Interaction Design and Architecture(s) Journal - IxD&A, N.55, 2022-23, pp. 139 - 173

Lessons Learned from Modeling the Interaction with Conversational Agents

Ulisses da Silva Fernandes[1], Glívia Angélica Rodrigues Barbosa[1,3], Bruno Azevedo Chagas[1], Gabriel Diniz Junqueira Barbosa[2], Simone Diniz Junqueira Barbosa[2], Raquel Oliveira Prates[1]

[1] Department of Computer Science, Federal University of Minas Gerais, R. Reitor Píres Albuquerque, ICEx - Pampulha, Belo Horizonte, MG, 31270-901, Brazil;
[2] Department of Informatics, PUC-Rio, R. Marques de Sao Vicente, 225, Gavea, Rio de Janeiro, RJ, 22451-900, Brazil
[3] Computer Department, Federal Center for Technological Education of Minas Gerais, Av. Amazonas 7675, Nova Gameleira, Belo Horizonte, MG, 30510-000, Brazil
{ulissesfernandes, bruno.chagas, rprates}@dcc.ufmb.br; glivia@cefetmg.br; {gbarbosa, simone}@inf.puc-rio.br

Abstract. Intelligent conversational agents have become widespread. Inspired by conversations in natural language, they present different degrees of intelligence and autonomy, bringing challenges for Human-Computer Interaction (HCI). One such challenge concerns design languages for modeling user-agent interaction. We focus here on MoLIC, a design-phase dialogue model based on Semiotic Engineering theory devised to represent user-system interactions as conversations. We performed two case studies with MoLIC interaction diagrams representing two conversational agents – the ANA chatbot and Samsung Bixby. We examined how the interactive aspects of these agents could be expressed in MoLIC. Although it was possible to express the general interaction, our results showed limitations related to the language expressiveness or its inadequacy to represent these systems. We identified limitations in the applicability of MoLIC in modeling and pondered on how to extend or adapt it; directing the HCI community to issues and initiatives that can help design and model these technologies.

Keywords: Intelligent Conversational Agents, · Modeling · Chatbots, · Virtual Assistants, · MoLIC

1 Introduction

Conversational agents, such as chatbots or virtual assistants, have become a topic of increased interest in recent years. Triggered by text or voice commands from users, they are able to respond accordingly in different types of conversations. In healthcare, for instance, they can be used to provide personalized care, having the advantage of being available 24/7, and thus reducing service and waiting times for patients who need more urgent care [17], [26]. In addition to answering user requests, conversational agents can also possess some level of autonomy, even if minimal, to assist and even lead users at interaction time [26]. These technologies can be reactive or proactive, based on user inputs or changes in the environment. Some of them are also adaptive and capable of learning to handle contextual information or consider user preferences in future dialogs [47].

The rise in popularity of conversational agents and virtual assistants offers opportunities and challenges to Human-Computer Interaction (HCI). The hidden structure behind this kind of technology can make it more difficult for an expert to explore the entire system, since there are many more interactive paths available in a conversation than in a GUI (Graphical User Interface) [45]. Furthermore, there is a need for further investigations on whether existing HCI approaches apply to modeling and evaluating communications between users and conversational agents, considering the very nature and mode of conversational interaction, the level of intelligence, and possible autonomy of these technologies during the interaction. For instance, Følstad and Brandtzæg suggest that we should change our view of design as an explanatory task (*i.e.*, telling the user what content or resources are available and what steps to follow to complete a task) to an interpretive task (*i.e.*, understanding users' needs and how they can be met) [14].

As this is an emerging topic, it can be explored in different perspectives and theories. We chose Semiotic Engineering because of its focus on communication at two different levels – designer-to-user metacommunication and user-system interaction –, which makes it especially attractive to analyze conversational systems, like chatbots. Semiotic Engineering is an explanatory theory that perceives human-computer interaction as a metacommunicative process between designers and users through the system. It offers methods and tools to assist in the design and evaluation of interactive technologies. MoLIC (Modeling Language for Interaction as Conversation) is one of Semiotic Engineering's tools to assist the designer in modeling system interactions as a conversation, explicitly defining the metacommunication [7]. This conversation takes place between the user and the designer's deputy (*i.e.*, system interface). MoLIC's main goal is to support designers' decision-making on interaction design. The diagrammatic representation of the conversations between users and the designer (through the system) can also help structure the designer's conception of the system as a whole [43].

Even though MoLIC models interactions as communications at an abstraction level that is independent of specific implementations, conversational agents such as chatbots and intelligent virtual assistants introduce changes that go beyond interface design. This is because the intensive use of natural language processing (NLP) and machine learning techniques in these systems can cause changes in the entire flow of interaction and communication between users and the system [41]. Furthermore, these systems may even present different degrees of autonomy, taking the initiative to talk to the user in some situations, instead of always waiting for the user's input as happens with more traditional (GUI-oriented) interactive technologies. Given this context, it would be interesting to analyze how MoLIC would represent this type of systems, since there is a lack of works that model conversational agents using this modeling language [8].

Although the metacommunication process is a problem in all design activities, we are interested in reflecting on it in the design of conversational agents. Thus, the goal of this work is then to investigate interaction modeling of conversational agents, using MoLIC as a representative of interaction modeling languages. We chose MoLIC because the language is quite consolidated and has been used to model different types of systems. We were interested in evaluating whether the interaction-as-conversation metaphor in which MoLIC is grounded would also prove valuable for modeling chatbots. The main focus of this work is to analyze whether or not MoLIC can be used to express the interaction proposal (i.e., conversations) between the user and a conversational agent. Furthermore, we have not found in our research other models that allow designers to focus on the design of the conversation process (see section 3.3), without having to detail the syntactic and lexical issues of the utterances. We therefore do not enter a discussion level about how the interface will be designed (in terms of which signs will be used) for the interaction to take place.

For our analysis, we conducted modeling sessions on two existing agents: ANA, a chatbot developed to aid in COVID-19 prevention and care; and Bixby, an intelligent assistant developed by Samsung Electronics. Fernandes et al. [12] conducted a preliminary analysis of users' interaction with ANA using MoLIC and identified potential limitations of modeling interaction of conversational agents. The study presented here goes beyond their initial findings, not only analyzing a general purpose, presumably more intelligent, conversational agent (Bixby) and identifying an extended set of problems, but also conducting an in-depth analysis of the role of a modeling language in helping to uncover those problems. This analysis revealed key issues that point to requirements any modeling tool would need to fulfill in order to express a set of typical situations we identified when modeling the interaction with conversational agents. Through these two case studies, we identified elements that are useful for modeling these kinds of systems, as well as limitations to MoLIC's expressiveness and ability to represent certain aspects of interactions with conversational agents. Although we limited our analysis to two case studies, we were able to find enough reflection points to corroborate our choice of MoLIC to support our analysis and to suggest necessary adaptations to the language going forward. We left the analysis of a wider range of agents for future work.

Our main contributions are insights and reflections that can direct the HCI community toward new initiatives to model interaction in conversational systems. In particular, we used MoLIC to identify and characterize limitations and problems in terms of the model's expressiveness and discussed some needs and possibilities to adapt or extend it for this context. Our work takes an initial step to organize the largely unexplored problem space of interaction modeling of conversational agents, contributing not only to the extension of MoLIC, but also to other modeling languages. The four groups of limitations we found reveal design issues that point to potential requirements that any notation proposed for this domain should address.

This study is structured as follows: Section 2 presents the theoretical background about Semiotic Engineering and introduces MoLIC; Section 3 reviews the literature to highlight some of the HCI challenges in the design and evaluation of intelligent conversational assistants and discusses the state-of-the-art of MoLIC's applicability; Section 4 presents our study's methodology; Section 5 presents the generated models; Section 6 contains the findings and discussion about these results; Section 7 contains the study limitations; and, finally, in section 8, we present our conclusions.

2 Theoretical Background

This section presents some important concepts and works that serve as the theoretical background for our contribution. We briefly present the Semiotic Engineering theory, as well as the MoLIC modeling language, with all of its core elements which is the focus of our work.

2.1 Semiotic Engineering

Semiotic Engineering (SemEng) [9] is a communicative theory of HCI that seeks to explain the nature of aspects involved in the design, use, and evaluation of interactive systems based on Semiotic theory. It studies theoretical and technical conditions that affect user interaction with technology based on how interactive systems are developed. It considers every system as an intellectual artifact that is the result of a designer's interpretation of a proposed problem and their understanding of the most appropriate solution thereof. The designer's intent and understanding is encoded and translated into the system's interface, which will communicate with the users on the designer's behalf during interaction time.

The foundations of SemEng come from the field of Semiotics, which studies the phenomena of communication and signification. At its core is the *'sign'*, which is *everything that means something to someone [27]* and relates an object to its representation (pictures, sounds, words, etc.), which produces an interpretation that is created in the mind of the person reading this information [9].

To Semiotics, signs are basic units that mediates all human signification processes, including creating signs for representing objects and expressing ideas, and using signs to communicate them to others. Therefore, Semiotics has a dual focus on signification, *i.e.*, ascertaining the meaning of signs, and communication, *i.e.*, using signs to communicate meanings to others [9].

At its core, Semiotic Engineering focus on studying the signification and communication processes on interactive systems, both in design and use time.

SemEng frames interaction between the user and the system as a metacommunication process, *i.e.*, a communication about the communication itself. In this sense, interaction is a user-system communication that also comprises a communication between the designer and the user where the system's interface conveys to the user the designer's interpretation about whom the system is intended for, what goals they can achieve through the system, and how to interact with it. Receiving this message also requires some effort on the user's part, since they must continuously interpret it as they interact with the system. The system and its interface, which contains the signs that communicate the designer's vision, is seen as the designer's *deputy* or *proxy* [9].

The interface design process is then a metacommunicative process where the designer must first understand the problem, then evaluate possible solutions, and finally make decisions regarding the final solution and how to encode these decisions into the system, translating them into interface signs that compose the "final" metacommunication artifact. By reflecting on how this process takes place, designers engage in what Donald Schön calls *reflection-in-action [36]*. In this paradigm, the designer engages with epistemic tools that allow them to raise hypotheses, experiment with other possible solutions to the problem, and evaluate their results. One of the tools Semiotic Engineering proposes to support designers to reflect on the various issues concerning the metacommunication artifact and to compare proposed solutions is an interaction modeling language called MoLIC.

2.2 MoLIC

The Modeling Language for Interaction as Conversation (MoLIC) is an epistemic tool that supports interaction design as the modeling of a conversation between the user and the designer's deputy (*i.e.*, the system) through its interface [1], [7]. In other words, it is a language for modeling human-computer interactions as conversations. Grounded in Semiotic Engineering, it allows the representation of all the possible paths that conversations between the user and the system may follow, including user-system expected dialogues, alternative ways to achieve a goal and ways to recover from communication breakdowns. It bridges a gap between the task model at the user goal level and the interface representation level (*e.g.*, the graphic design in a GUI), allowing designers to model what and how users will interact with the system without committing to a particular interface design [1]. MoLIC is mainly focused on communication and supports the modeling of the usage situations envisioned by the designer, including points where there may be problems during the user's interaction with the technology. The designer represents the possible interactive paths in a diagram showing the scenes and dialogues that will be available in the system for the users to achieve their goals. Figure 1 on the left (I) shows an example of a MoLIC model of a banking system, showcasing the main elements that compose the language [7].

