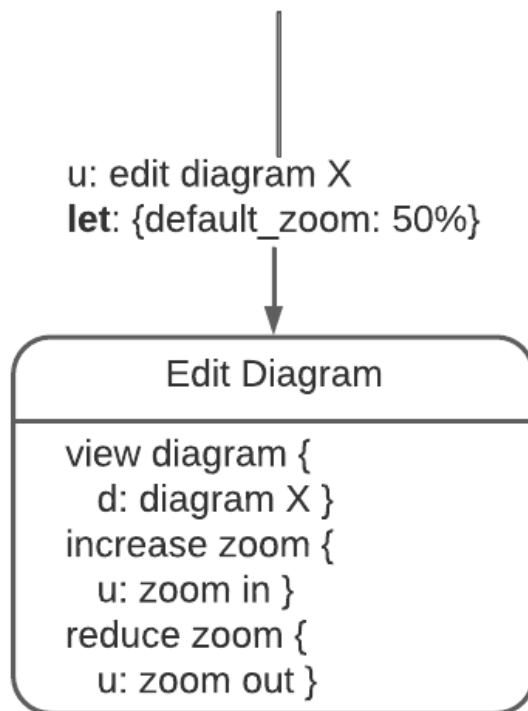
Não compartilhado

Elemento Proposto

Para representar o conceito de **especificar um valor inicial para um signo do contexto**, foi proposto o elemento a seguir:

Exemplo: sempre que o usuário entrar no site do banco, o período do extrato será de 15 dias, a não ser que tenha sido personalizado pelo usuário.

Considerando que este elemento deve colaborar com a expressividade da linguagem e estar alinhado à metáfora de interação como uma conversa, sem no entanto torná-la excessivamente complexa, o quão bem este elemento ilustra o conceito indicado?



Muito mal 1 2 3 4 5 6 7 Muito bem



O quanto você concorda com o alinhamento desta sugestão de nome para o elemento com a metáfora de interação como conversa?

Nome sugerido: Let

1 2 3 4 5 6 7

Discordo totalmente



Concordo totalmente

Se puder, **fale um pouco mais** sobre a sua avaliação.

Sua resposta

Caso você não concorde com o nome desse elemento, **que nome você sugere**, que estaria mais alinhado à metáfora de interação como conversa? Por que ele seria mais adequado?

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)

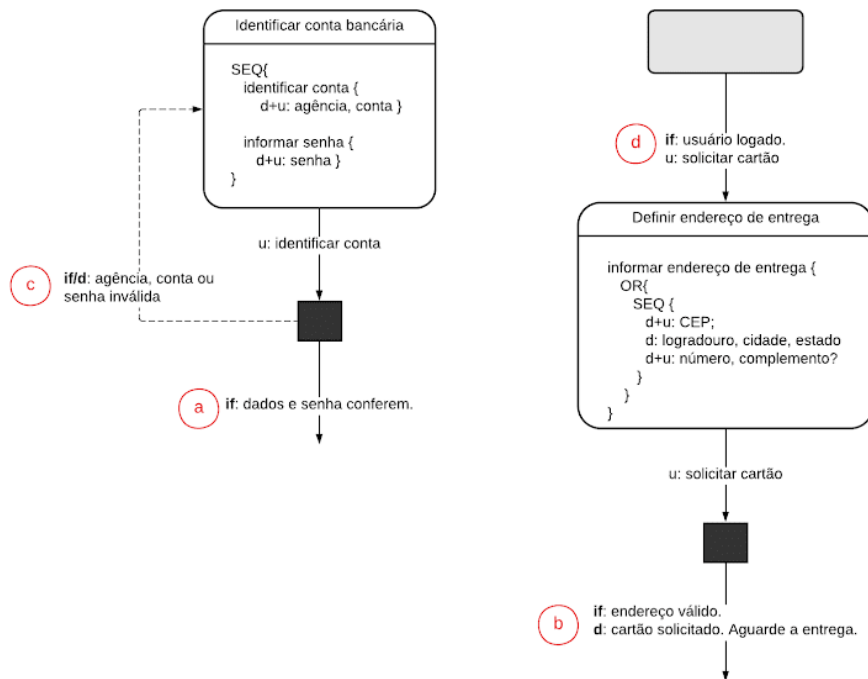


Não compartilhado

Condições e falas do designer e do usuário

Considere os casos a seguir. Para cada item descrito abaixo e ilustrado no diagrama, o quão adequado você considera cada cenário?