A scene is a stage of the conversation between the user and the system (*i.e.*, the *designer's deputy* talking on behalf of the designer) about a specific topic. A scene is represented by a rectangle with rounded edges (a). The scene's topic is represented by a sentence at the top (b), which is a high-level message from the designer's deputy to the user. The verb in infinitive form indicates what should be done at that moment, as in the example "Enter data for bank transfer" in the user's perspective. The scene content in (c) contains the scene dialogues, which are units of conversation that focus on the different subtopics or parts of a scene, such as entering various data (target account, amount to transfer, etc.). The user transition utterances in (d) represent a turn-taking in which users pass the control to the designer's deputy indicating that they want to move to a new scene, as in the example. User utterances are represented by solid directed lines alongside the tag "u:", for example: "u: proceed". The black box (e) represents internal system processes, which are hidden from the users who should wait for a requested operation or process to be completed by the system (users cannot see under the "interface hood"). In (f) we have a designer's deputy utterance, which may represent its intent to change scenes, pass the control over to the user, or inform the user of the result of an internal process. They are represented by solid directed lines labeled with a "d:" that stands for "designer", for example: "d:

everything looks ok to me". In (g) we have an utterance from the designer's deputy for recovering from a breakdown by informing the user of an unexpected process' outcome and passing the control back to the user. These are represented by directed dashed lines labeled "d:". Finally, in (h), we have the user's utterance to recover from a breakdown, for instance when users realize they had made a mistake and want to return to a scene in which they can rectify part of the conversation, for example, "u: change data". These are represented by dashed directed lines labeled "u:" [7].

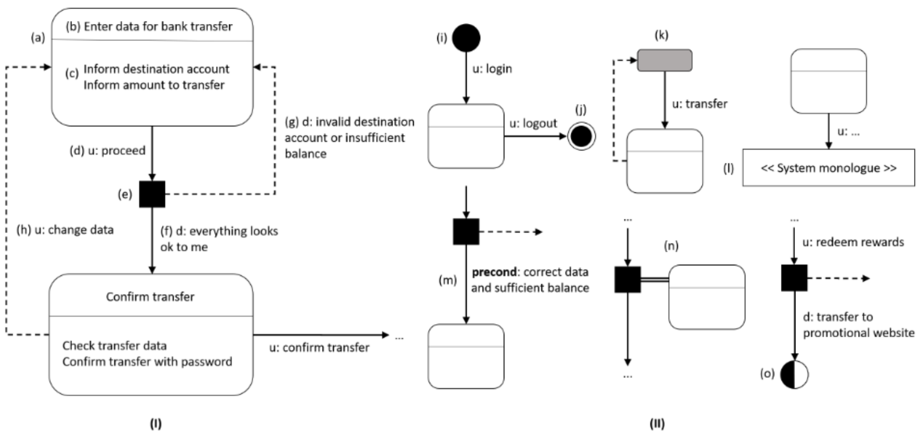


Fig. 1. (I) Modeling example using MoLIC. (II) Other elements of MoLIC. Figures adapted from [7].

Figure 1 on the right (II) shows other MoLIC elements [7]: in (i) we have the starting point of the conversation while in (j) we have the ending point. In (k) we have an ubiquitous access, which means "in any scene where users are, they can enunciate <<talk>>. In this example, ubiquitous access indicates that from any point in the system the user can say "u: please, make a transfer", which will take him to the scene related to transfers. It is important to note that once the conversation is started, users can change their minds and give up on the transfer, returning to the scene where they were before. In (l), there is a system monologue, which is used for alerts and messages where the system says <<something>> and no answer from the user is expected or needed. In (m), we have the keyword "**precond**" which indicates the conditions that must be met for the related transition to take place. In (n) there is a synchronous scene, which means that during the processing to which it is coupled, the system informs the user about the processing itself (*e.g.*, the progress), and may give the user an opportunity to interfere (*e.g.*, cancel) the processing. Finally, in (o) we have the representation for an external interlocutor which represents another system that will be triggered or affected by the system being designed.

It is important to note that the utterances in MoLIC represent the content of the utterance, not its expression. For instance, the utterance "u: please make a transfer" represents any concrete utterance with similar content, and not a requirement for nor a specification nor of that exact expression. There are countless ways to map a MoLIC utterance onto the concrete user interface, e.g., in a GUI, clicking on a toolbar button, menu item or link with one of a range of labels, such as Transfer, Make transfer etc.; in chatbots, writing “I’d like to make a transfer”, “Make a transfer”, “I want to transfer money”, just to name a few. In other words, it is not the goal of MoLIC to specify the different expressions allowed for each piece of content at the interface level.

3 Related Work

In this section we present previous works related to our research. First, we present work regarding studies that have investigated MoLIC’s consolidation as an interaction model, applicability to different contexts and proposed extensions. The other two subsections focus on conversational agents, presenting the new challenges involved with interacting with them, and studies that have investigated how to model them.

3.1 MoLIC´s Consolidation and Applicability

As de Carvalho et al. [8] explain, MoLIC is a consolidated tool that has already been used in various contexts for different purposes. According to them, among the use cases in the literature, we can divide these studies into two groups: (1) those that focus on MoLIC itself, and (2) those that include but do not focus specifically on MoLIC. In the first group, the modeling language was used in six different subgroups: (1) studies where the focus was on presenting the tool; (2) studies with a focus on proposing changes and revisions; (3) studies that evaluate or present case studies that use MoLIC; (4) discussions of a new context, purpose or situation for using MoLIC, e.g. use of MoLIC to facilitate communication between HCI and Software Engineering areas; (5) studies that seek to extend the tool; and (6) studies that propose some methods, tools or techniques that applies to MoLIC and its use. Similarly, the second group can also be divided into six subgroups: (1) studies that cite MoLIC; (2) studies that use it for modeling; (3) studies that compare it with other modeling languages; (4) studies that use it for teaching and creating educational material; (5) studies that use it as part within a larger model; and (6) other types of studies (papers not classified in any of the above categories). Although there is a constant interest of researchers in the tool, showing its adoption in several research projects, we can observe that, among the works cited by de Carvalho et al. [8], there were no studies that address the use or applicability of MoLIC for the context of conversational agents, such as chatbots or intelligent virtual assistants.

Marques et al. [20] reported an empirical study comparing MoLIC to another modeling tool, namely the CTDM (Comprehensive Tasks Interaction as a Conversation), with respect to their support for modeling interactions. Differently from MoLIC, CTDM focuses on representing tasks that a user can perform in a system, with notations that represent dialogues between the system and the user. In the study, 60% of the participants pointed to MoLIC as more useful and easier to use for interaction design. Moreover, 56% of them indicated MoLIC as being the most complete model compared to CTDM. The authors further explain that this preference may have been due to the fact that MoLIC represented more aspects of the interaction than CTDM, which is limited to tasks and transitions. MoLIC also allows designers to think about conditional and alternative interactions in a simpler way. This work shows the potential of MoLIC as an interaction design tool, highlighting some of its advantages. The combination of a conversational approach and level of abstraction can make MoLIC particularly suitable for the design of conversational agents, justifying our choice of further investigating MoLIC's applicability in this context.

A different line of study was made by da Silva and Barbosa [38], who investigated some of MoLIC's limitations for representing interactive collaborative systems. Through a case study on a system called "NiTA" (Notes in the Air), a multi-user asynchronous communication application for mobile devices, some extensions to MoLIC were proposed, such as: dialogue structuring, transition utterances preconditions, and the relationship between interaction models designed for different users. According to the authors, by defining dialogues between external users in MoLIC, they have taken another step towards modeling interactions with group applications, enabling MoLIC to represent interactions on an interpersonal level. In this same direction, [39] proposed a new extension to MoLIC to extend its expressiveness to collaborative systems, denominated MoLICC [39], [40]. MoLICC incorporated three new elements to the original MoLIC – a shared space indicator, incoming message indicator and outgoing message indicator. These works have similar goals to our study, since they investigated limitations in MoLIC in order to find opportunities for improving and extending it. More recently Ferreira et al. [11], have discussed and proposed an adaptation in MoLIC to model human-AI interactions with intelligent user interfaces. However, their focus was on generating and adapting interactions in graphical user interfaces and not interpreting users' input, which is a strong component of conversational agents. Moreover, their proposal is only preliminary and has not been evaluated yet.

3.2 HCI Challenges brought by Conversational Agents

Interacting with conversational agents is significantly different from interacting with other kinds of technologies and presents new challenges to interaction design [14], [2], [28], [13]. The focus shifts from the design of interfaces and functional interactive elements to the design of the conversations that the agent can hold with its users [14].

According to Valerio et al. [45], a major difference between traditional GUI interfaces and conversational agents is the way the technology presents itself to the user. In the former, there are several buttons and menus that the user can click on and choose from. In the latter, the difficulty lies in the fact that the technology is presented to the user phrase by phrase, step by step. Furthermore, conversational interfaces are also more open to variations of user inputs since the range of users' expressions in natural language is unconstrained, opening a larger set of possibilities and interactive paths that the user can follow. Instead of a limited number of buttons, inputs and other interactive elements for the user to click, a user can say anything at any given point during a conversation, which naturally amplifies the possibilities of unexpected inputs, errors and breakdowns. Therefore, there is a big challenge to model conversations in a way that the user can fully grasp what the agent is capable of doing and how to make it respond accordingly to the user needs, expectations and "deviations", in the sense of inputs that the agent is not capable of understanding or does not know how to respond.

In addition, there are challenges related to the humanness and sociability of agents. Among these challenges we can mention: designing the right "personality" and interactive style that best fits the agent's purpose, context and goal [31], [15], [41], [21]; being able to adapt and serve different user profiles [2], [13]; applying personal touches to communicate in an engaging way with diverse types of users [21], [42]; building up trust, empathy, and rapport; ensuring ethics and privacy in agent development and deployment [28], [13]; and certainly, many more.

Moreover, when new situations that have not been foreseen by the developers or that the agent has not learned yet arise, it can get confused and provide incorrect information to users. Issues concerning the understanding and generation of texts by the agents still require further investigation, with improvements in natural language processing algorithms, in their ability to learn, and in their ability to adapt to different conversational contexts properly [13].

Finally, agents are usually integrated with other software-enabled services [13] and can act as a conversational touchpoint for accessing these services [14]. For instance, booking a flight through one can be only one of multiple interfaces for a flight booking service, which might also be accessed through a website, an app, a store/travel agency, etc. In this sense, it is important to consider issues related to the technologies, platforms and architectures used in conversational agents development since they are crucial to service integration and can therefore significantly influence the user experience [21].

All the challenges that we have briefly discussed above point to the need of advancements in the interaction design of conversational agents. In order to overcome them, designers of this particular kind of technology need the right knowledge and tools that can help them in their design process. A book organized by Moore and colleagues have discussed the concept of conversational user experience (UX) design and some of the several nuances and complexities of interaction design with conversational agents [25]. According to Moore et al. [23], conversational UX is determined primarily by the choice of words, style and sequence of utterances, which are processes where NLP and AI alone are insufficient, at least to date [23]. According to them, conversational interactions must be carefully designed and engineered by means of a "conversation first" approach that focuses primarily on the sequence and structure of utterances and turn-taking between agent and user since interface visual elements play only a secondary role. This work follows the same approach for studying ways to support designers to design better conversational agents. In particular, we focus on the application of a commonly adopted design technique, namely interaction modeling.

3.3 Conversational Agents Interaction Modeling

Despite the increasing interest in research on interaction with conversational agents in recent years [31], [1], there is no consolidated approach or consensus on how to design interactions in this domain. Several challenges exist, as we briefly saw in the previous section. As the technology becomes more and more popular and adopted in actual systems, researchers and practitioners try to provide designers with knowledge and practical instruments to support the several aspects and tasks involved in designing conversational agents (*e.g.*, [37], [24]). One of the approaches yet to be explored is interaction modeling. Interaction modeling can help designers explore solutions, devise alternatives, and reflect on decisions prior to implementation [1]. However, research on interaction models of conversational agents is scarce, since most works we found tend to investigate technical frameworks and AI techniques for supporting the development of those agents.

A frequently found type of work is exemplified by Wachtel et al. [46], who present guidelines on how to model a dialog-based conversation with a chatbot and a framework built with Microsoft Conversational Services. They created a framework that manages user dialogs and sessions. For this, they used a systematic classification of inputs and intentions through the use of LUIS (Microsoft Language Understanding Intelligent Service), a language processing and interpretation framework developed by Microsoft that allows developers to quickly enhance existing programs with natural language controls. The framework that the authors have created provides functionality for modeling dialogs as well as managing sessions with little amount of code and reducing the complexity of different tasks during the chatbot development process.

In a similar vein, Pérez-Soler et al. [29] propose to automate the design task and agent creation, using a dynamic modeling service based on a meta-model. Guzzoni et al. [16] presented a novel architecture for developing intelligent assistants that provides a unified tool and approach for rapid application development incorporating natural language interpretation, dialog management, plan execution, and web service integration. Pérez-Soler et al. [30] propose a web IDE, CONGA, that provides neutral, domain-specific language (DSL) for chatbot modeling that is able to build functional bots and migrate agents' logic between different tools such as Rasa or Dialogflow. Valtolina and Neri [44] propose a platform to support domain experts in creating bots using End-User Development (EUD) strategies by means of a graphical conversational flow editor.

Internally, conversational agents are made of a combination of natural language processing (NLP) and/or natural language understanding (NLU) engines, dialog management, response generation, user interface, and integration with knowledge bases and external systems [1]. Therefore, their development is complex, heavily based on machine learning and demands a mix of AI training and coding. The works cited above go in the line of alleviating the development workload. Even when targeted to domain experts, their goals are oriented towards supporting the technical development and deployment of agents. In DialogFlow [33], for instance, interaction modeling is coupled with the development process. This may hinder the designer's ability to explore interactive alternatives and solutions, as they need to deal primarily with NLP (basically, intent classification and entity recognition) while designing the

UX implicitly through the dialog sequencing. Moore [22] proposes another level of concern UX designers should consider for a better UX in conversational agents, which he calls Natural Conversational Framework for modeling the basic structure of natural conversations. In a similar vein, we chose to investigate modeling the interaction alone as it may help designers of conversational agents focus more freely on a "conversation-first" approach [23] in favor of better conversational UX.

An interesting work was done by Castle-Green et al. [5] addressing the differences between rule-based and corpus-based (stochastic) conversational agents. Rule-based ones are created by mapping and designing the possible predicted interactive sequences that can occur between systems and their users. Each path is dictated by specified rules, created from a decision tree. Although simpler, rule-based agents are the most common (such as ANA, one of the agents we use in our case studies). However, they tend to be more fragile and require a large amount of information to model all forms of interaction, necessitating the use of a decision tree. In contrast, corpus-based, also called stochastic, chatbots are based on the use of training data from similar domains to learn and generate probabilistic responses based on an algorithmic model, while also relying on search algorithms to select responses from a dialog graph. This type of agent can be highly flexible in terms of its interactions. However, they rely on training data, not providing the designer much control over the user experience and arrival at specific navigation points. This paper sought to provoke thought about some of the challenges in using branched structures when designing conversational agents and to indicate promising next steps in the research area: (1) analyzing the role of diagramming with its benefits and implications; (2) understanding the impacts that assumptions have on the designed interfaces and exploring other options, while recognizing any constraints branched diagrams present; (3) reconsidering design metaphors to suit both conversationally sensitive design in general and the user experience goals of the specific agent. For example, rather than thinking about paths and branches, a designer could think about flows and topics. However, the study does not present a proposal for some new tool or modification of an existing one that can model the interactions of current conversational technologies.

Focusing on the interaction, Carlmeyer et al. [4] present a first interaction model for incremental information presentation, including different timing strategies for the presentation of the next piece of information in a given task. These strategies can be selected based on the availability of sensors in the smart home ambient system or the inherent requirements of the task at hand. The results of this study led to the formulation of the interactive and incremental model presented in the research. The model distinguishes between the dialog and task levels, allowing for an overall description of the interaction. It also includes different modalities of user input, as well as being able to monitor task progress. One current limitation is that in case of failure in a sub-task, the entire interaction fails. In our view, it is important to use a modeling tool that provides an entire conversation view with the conversational agent without missing any specific sub-tasks, which could compromise the modeling.

In a similar vein, Cambre and Kulkarni [3] consider how researchers and designers can build novel and intuitive voice interfaces. They briefly describe two of the main voice design techniques, elicitation methods and Wizard of Oz, and then address the tools available for prototyping and implementing voice interfaces, characterizing them according to their function, ease of use, and the fidelity of the prototypes they can produce. They also describe three main aspects of voice communication that are not yet supported by existing prototyping tools, which are: (1) long-term interaction – managing data from previous conversations over days, months or years; (2) Turn-taking behavior – today's voice assistants need a structured conversation pattern where the user and the assistant alternate in turns and where there are words that wake up the assistant like "Hey Siri" for example. Dealing with conversation interruptions and conversations without the need for wake-up words will change the nature of interactions; (3) paralinguistic communication – interfaces should be able to recognize the various elements of communication other than words, such as tone of voice, emphasis or sentence speed. Although the authors study voice design techniques and point out opportunities for future studies, they do not present a way to model interactions of conversational voice agents or other types of conversational agents in general.

Moore proposes a framework for conversational UX design called Natural Conversation Framework (NCF), which is grounded in Conversational Analysis and provides a set of resources for supporting the design of conversations with agents [22]. The framework consists of an interaction model, a library of reusable conversational patterns, a method for navigating conversational interfaces, and a set of performance metrics. Furthermore, he also proposes the use of *transcripts* to represent and communicate conversational designs as simple samples of representative dialogs in plain text, as in a movie script. Although it does appear to be powerful and helpful at a lower level of abstraction of conversational design, his approach does not cover a higher level of abstraction in interaction design, and does not provide an overall view of what could or should happen during interaction. We see that the utterances and sequences are analogous to concrete interface design and the use of MoLIC can help at a higher level of abstraction, when designers are still reflecting about major use cases, goals that an agent should fulfill, and the best paths to reach them.

As we have seen in the literature presented in this section, most existing studies about models of conversational agents are oriented towards facilitating the implementation and do not focus on interaction. As a result, the interaction model is implicit and coupled with the underlying technology (*e.g.*, AI and NLP machine-learning models). As we explained in the text, an interaction model decoupled from implementation constraints can help design the behavior of conversational agents.

As presented in section 2.2, MoLIC has been proposed as an epistemic tool grounded on Semiotic Engineering that allows designers to consider the overall user-system interaction by modeling the possible interactive paths that users can take. MoLIC does not purport to be the only model capable of modeling interaction with conversational agents. However, we chose it for its characteristics as an epistemic and communication-oriented tool, which are closely aligned with the interactive style of conversational agents and with the need for exploring different behaviors and interactive paths by the designers. As shown in [8] MoLIC is a consolidated model, which has been applied successfully to different contexts and domains, but not conversational agents interactions, which bring novel challenges to user-system interaction. Thus, our goal in this paper is to investigate MoLIC's applicability to this context.

4 Methodology

We conducted a study to assess MoLIC's expressiveness and applicability to model conversational agents and intelligent assistants. The study comprised three main steps. First, we selected two conversational agents with at least a moderate level of intelligence: ANA and Bixby, which will be described in section 4.1. Next, we devised scenarios for the evaluation of these agents, as explained in section 4.2. Finally, we conducted reverse engineering sessions where we modeled the scenarios using MoLIC, as discussed in section 4.3.

4.1 Selecting Conversational Agents

For the development of the study's design, we analyzed various software artifacts and selected the following task-oriented conversational systems: ANA and Samsung Bixby (version 3.1.44.26). ANA was chosen for its relevance to the topic of combating COVID19 (section 4.1) [6], while Bixby was chosen due to its widespread adoption, as it is available in Samsung smartphones (section 4.1). Our choice of task-oriented systems was driven by Luger and Sellen's work [19], who state that the purpose of conversational agents "is both support for real time task completion and to develop sufficient knowledge about the user in order to exert agency on their behalf". Moreover, Li et al. [18] differentiate task-oriented chatbots from chatbots for chatting, stating that the former "have been used widely because it can potentially reduce a substantial amount of human labor for customer support if it can resolve relatively simple, but repeated, requests from users."

The chosen conversational systems are described in the following sections.

ANA Chatbot: ANA is a conversational agent that seeks to help combat COVID-19. It was developed by the Telehealth Center of the Federal University of Minas Gerais (UFMG) by a multidisciplinary team of linguists, computer scientists and medical researchers [6].

ANA serves patients seeking to get diagnoses about what they are feeling or seeking relevant information about COVID-19: symptoms, treatments, diagnosis, advice for suspected virus infection, hygiene, pregnancy care, lifestyle, pets, and mask use. It also assists in the triage of suspected cases according to symptom information reported by the patients themselves. At the time of this study, ANA was available to the public on the UFMG Telehealth Center page, in the city of Divinópolis' official government mobile app and on the official government website of Teófilo Otoni, two cities in the state of Minas Gerais [6]. For our study, we utilized the version of the system that was available in April 2021.

Figure 2 shows a decision tree of ANA, wherein the sequence of responses from users who consult it determine the final recommendation. As we can see, it directs users to the following outcomes: "No symptoms", "Orientation" (in milder cases of the disease) or "Monitoring" (in case of risk groups), "Hospital Care" (when the user has persistent fever) or even to "Emergency Care" (in more severe cases of the disease).

Figure 3 shows a snippet of the agent's conversation flow during an interaction with a user in section (a). Users can provide answers by typing out full responses on the keyboard or by clicking on answer suggestions that appear on the screen. Section (b) displays an example of diagnostic provided by ANA while section (c) shows the various options that users have when trying to request information from the system.

Bixby Assistant: Bixby by Samsung [35] is an intelligent assistant that communicates with its users by offering a variety of services that can help them in their daily lives. With it, users can receive service recommendations according to their routine, create reminders of activities to do, send SMS text messages among other features [35].

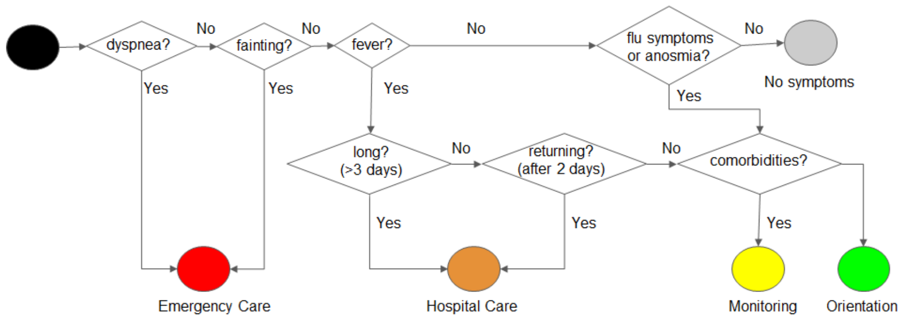


Fig.2. ANA Decision Tree [6].