Itens a-b-c-d



Item a:

Existe uma condição para que uma troca de turno ocorra após um processamento do sistema, mas o designer não precisa fazer uma comunicação explícita sobre isso. Neste caso, a próxima comunicação do designer para o usuário será sobre a apresentação da cena de destino da "troca de turno silenciosa".

if: dados e senha conferem.

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

Item b:

Existe uma condição para que uma troca de turno ocorra após um processamento do sistema, e associada à troca de turno haverá uma fala do designer sobre o resultado do processamento.

if: endereço válido

d: cartão solicitado. Aguarde a entrega.

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

Item c:

Existe uma condição para que uma troca de turno ocorra após um processamento do sistema, que coincide com a fala do designer sobre o resultado do processamento.

if/d: agência, conta ou senha inválida.

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente



Item d:

Existe uma condição para que a fala do usuário seja enunciada.

if: usuário possui score de crédito suficiente para obter cartão

u: solicitar cartão

1 2 3 4 5 6 7

Discordo totalmente

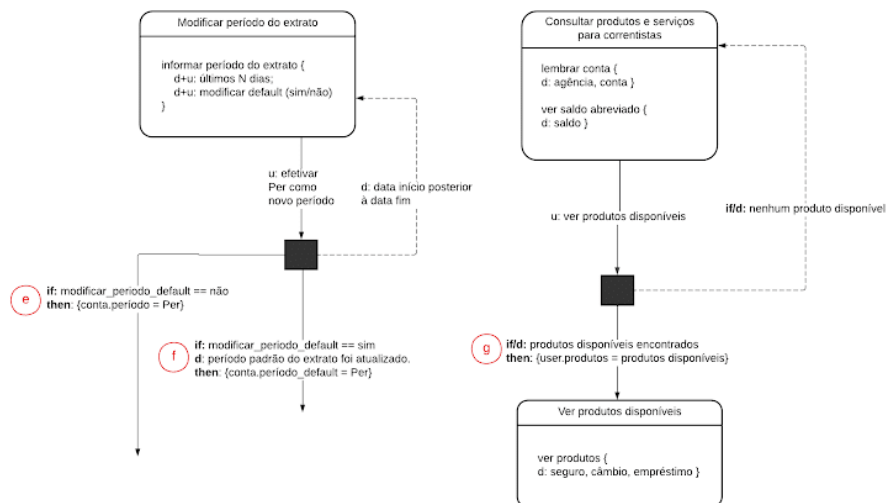
☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre as suas avaliações.

Sua resposta

Itens e-f-g



Item e:

Existe uma condição para que uma troca de turno silenciosa ocorra; essa troca de turno afeta o contexto da conversa.

if: modificar_periodo_default == não
then: {conta.periodo = Per}

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Item f:

Existe uma condição para que uma troca de turno com fala do designer seja emitida; essa troca de turno afeta o contexto da conversa.

if: modificar_periodo_default == sim
d: período default do extrato foi atualizado.
then: {conta.período_default = Per}

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Item g:

Existe uma condição para que uma troca de turno ocorra que coincide com a fala do designer; essa troca de turno afeta o contexto da conversa.

if/d: produtos disponíveis encontrados
then: {user.produtos = produtos disponíveis}

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre as suas avaliações.

Sua resposta

[Voltar](#)

[Próxima](#)

[Limpar formulário](#)



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Estruturação dos diálogos de uma cena

Considere a cena abaixo contendo 3 diálogos: (1) lembrar conta; (2) informar conta de destino; e (3) informar valor a transferir. O quanto você concorda com as afirmativas abaixo sobre como a cena deve ser interpretada?

Informar dados para transferência

```
lembrar conta {  
  d: agência, conta  
}
```

```
informar conta de destino {  
  d+u: agência e conta  
}
```

```
informar valor a transferir {  
  d+u: valor  
}
```

Todos os diálogos devem ocorrer.

1 2 3 4 5 6 7

Discordo totalmente

☐ ☐ ☐ ☐ ☐ ☐ ☐

Concordo totalmente



Nenhum diálogo precisa ocorrer.

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

Algum dos diálogos deve ocorrer.

1 2 3 4 5 6 7

Discordo totalmente

☐☐☐☐☐☐☐

Concordo totalmente

Se puder, **fale um pouco mais** sobre as suas avaliações.

Sua resposta

Voltar

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



Questionário sobre a MoLIC

loppi.clg@gmail.com [Alternar conta](#)



Não compartilhado

Declaração de interesse em participar de entrevista sobre a MoLIC

Caso você tenha interesse em participar de uma entrevista sobre a MoLIC, deixe seus dados de contato a seguir.

Nome:

Sua resposta

E-mail:

Sua resposta

Afiliação:

Sua resposta

[Voltar](#)

[Enviar](#)

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários