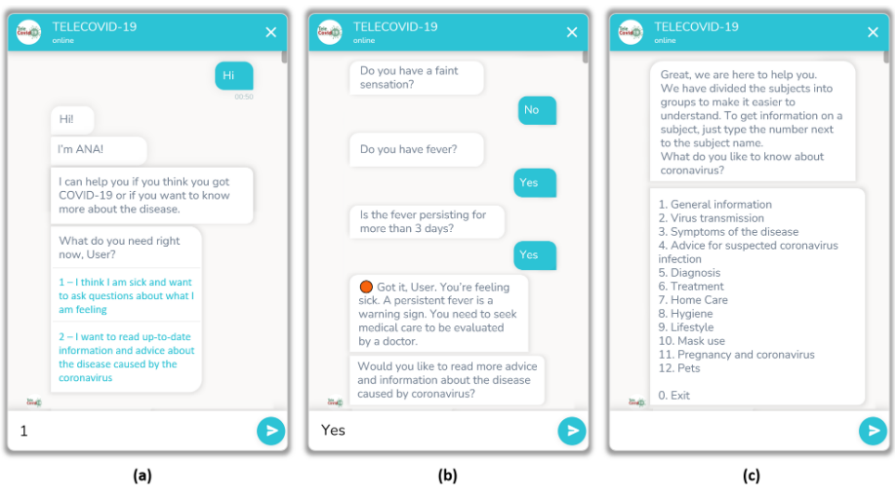


Fig.3. Interaction with ANA: (a) Initial conversation, (b) Diagnostics and (c) Information. (Translated and adapted from Portuguese).

By setting configurations in the assistant, such as voice calibration or giving access permissions to your applications and equipment, users can control their television or the lights in their house, through Bixby. By configuring the assistant's advanced features, they can also allow Bixby to learn from their usage patterns and automate tasks to make their daily lives easier [34]. Currently[1], versions of the assistant are available in English, French, German, Italian, Korean, Mandarin Chinese, Spanish, and Portuguese [34].

4.2 Choosing and Defining Scenarios

In the second step of the study, we defined a couple of scenarios to reverse engineer with MoLIC. Due to the different nature of the selected software, the scenarios were different for each one. In the case of ANA, the scenario included three parts: (i) the initial interaction, where user and agent introduce themselves to each other and the agent asks for personal data, and then presents the option for the user to choose between the two possible paths for interacting with the system; (ii) the interaction for the triage support when users choose the first option "1 - I think I'm sick..."; and (iii) the interaction for the second–option "2 - I want to read updated Information..." (question and answers) covering one of the information topics (Disease Symptoms).

For Bixby, we defined five scenarios to be modeled. In four of them, the goal was to choose different situations that represented everyday user tasks. The first one involved sending e-mail and text messages (SMS and WhatsApp) from the assistant. The second scenario considered a tourist in a foreign country, requesting text translation, hotel recommendations, restaurants, tourist spots, weather forecast, and subway schedules in a given city. In the third scenario, we chose everyday activities, such as using a calculator, using a music player, accessing files from the download folder on the smartphone, bank applications, smartphone settings, and turning on the flashlight. In the fourth scenario, we asked the agent about various web queries such as cooking recipes, income tax information, access to academic research (*i.e.*, the periodic table of elements and works about the French revolution), among others. The last scenario focused on health and wellness, as this was a hot topic due to the pandemic.

For all the scenarios, in both ANA and Bixby, we explored diverse conversational paths within the system, capturing result screens throughout. These were later consolidated in our meetings.

4.3 Using MoLIC to Reverse Engineer Interactions with Conversational Agents

In the third step of this study, we conducted reverse modeling sessions of ANA and Bixby using MoLIC. We modeled all the scenarios defined in the previous section. By

[1] Bixby was inspected in June 2021 for this study.

[2] This image was captured in a previous version of Bixby and we were unable to reproduce

taking this approach we were able to analyze relevant and realistic interaction paths. For each of the conversations analyzed we were able to explore how MoLIC might be used to express the interaction, identifying limitations, and raising modeling requirements. It is important to highlight that MoLIC was designed to represent the interaction from a communicative perspective and it does not specify the internal functionality of the application, but only the behavior as it is presented to and perceived by the users, which reduces the risk of misinterpretations and of making wrong assumptions about the systems' inner workings.

For clarity, focusing on the issues we found most relevant and avoiding repetitions, we omitted from the diagrams options that were quite similar to one another, using ellipses ("...") to indicate those omissions, *i.e.*, places where the conversation continues but did not need to be fully represented for the purposes of this study.

In order to conduct the analysis, the authors were split into two groups and each one generated a version of the model for each conversational agent. All authors had previous knowledge of modeling in general and of using MoLIC in particular. Three had previous experience in research and/or implementation of chatbots. Each group comprised at least one expert in MoLIC and one expert in chatbots. Next, the models generated by each group were consolidated by all authors. During consolidation, we adopted a qualitative approach to analyze the differences and reach a consensus. In fact, we found no significant differences between the models; in general, the few differences were related to alternative possible representations of a given interaction, analogous to stylistic differences. Thus, the discussion focused on which representation would better express the communicative structure being analyzed. Furthermore, during the consolidation the group identified inadequate or missing expressiveness in MoLIC for representing the investigated conversational agents. In the next section we present the consolidated diagrams.

5 Generated Models

In this section we describe the models that resulted from our reverse engineering, explaining how the interaction we observed with each system was modeled with MoLIC. We omitted the models for some scenarios for the sake of space, since the main aspects that appeared in them were already covered in other models. These models will be recalled in section 6, where we discuss the problems, limitations, and challenges that we observed.

Figure 4 depicts a MoLIC diagram for the ANA scenario part (ii) "1 I think I'm sick...", where the conversational agent triages users according to their answers, following the decision tree illustrated in Figure 2. Starting from the upper left, users will pass through different scenes to check their symptoms in order of gravity. After the user tells ANA "I think I'm sick", the next scene is "Check for dyspnea" because it is the most severe symptom, where the system asks the user if they have shortness of breath or difficulties while breathing. By answering this question, the user gives the floor back to the system. Each black box represents an internal system process of interpreting the user's answer and deciding what to do next: no matter the algorithm used here, whether it is just a pattern matching rule or an advanced NLP or AI engine, during system processing users only see an ellipsis, mimicking a person writing at the other side, and after the processing they are informed of the outcome. If the user answers 'yes' to the question, they will be directed to an emergency orientation message (represented by the rectangle <<Emergency care>>, indicating a system monologue), which terminates the interaction. Otherwise, if they answer 'no', they are led to a follow-up scene "Check for fainting", proceeding to the next scenes in a similar interaction pattern: a scene in which the system asks about a symptom, a user utterance as an answer to the question, an internal system processing to interpret the answer, and either a follow-up question or a final orientation depending on the user's answers up to that point.

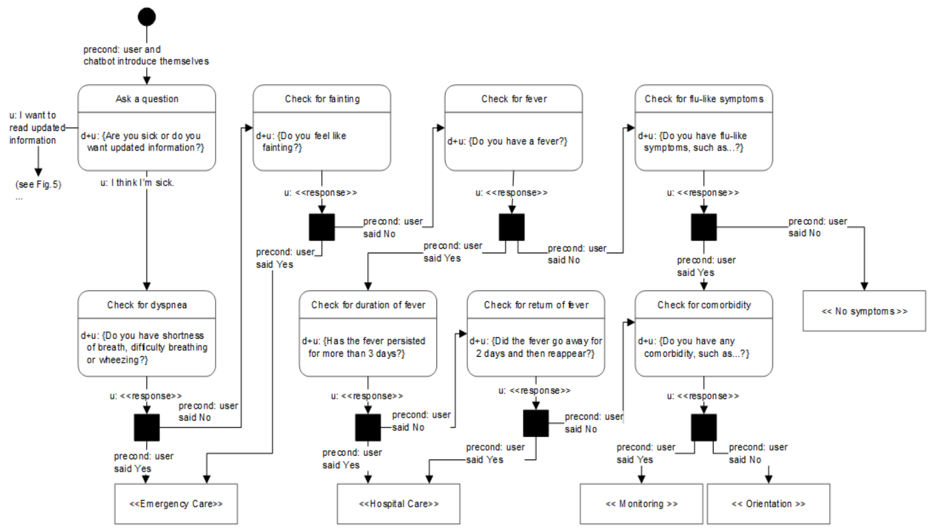


Fig.4. Modeling ANA scenario part (ii) "I think I'm sick..." (triage).

Figure 5 depicts the third part of the scenario with ANA, where users ask for information about the disease ("2 - I want to read updated information..."). The user starts in the upper left corner of the figure in the scene "Choose information topic". In this scene, ANA tells the user all the topics that it can talk about and asks the user to choose. Users tell the chatbot which topic they are interested in by saying "u: response". ANA interprets (*i.e.*, processes) the user's response and opens a different scene according to the user's choice, for instance: if ANA understands they want to receive general information about COVID ("precond: 1", short for "precond: user said 1") it leads the user to the scene "General information". ANA can talk about 12 topics with each being their own scene. Note that the conversation about each topic is not entirely modeled here, as indicated by an ellipsis "..." next to each scene, which includes a conversation path back to the scene "Choose information topic". If ANA understands that the user asked for something that is not one of the choices, it replies that it cannot answer, taking the user back to the "Choose information topic..." scene (alternative path "d: invalid option"). Finally, users can also ask to end the conversation (option “0”), which will take them to the “Provide feedback” scene and then terminate the interaction.

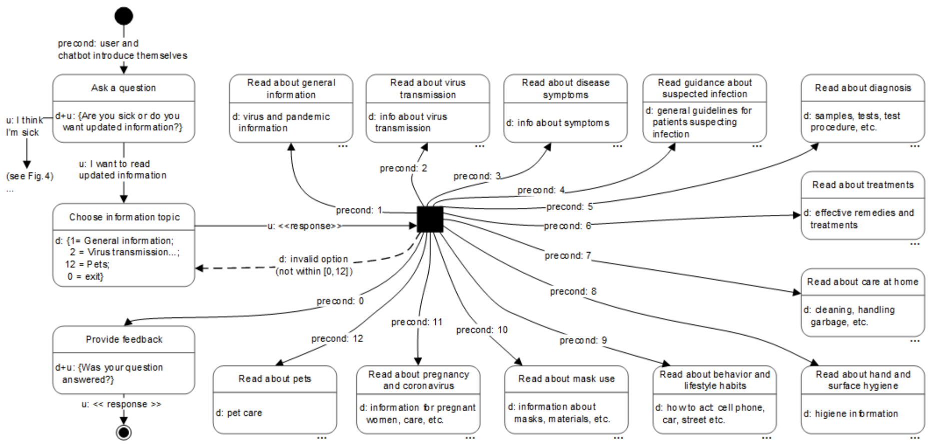


Fig.5. Modeling ANA scenario part (iii): “2 - I want to read updated information on...” (questions and answers). Ellipses indicate purposeful omissions of conversation threads that have a similar structure as the one represented in Figure 6.

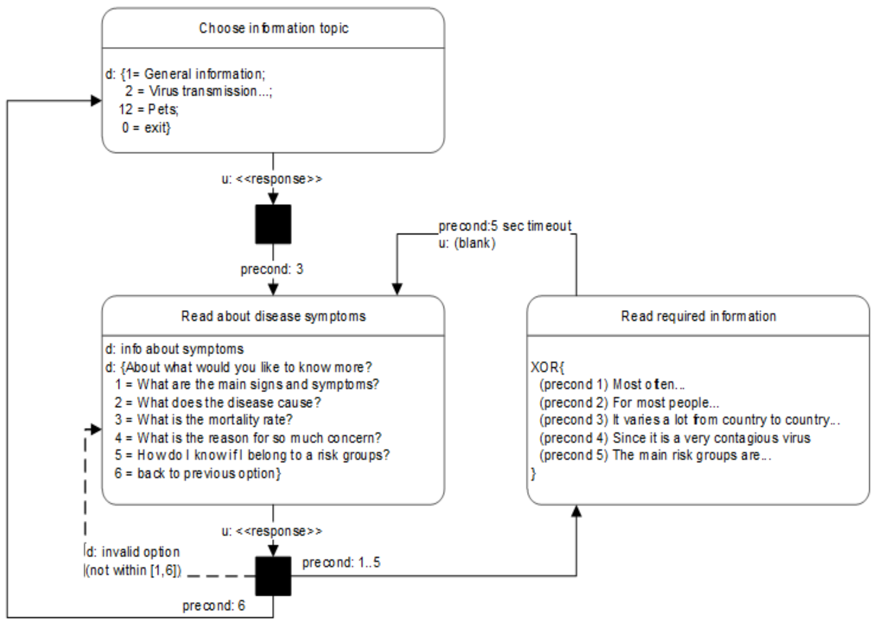


Fig.6. Modeling the details of one of the topics: “Read about disease symptoms”.

The scenes for each topic comprise a conversation that differs in content but are similar in structure. These comprise a dialog about the set of questions that ANA is able to answer related to the respective topic of choice. Figure 6 depicts the follow-up details of one of these topics, namely “Disease Symptoms”. After the user chooses the topic, ANA opens the scene “Read about disease symptoms”, where it tells the user the five questions it can answer plus one option for going back to the previous scene, and asks for the user’s choice. After the user states their choice, ANA interprets this utterance and either takes the user to the “Read responses” scene (“precond: [1..5]”) or to the previous scene (“precond: 6”). In case it does not understand what the user wants, a breakdown happens and ANA goes back to the “Read about disease symptoms” scene (dashed line “d: invalid option”). The “Read requested information” scene is where ANA answers the user’s question appropriately, showing the requested information, and then, after about five seconds, takes the user back to the previous scene to keep the conversation about the topic going. All topic scenes follow this structure, differing only in the number and content of questions and answers, which are specific to each topic.

Figure 7 shows a MoLIC diagram for the Bixby scenario “1 - Sending and Managing Messages”. The main “Start Conversation” scene in the upper left part of the diagram comprises a dialog where the user says “Hi, Bixby!” and wakes up the assistant, which in turn waits and handles the following user requests. After Bixby is activated, an indefinite number of conversations can be started (through the ubiquitous accesses represented in the gray rounded rectangle). The diagram depicts only *some* of the possible interactive paths for scenario 1, comprising calling a contact, sending a SMS, sending a Telegram message, and deleting an email. We use placeholders in some points of the diagram (*e.g.*, “<name>”) to indicate the omissions or variations for that scenario because it is neither practical nor feasible to include all the possible utterances associated with all possible transitions. While reverse engineering the interactions with Bixby, we selected a single utterance for each transition to represent a set of utterances that yield the same effect. This points to an interesting issue of modeling the interaction of conversational agents, which is related to the unconstrained possibilities of natural language inputs users can provide such systems.

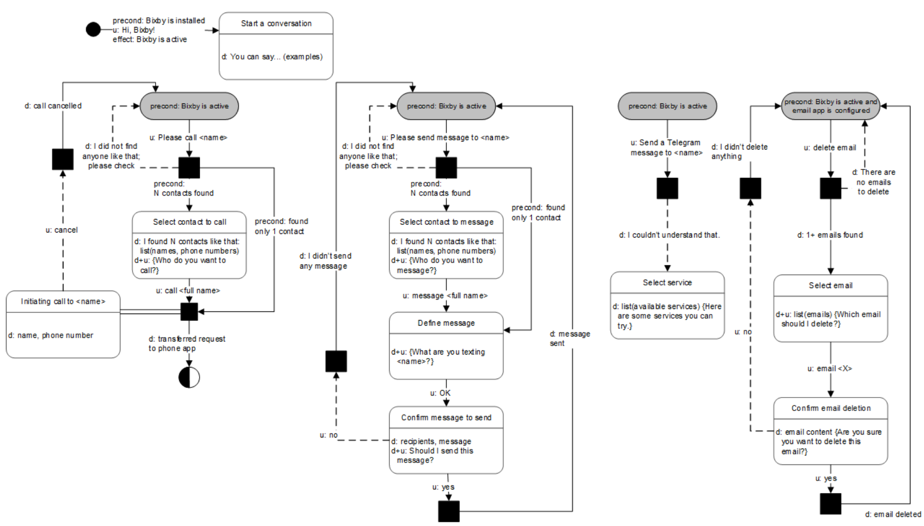


Fig.7. Modeling Bixby Scenario 1 - Sending and Managing Messages.

This and other issues we identified are discussed further ahead, in the next section.

The user asks Bixby to make a call by saying “Please, call <name>”. Bixby will try to interpret the request. In case the assistant understands that the user provided an unknown contact, it says that it did not find the person to call and redirects them back to the scene where they were when uttering the request, represented by the dashed line (breakdown recovery path). If Bixby does not find any contact with the given name, it informs the user of the problem and waits for a new request. If Bixby finds multiple contacts, it presents their names and numbers and asks the user to select which one to call. If it finds a single contact, it will initiate the call, but with a little delay to give the user the opportunity to cancel the call, if necessary. This is represented through the synchronous scene “Initiating call to <name>”, which means that during the processing to which it is coupled, the system informs the user about the processing itself, and may give them an opportunity to interfere (or, in this case, cancel) the processing [7]. It is important to note that Bixby does not make the call itself; instead, it transfers the request to the phone app (an “external system”, semi-filled circle) and, by doing so, terminates its conversation with the user.

The second modeled conversation concerns sending an SMS message. The identification of the message’s recipient is similar to the phone call case mentioned above. However, to fulfill the request, Bixby needs to ask about the content of the message to be sent. Afterwards, it asks for confirmation and, upon receiving it, proceeds to send the message, letting the user know it has done so. It is interesting to note that, although Bixby itself does not send messages, it is able to send the request to the message application, wait for its execution, and interpret its feedback. Moreover, because the operation is almost immediate, the turn-taking between Bixby and the messaging application may go unnoticed by the user. This is why we did not model the external system in this case.

Although the third case (“d: send a Telegram message to <name>”) may be viewed as analogous to sending an SMS message, it is not understood by Bixby. Bixby then presents the available services to the user, in case they misspoke or need to look for alternative services to achieve their goal.

Finally, the user may ask Bixby to delete an email by saying “Delete Email”. Bixby presents a list of emails to the user and asks which one it should delete. The user answers, and Bixby asks for confirmation before deleting the message. In any case, Bixby returns to the “listening, active stage” to wait for further requests.

In modeling the two systems with MoLIC, we found points that are worth discussing and others that need to be explored further in the future. Although we found MoLIC to be applicable to conversational agents, we noticed some limitations or potential challenges of using the language to represent interactions with conversational agents at design time, as discussed in the next section.

6 Findings and Discussion

During our analysis, we identified a few limitations in MoLIC that can be categorized in four topics: 1) Standardized communication snippets, 2) Transfer of responsibility/interlocutor during communication, 3) Breakdowns in modeling, and 4) Conversational agents' Intelligence. For each topic, we present our findings and discuss their implications to the modeling of interactive chatbots.

6.1 Standardized communication snippets

One of the situations we identified in our analysis was when the conversational agents could talk about different contents while employing the same communication structure. An instance of this situation was observed in ANA in the option related to receiving updated information about COVID-19, as described in section 5. If users choose to receive updated information on a topic, ANA presents 12 different topics for users to choose from (Figure 5). For any of the topics a conversation would follow about it employing the same structure shown in Figure 6 for "Disease Symptoms". The difference between them would be on the content of the choices and responses available and depicted in each scene's dialog.

As presented, MoLIC was devised to allow designers to represent user-system communicative paths, in order to reflect on them and on how they convey the designers' metamessage to users. However, we argue that, in cases like this one, in which the communicative structure is the same, but the content of the conversation is different and in which the set of distinct topics is large, representing all of them could generate an overloaded interaction diagram, which could hinder the designers' holistic view of the intended metacommunication.

In the context of our study, a template of the conversation structure could be generated to represent the interactive paths of this communication, independent of the content of each possible conversation. Figure 8 depicts a possible representation of such a template. Furthermore, representing a template could also add to MoLIC's epistemic support to designers, as it would clearly represent that a number of conversations should follow the same structure, allowing designers to reflect upon and define the expected consistency of how the conversations about topics are carried out with users.

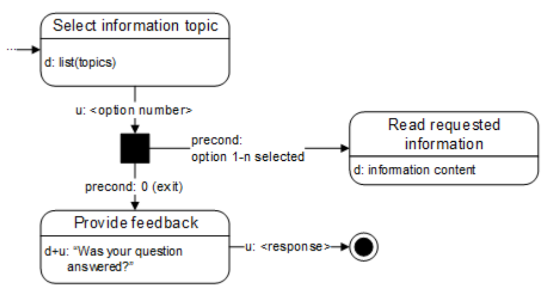


Fig.8. Example of a conversation template.

In proposing an adaptation of MoLIC to include these communication snippets template, it would be necessary to consider that it would mean introducing a new level of abstraction into the modeling language. Having two different levels of abstraction may increase the complexity of generating and understanding the diagrams. Thus, an investigation of the appropriate ways of representing the template and its impact on the use of MoLIC would be necessary.

6.2 Transfer of responsibility/interlocutor during communication

Another situation we identified in our studies was when the conversational agent with whom the user was communicating transferred control over to a third-party who was not initially involved in the conversation. In the Bixby case, we observed that some questions were answered by the assistant itself, while others were redirected to other systems. For instance, Figure 9 shows a text translation case where Bixby uses its own resources to translate some text into another language, previously chosen by the user from the list of languages offered. By contrast, Figure 10 shows examples where Bixby delegated a user's search to Google or used features of other apps, such as the phone's calculator or Google Maps.

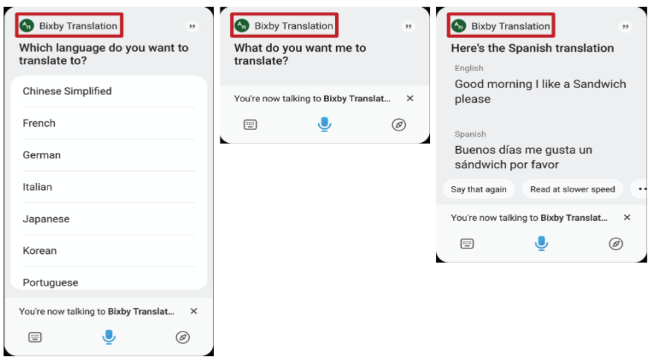


Fig.9: Situations where Bixby uses its own resources.

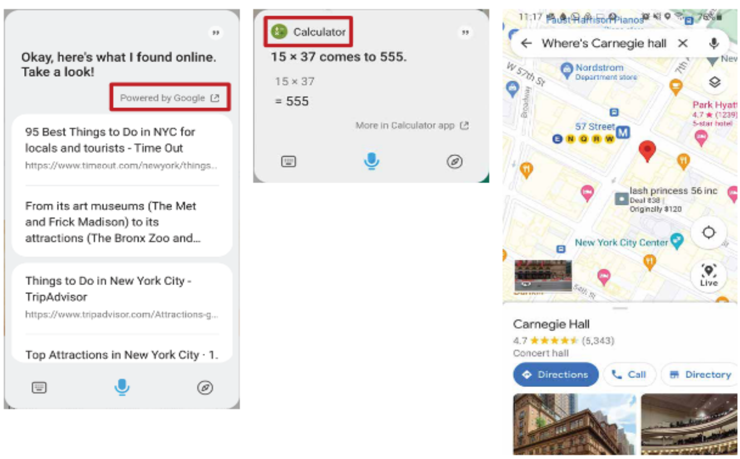


Fig.10. Situations where Bixby uses third-party resources.

In the case of Bixby, we identified two different situations in which a third-party system was brought into the conversation. In the first one, it would bring a result generated by another system and inform the user of its source. In the second one, it ceded control to the other system to present the response to the user (*e.g.*, Google Maps). The first case can easily be represented using MoLIC, as designers can indicate the source of the response as part of the system's utterance after a processing, and in the next scene in which the response is presented, include the source as a sign to be represented in the dialog.

The second situation could be represented using MoLIC's external interlocutor element (as in Figure 7 to represent the phone app). However, MoLIC does not consider how to indicate the interactive paths that could continue from the external system's processing. From the designer's point of view, it would be interesting to represent the transitions that users might be able to make (or not) between these systems. For instance, if the interaction is just a request for the third-party to fulfill (*e.g.*, as in the case of asking Bixby to send an email, in which the email app is not explicitly represented in Figure 7), or whether the user's interaction with the system would be transferred to the third-party system, temporarily or definitely. If designers intended to represent a temporary transfer of the interaction, it might be necessary to model in which situations it would be transferred back, and to which scenes. In case of a transfer intended to end the interaction with the system, it should be clear in the model as well.

A proposal to extend or adapt MoLIC to better represent the possible interactions with third-party systems would have to take into consideration which aspects of these interactions or transitions between interlocutors should be represented and how. It is worth noting that the external system is similar to the system´s processing, in the sense that in MoLIC diagrams it would be represented as a "black box", that is, whatever went on within the system would not be represented, only the utterances that would lead to yielding of the floor to and from the system. Nonetheless, it might be relevant to make the relationship between the systems explicit, distinguishing, for instance, if the third-party system is a service provider (*e.g.*, sends a requested email), if it is a partner (*i.e.*, users might transition from one to the other throughout the interaction), or if it takes responsibility from the system in specific situations.

6.3 Modeling breakdowns

Another interesting finding is the variety of types of breakdowns that may occur in users' interaction with conversational agents. As these systems need to deal mostly with open-ended input, without predefined input widgets, buttons, and menus that constrain the range of interactions, unexpected behaviors are more likely, as they depend on the system's interpretation of the user's expression (in voice or written text) and of the content of their utterance. This has to do with the system's reliability in performing complex and interrelated processing. However, this is currently hidden in MoLIC, as represented by its black box element. We need a way to capture the complexity involved with these processes and respond appropriately to the different kinds of breakdowns that may occur. We have identified six cases that should have clear and distinct representations in an interaction model:

1. The system correctly understands the user's request and returns the answer expected by the user;
2. The system understands the user's request, but identifies that there is relevant information missing and asks the user follow-up questions to complete the information.
3. The system does not completely understand the user's request, but understands parts of it, and engages in a conversation to try and clarify what the user means.
4. The system does not understand the user's request, but is aware of the breakdown and asks the user to repeat or try an alternative message.
5. The system understands the user's request but is unable to satisfy it and informs the user about it.
6. The system misunderstands the user's request, and returns an incorrect answer. Note that this case differs from the others, since it is not possible for the system itself to identify this kind of error. However, designers need to acknowledge this possibility and model mechanisms for the user to recover from this type of error;

The first case was already exemplified by Figures 9 and 10, when Bixby understands the user's request correctly and returns a correct answer (either using its own resources or third-party ones).

Figure 11 illustrates the behavior for the second case, where the system understands the request, does not have enough information to satisfy it, and asks for the missing information. The user asks Bixby to set an alarm; Bixby then asks "What time should it ring?"; the user answers "5AM", and then Bixby successfully sets the alarm and informs the user about the result.

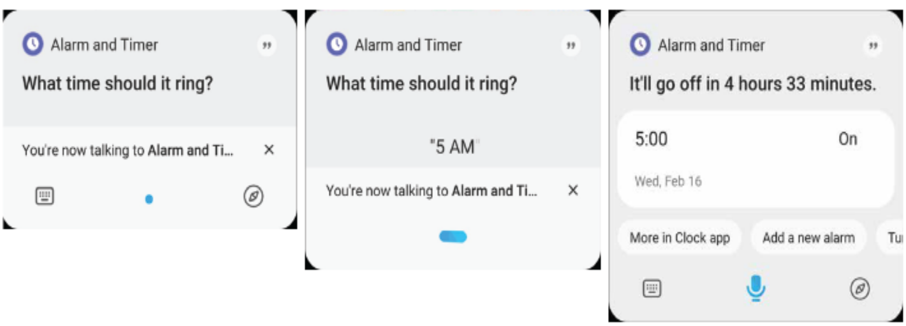
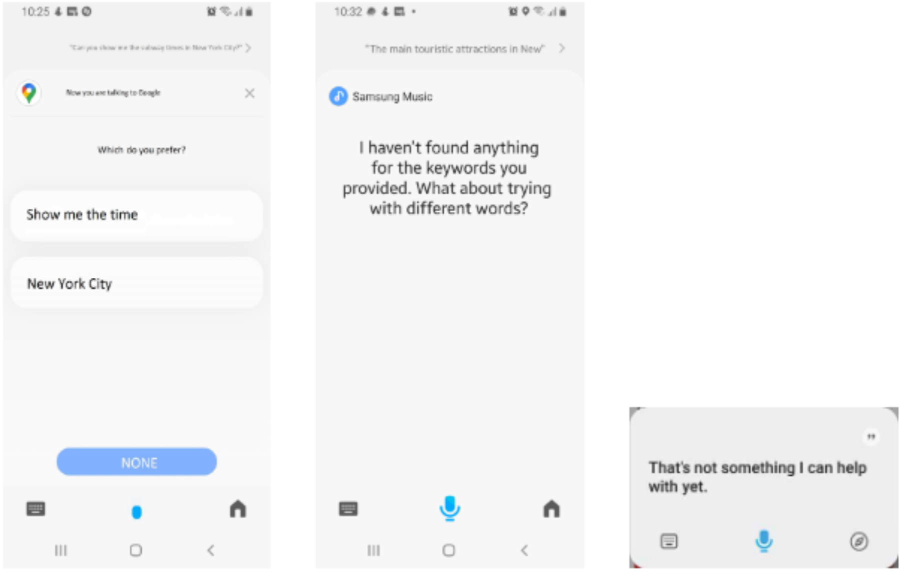


Fig.11. Case 2 – The system cannot satisfy the user request at first and asks for more information ("Set an alarm.").

In the third case, the system does not know how to provide an answer but recognizes some fragments of the request and asks the user to select the fragment in which they are interested. In the example shown in Figure 12a, the user asked, "Can you show me the subway times in New York City?" and Bixby asked the user to decide on whether they are interested in the time or in New York City.



(a) Case 3 – The system understands different fragments of the user's request and asks for the user to select which path to take ("Can you show me the subway times in New York City?").

(b) Case 4 – The system cannot satisfy the user request and asks for the to rephrase it.

(c) Case 5 – The system cannot satisfy the user request when asked "What are the bus schedules in Madrid?".

Fig.12. Cases 3, 4, and 5

In the next two cases, the system realizes the breakdown in the communication and acknowledges that either it does not understand the user or it does understand but cannot satisfy the request yet, possibly due to some missing service. In the fourth case, the system does not understand the user's request, informs them of the problem, and asks them to repeat or even use other terms in the message (Figure 12b).[2] Figure 12c exemplifies the fifth case, in which the system cannot provide an answer. In the example, the user asked, "What are the bus schedules in Madrid?" Bixby then informs them that the task was unsuccessful, stating the message "That's not something I can help with yet."

Finally, Figure 13a shows the behavior of the system for the sixth and last case, when it misunderstands what was asked and provides an incorrect answer. In the example, Bixby was asked again, through voice command, the following question: "What to do in New York City?" However, the assistant displayed the weather in that city. This behavior may cause frustration in the user. It is worth noting that we would typically not model this during design, as this is an incorrect behavior. As we have

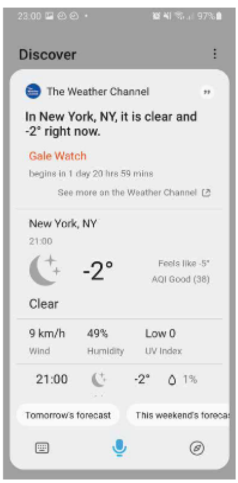
[2] This image was captured in a previous version of Bixby and we were unable to reproduce this case in the latest version.

reverse engineered the actual behavior, however, we illustrate this in Figure 14 with a light gray shading.

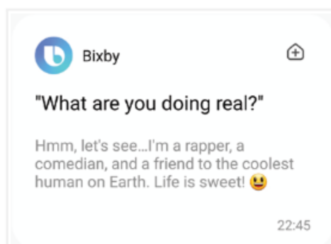
This last case raises an additional question: When it comes to phonetics, how is speech processing performed in this kind of technology? In Figure 13b we have an example where this problem occurs. Through the voice command, the user asked "What to do around here?", which the system misinterpreted as "What are you doing real?"

Regarding the representation of these breakdowns in MoLIC, the challenge is not so much in how to express them in MoLIC, but rather how to do so more efficiently. Case 1 (system understands and responds appropriately), Case 4 (system does not understand and asks user to repeat), and Case 5 (system understands, but identifies that it is unable to comply) would be communicative paths that are usual for traditional interactive systems. In cases 2 and 3, the system engages in further conversation with the user to either complement the required information, or try and understand the request. These two cases could be represented in MoLIC by structuring alternative paths that could be taken to turn a "partially" understood request into a fully understood one. Figure 14 illustrates the six cases in MoLIC diagram fragments, assuming that the conversation with Bixby was already initiated.

The challenge would be that, for each possible user utterance, all these cases would need to be represented. Considering the open-ended aspect of the interaction with conversational agents, modeling partially understood messages (*i.e.*, case 3) could require a different number of interactive paths to be considered. Thus, representing all the possibilities for all the utterances could lead to an overloaded interaction diagram that would be difficult to generate and might hinder its purpose as an epistemic tool.



(a) The system misunderstood the user's question "What to do in New York City?" and provided the weather forecast instead.



(b) A misunderstanding caused by faulty speech processing.

Fig.13. Some system misunderstandings.

An investigation on how to deal with this issue in MoLIC could consider whether the templates proposed in Figure 8 (for the standardized communication snippets) could be a solution in this scenario. This might depend on whether these alternative paths would be similar in different intended communicative acts. Another proposal that might be worth considering would be to create the possibility to represent more than one abstraction level, representing in one level the intended communication, the expected interaction, and complete breakdowns (*i.e.*, cases 1, 4 and 5) as well as, for each intended communicative act, detail (at another level) all the alternative interactive paths that would be anticipated.

Finally, case 6 breakdowns would not be modeled as interactive paths at design time, as it means that the system has failed to perceive the breakdown. Nonetheless, designers should be able to model how users could recover from this situation. Furthermore, these breakdowns are often associated with utterances that sound or are similar to other messages the system can understand. For these cases, designers or natural language processing algorithms might identify utterances that could be easily misinterpreted. With this, these situations would fall into cases 3 (partially understands) or 4 (does not understand, or perceives that it is not clear enough and asks users to repeat).

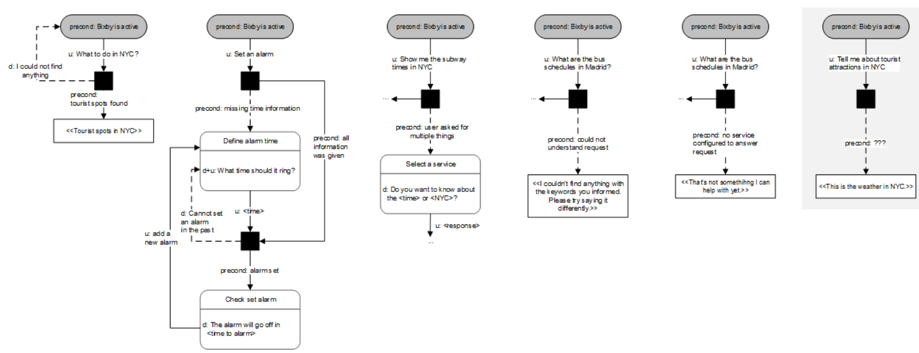


Fig.14. Modeling the six breakdown cases for Bixby (Scenario 2 - Tourist in Foreign Country) using MoLIC.

6.4 Conversational agents' intelligence

The results discussed so far correspond to the use of MoLIC to model the conversational agent ANA, where the flow of the conversation was well defined according to the agent's goals, and results from the Bixby assistant, which exhibits a more dynamic behavior. If the technology is more static, *i.e.*, if its communicative paths have been all previously defined and it has a limited set of questions and answers that it can understand, as is the case with ANA, MoLIC can be used (taking into consideration the limitations described in the previous scenarios). However, when the technology has more intelligence and autonomy, as described by Meyer von Wolff et al. [47], and, as we can see in Bixby or in cases where the agents are able to learn from the users' answers, increasing their knowledge base, the modeling becomes increasingly complex. What aspects of a system's intelligence and autonomy would then be possible and worth modeling?

As we have seen, it is important to note that the differences between conversational agents can be an impacting factor in using MoLIC to model this technology. The role of the designer during the design phase can be modified if the system has a high degree of autonomy. Given these points, we observe that MoLIC is able to model conversational agents with up to a certain degree of autonomy, and with pre-established conversations, but it needs more adaptations if we want to adopt it for modeling more complex technologies.

The different interactive paths that can arise during interaction with conversational agents confirm the need to adapt MoLIC to model virtual assistants. For instance, how could we represent the multiple interactive paths that can arise from system processes? Would it be feasible to create a MoLIC diagram for more dynamic (*i.e.*, paths defined in execution time) conversations? If we do not know the possible paths, this is a problem that goes beyond MoLIC. In these cases, it would be relevant to take a step back and discuss the role of the designer and how it changes when instead of designing all possible interactive paths (as in traditional GUI-based systems), designers accept that the system will autonomously define them in use time.

It is worth pointing out that, in considering conversational systems, they can vary from the most traditional, where all communicative paths are well-defined, to the highly dynamic ones. In our case studies, we analyzed systems in different points of this range. In ANA, because the system aims to guide the user to achieve specific goals, they were previously defined, with all possible communicative paths designed. In this case, MoLIC, in spite of the issues described, could support designers in defining (*i.e.*, ensuring that the agent will ask the necessary questions to classify the user's specific case) and reflecting on the possible communicative paths, achieving its goal as an epistemic tool.

Our analysis of Bixby indicated its more dynamic behavior, with its capacity to learn, offering open conversations. For these more dynamic systems, would it be useful to use MoLIC to model anticipated communicative paths? It would be worth investigating what would be relevant, in terms of interaction, to model for more open and dynamic paths. What should be represented in an epistemic tool to support designers in reflecting upon them? Could MoLIC be adapted or extended to include these representations?

Intelligent systems bring new issues that would need to be considered in adapting or extending MoLIC (and probably any other model): the system's autonomy (*i.e.*, the system being responsible for initiating the conversation [47]) and its transparency and explicability (*i.e.*, the need for users to understand the system's rationale and decisions). Systems' autonomous behaviors change the stimulus-response paradigm to one of partnership [10], thus, the model would need to be able to represent who initiates an interactive path, about what topics, and to which interactive paths it leads. Regarding transparency and explicability, it may be useful to indicate which of the system's processes relating to inferences and decision making users should be able to understand, and which signs would help them do so. MoLIC proposes that system processes be represented by "black boxes": users are only aware of what is communicated to them about the process. However, MoLIC was conceived to represent “traditional” systems, in which the system processes are deterministic and basically directed by the user’s utterances, so the possible outcomes of the system processes can be fully determined at design time. In contrast, in AI-infused systems, a user’s request is only one of the inputs to the system processing, as those systems are built to learn and reason about knowledge acquired not only from that user, but also from other users and potentially huge data sets used for training an embedded machine learning model. In such systems, the interaction should be designed to allow users to “see through” (or be told about) some of what is taking place within the system processing, so they can understand the rationale behind the system’s inferences, decisions, and behaviors.

7 Limitations

The main study limitations concern the choice of agents, scenarios, and modeling language. First, we only analyzed two conversational agents that are mainly task-oriented, which cannot be considered fully representative of the whole class of conversational agents, task-oriented or otherwise. However, to broaden our perspective on the design issues, we analyzed two very different systems, ANA (a rule-based textual chatbot) and Bixby (a full-fledged voice-based intelligent virtual assistant). Second, regarding the scenarios, as ANA was a simple chatbot, we modeled it in its entirety, double-checking it with the development team. For Bixby, our main criterion for choosing the scenarios was to explore a wide diversity of everyday tasks and behaviors, from simple information finding (which is a typical purpose of using chatbots) to interacting with other applications (such as e-mail, alarm, weather, and maps applications, for instance). Third, as we have used only MoLIC, the language may have constrained the range and kinds of issues analyzed. Our choices allowed us to analyze MoLIC’s expressiveness to model the interaction design with a wide range of design choices in this kind of technology.

As stated in Section 4.3, to increase the robustness of our analysis, four researchers built the MoLIC diagrams for the selected scenarios, and they converged in their conclusions, as the differences in the models they produced were stylistic in nature.

However, we recognize that this analysis is not exhaustive, *i.e.*, the set of issues identified cannot be deemed complete. Further studies, especially with open-ended conversational agents (*e.g.*, for interviewing or counseling), are necessary to broaden the coverage of the analysis and allow for generalizations. Even if they do not exactly replicate the interactions present in this study, they may encounter similar issues related to modeling interactions with conversational agents.

8 Conclusions

In this study, we aimed to investigate whether the user-system interaction with conversation agents could be modeled using a design-phase dialog model, supporting interaction design decisions decoupled from development decisions regarding natural-lower-level language processing and utterance disambiguation. To reflect on interaction design decisions, we chose MoLIC, a dialog model grounded in Semiotic Engineering. MoLIC can be used to create models at design time or at evaluation time, when analyzing a given system and its behaviors, either by inspecting an existing MoLIC diagram or by reverse modeling one from the system behavior. We used it to model two existing task-oriented conversational agents, ANA and Samsung's assistant, Bixby, in a selected set of scenarios.

We were able to analyze several aspects of the interaction with conversational agents using such a design model. The study illustrated the usefulness of keeping design-level decisions separate from development-level decisions. Our study also revealed some modeling requirements that MoLIC satisfies and others that it does not.

Our analysis revealed four different types of design decisions in which we identified some limitations of MoLIC. Although in the first three situations – standardized communication snippets, transfer or interlocutor during communication, and modeling breakdowns –, MoLIC could be used to represent the interaction, it generated an overloaded model that could hinder the designer's holistic view of the possible user-system conversations, and could end up limiting its use an epistemic tool. Thus, considerations on how to adapt or extend MoLIC to allow for a more efficient representation of these situations would be worth investigating. Any new proposal should take into account the cognitive cost for designers of adding new elements to the language, especially if the concept of different abstraction levels is introduced, as well as its impact on how MoLIC is used.

MoLIC emphasizes the need to model breakdown recovery paths, *i.e.*, what the user is able or needs to do when something goes wrong. However, in cases where the system misunderstands what the user asked for and provides an incorrect answer, there is no way to predict that breakdown at design time, as the system is not aware of its own misinterpretation. The very processing of the system, represented by the black boxes, can lead to an unpredictable scene or outcome. How could MoLIC be used to represent this kind of case, in which the system processing can have unexpected outcomes? Would it suffice to create a new element to represent a non-deterministic processing?

Finally, situation four uncovered a more complex limitation regarding *intelligent* conversational agents. MoLIC's applicability to model conversational agents may depend on the level of autonomy and proactivity of the system being modeled. If all possibilities of interaction are foreseen, MoLIC is able to model them. If the possibilities of interaction and communication expand at interaction time, MoLIC would require adaptations or extensions to express which new types of interactive paths, the system itself could create and in which situations. However, this would also require a theoretical reflection of the impact in Semiotic Engineering's perception of the metamessage being sent by designers and what epistemic aspects would be worth fostering in these situations.

The results presented and discussed are relevant, as they increase the knowledge of the HCI community about modeling interaction in general, and using MoLIC in particular, for systems that have some degree of artificial intelligence. They also point the HCI community towards new initiatives, such as potential adaptations or extensions to MoLIC or the proposal of novel modeling languages, alongside new issues to be considered in these efforts.

Even though we focus on design-level decisions, we acknowledge that the type and range of modeled interactions must be realized in the implementation. Therefore, the model can drive the selection of the underlying technology. In case the chatbot development technology cannot produce the information necessary for the interaction to occur as modeled, the design model might even be used to drive technological advances. For instance, if the designer decides to model a conversation thread aiming to correct inappropriate utterances, the underlying technology needs to be able to detect such utterances to be able to realize the designer's vision, as discussed by Rangel et al. in [32].

This work points to a broad range of future work. First, we intend to propose refinements to MoLIC, to meet the demands raised in this study. Then, we would model additional chatbots and conversational agents, going beyond task-oriented agents, investigating the expressiveness of the refined MoLIC for this kind of technology, which is increasingly present in our daily lives. Finally, we would investigate how a design-level model such as MoLIC can aid in guiding development decisions using various conversational agent platforms.

Acknowledgement. The authors thank CAPES and Simone Barbosa thanks CNPq for partially funding her work (process #313049/2021-1).

References

[1] Adamopoulou, E., Moussiades, L.: An Overview of Chatbot Technology. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations, IFIP Advances in Information and Communication Technology, vol. 584, pp. 373–383, Springer International Publishing, Cham (2020), ISBN 978-3-030-49186-4, https://doi.org/10.1007/978-3-030-49186-4_31

[2] Brandtzaeg, P.B., Følstad, A.: Chatbots: changing user needs and motivations. Interactions **25**(5), 38–43 (Aug 2018), ISSN 1072-5520, https://doi.org/10.1145/3236669,

[3] Cambre, J., Kulkarni, C.: Methods and Tools for Prototyping Voice Interfaces. In: Proceedings of the 2nd Conference on Conversational User Interfaces, pp. 1–4, CUI ’20, Association for Computing Machinery, New York, NY, USA (Jul 2020), ISBN 978-1-4503-75443, https://doi.org/10.1145/3405755.3406148,

[4] Carlmeyer, B., Chromik, M., Wrede, B.: Interaction Model for Incremental Information Presentation. In: Proceedings of the 5th International Conference on Human Agent Interaction, pp. 335–339, HAI ’17, Association for Computing Machinery, New York, NY, USA (Oct 2017), ISBN 978-1-4503-5113-3, https://doi.org/10.1145/3125739.3132582,

[5] Castle-Green, T., Reeves, S., Fischer, J.E., Koleva, B.: Decision Trees as Sociotechnical Objects in Chatbot Design. In: Proceedings of the 2nd Conference on Conversational User Interfaces, pp. 1–3, CUI ’20, Association for Computing Machinery, New York, NY, USA (Jul 2020), ISBN 978-1-4503-7544-3, https://doi.org/10.1145/3405755.3406133,

[6] Chagas, B.A., Ferreguetti, K., Ferreira, T.C., Prates, R.O., Ribeiro, L.B., Pagano, A.S., Reis, Z.S.N., Jr, W.M., Ribeiro, A.L., Marcolino, M.S.: Chatbot as a Telehealth Intervention Strategy in the COVID-19 Pandemic: Lessons Learned from an Action Research Approach. CLEI Electronic Journal **24**(3), 6:1–6:17 (Dec 2021), ISSN 07175000, https://doi.org/10.19153/cleiej.24.3.6, URL http://www.clei.org/cleiej/index.php/cleiej/article/view/515

[7] Da Silva, B.S., Barbosa, S.D.J.: Designing human-computer interaction with MoLIC diagrams–a practical guide. Tech. Rep. 12/07, PUC-Rio, Rio de Janeiro (2007), ISSN: 0103-9741. Available online at: http://bib-di.inf.puc-rio.br/ftp/pub/docs/techreports/07_12_silva.pdf

[8] de Carvalho, A.P., Pereira, F.H.S., Assunção, O.B., Pereira, A.F., Prates, R.O.: An analysis of MoLIC's consolidation. In: Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems, pp. 1–15, IHC '19, Association for Computing Machinery, New York, NY, USA (Oct 2019), ISBN 978-1-4503-6971-8, https://doi.org/10.1145/3357155.3358461,

[9] De Souza, C.S.: The semiotic engineering of human-computer interaction. MIT press, New York, NY, USA (2005) https://doi.org/10.7551/mitpress/6175.001.0001

[10] Farooq, U., Grudin, J.: Human-computer integration. Interactions **23**(6), 26–32 (oct 2016), ISSN 1072-5520, https://doi.org/10.1145/3001896,

[11] Ferreira, J.J., Fucs, A., Segura, V. (2019). Modeling People-AI Interaction: A Case Discussion with Using an Interaction Design Language. In: Marcus, A., Wang, W. (eds) Design, User Experience, and Usability. User Experience in Advanced Technological Environments. HCII 2019. Lecture Notes in Computer Science(), vol 11584. Springer, Cham. https://doi.org/10.1007/978-3-030-23541-3_27

[12] Fernandes, U.d.S., Prates, R.O., Chagas, B.A., Barbosa, G.A.R.: 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, Association for Computing Machinery, New York, NY, USA (2021), ISBN 9781450386173, https://doi.org/10.1145/3472301.3484367,

[13] Følstad, A., Araujo, T., Law, E.L.C., Brandtzaeg, P.B., Papadopoulos, S., Reis, L., Baez, M., Laban, G., McAllister, P., Ischen, C., Wald, R., Catania, F., Meyer von Wolff, R., Hobert, S., Luger, E.: Future directions for chatbot research: an interdisciplinary research agenda. Computing 103(12), 2915–2942 (Dec 2021), ISSN 1436-5057, https://doi.org/10.1007/s00607-021-01016-7,

[14] Følstad, A., Brandtzæg, P.B.: Chatbots and the new world of HCI. Interactions **24**(4), 38–42 (Jun 2017), ISSN 1072-5520, https://doi.org/10.1145/3085558,

[15] Go, E., Sundar, S.S.: Humanizing chatbots: The effects of visual, identity and conversational cues on humanness perceptions. Computers in Human Behavior **97**, 304–316 (Aug 2019), ISSN 0747-5632, https://doi.org/10.1016/j.chb.2019.01.020, URL https://www. sciencedirect.com/science/article/pii/S0747563219300329

[16] Guzzoni, D., Baur, C., Cheyer, A.: Modeling Human-Agent Interaction with Active Ontologies. In: The Association for the Advancement of Artificial Intelligence Spring Symposium: Interaction Challenges for Intelligent Assistants, pp. 52–59, AAAI, Vancouver, British Columbia, Canada (2007). Available online at: http://www.aaai.org/Library/Symposia/Spring/2007/ss07-04-009.php

[17] Laranjo, L., Dunn, A.G., Tong, H.L., Kocaballi, A.B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A.Y.S., Coiera, E.: Conversational agents in healthcare: a systematic review. Journal of the American Medical Informatics Association 25(9), 1248–1258 (Sep 2018), ISSN 1527974X, https://doi.org/10.1093/jamia/ocy072,

[18] Li, C.H., Chen, K., Chang, Y.J.: When there is no progress with a task-oriented chatbot: A conversation analysis. In: Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI ’19, Association for Computing Machinery, New York, NY, USA (2019), ISBN 9781450368254, https://doi.org/10.1145/3338286.3344407,

[19] Luger, E., Sellen, A.: ”like having a really bad pa”: The gulf between user expectation and experience of conversational agents. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, p. 5286–5297, CHI ’16, Association for Computing Machinery, New York, NY, USA (2016), ISBN 9781450333627, https://doi.org/10.1145/2858036.2858288,

[20] Marques, A.B., Barbosa, S.D.J., Conte, T.: A comparative evaluation of interaction models for the design of interactive systems. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp. 173–180, SAC ’16, Association for Computing Machinery, New York, NY, USA (Apr 2016), ISBN 978-1-4503-3739-7, https://doi.org/10.1145/2851613.2851679,

[21] Montenegro, J.L.Z., da Costa, C.A., da Rosa Righi, R.: Survey of conversational agents in health. Expert Systems with Applications 129, 56–67 (Sep 2019), ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2019.03.054,
URL https://www.sciencedirect.com/science/article/pii/S0957417419302283

[22] Moore, R.J.: A Natural Conversation Framework for Conversational UX Design. In: Moore, R.J., Szymanski, M.H., Arar, R., Ren, G.J. (eds.) Studies in Conversational UX Design, pp. 181–204, Human–Computer Interaction Series, Springer International Publishing, Cham (2018), ISBN 978-3-31995579-7, https://doi.org/10.1007/978-3-319-95579-7 9,

[23] Moore, R.J., Arar, R.: Conversational UX Design: An Introduction. In: Moore, R.J., Szymanski, M.H., Arar, R., Ren, G.J. (eds.) Studies in Conversational UX Design, pp. 1–16, Human–Computer Interaction Series, Springer International Publishing, Cham (2018), ISBN 978-3-319-95579-7, https://doi.org/10.1007/978-3-319-95579-7 1,

[24] Moore, R.J., Arar, R.: Conversational UX Design: A Practitioner’s Guide to the Natural Conversation Framework. Association for Computing Machinery, New York, NY, USA (2019), ISBN 978-1-4503-6301-3 https://doi.org/10.1145/3304087.3304096

[25] Moore, R.J., Szymanski, M.H., Arar, R., Ren, G.J. (eds.): Studies in Conversational UX Design. Human–Computer Interaction Series, Springer International Publishing, Cham (2018), ISBN 978-3-319-95579-7, URL https://doi.org/10.1007/978-3-319-95579-7

[26] Oliveira, N., Costa, A., Araujo, D., Portela, C.: HelpCare: Um Protótipo de ChatBot para o Auxílio do Tratamento de Doenças Crônicas. In: Anais do Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS), pp. 282– 287, SBC, Niterói (Jun 2019), https://doi.org/10.5753/sbcas.2019.6263,
URL https://sol.sbc.org.br/index.php/sbcas/article/view/6263, iSSN: 2763-8952

[27] Peirce, C.S.: The Essential Peirce: selected philosophical writings, vol. 2. Indiana University Press, Bloomington (1992). ISBN: 978-0-253-21190-3.

[28] Piccolo, L.S.G., Mensio, M., Alani, H.: Chasing the Chatbots. In: Bodrunova, S.S., Koltsova, O., Følstad, A., Halpin, H., Kolozaridi, P., Yuldashev, L., Smoliarova, A., Niedermayer, H. (eds.) Internet Science, pp. 157– 169, Lecture Notes in Computer Science, Springer International Publishing, Cham (2019), ISBN 978-3-030-17705-8, https://doi.org/10.1007/978-3-03017705-8 14

[29] Pérez-Soler, S., Guerra, E., de Lara, J.: Flexible Modelling using Conversational Agents. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 478–482, IEEE, Munich, Germany (Sep 2019), https://doi.org/10.1109/MODELS-C.2019.00076

[30] Pérez-Soler, S., Guerra, E., de Lara, J.: Creating and Migrating Chatbots with CONGA. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), pp. 37–40, `IEEE, Madrid, Spain (May 2021), https://doi.org/10.1109/ICSECompanion52605.2021.00030, iSSN: 2574-1926

[31] Rapp, A., Curti, L., Boldi, A.: The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots. International Journal of Human-Computer Studies 151, 102630 (Jul 2021), ISSN 1071-5819, https://doi.org/10.1016/j.ijhcs.2021.102630,
URL https://www.sciencedirect.com/science/article/pii/ S1071581921000483

[32] Rangel, M.C., de Castro Salgado, L.C., Bim, S.A.: Tradubem: Towards a text verification tool for an altruistic language. In: Proceedings of the 21st Brazilian Symposium on Human Factors in Computing Systems, IHC '22, Association for Computing Machinery, New York, NY, USA (2022), ISBN 9781450395069, https://doi.org/10.1145/3554364.3559143,

[33] Sabharwal, N., Agrawal, A. (2020). Introduction to Google Dialogflow. In: Cognitive Virtual Assistants Using Google Dialogflow. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-5741-8_2

[34] Samsung Electronics: Samsung Bixby: Your Personal Voice Assistant (Apr 2017), URL https://www.samsung.com/us/explore/bixby/

[35] Samsung Electronics: Getting to know bixby (Feb 2022), URL https:// www.samsung.com/uk/smartphones/galaxy-s9-need-to-knows/bixby/

[36] Schön, D.A.: The reflective practitioner: How professionals think in action. Basic books, USA (1983). ISBN: 978-0465068784.

[37] Shevat, A.: Designing Bots. O'Reilly Media, Inc., [s.l.] (May 2017), ISBN 978-1-4919-7482-7, URL https://www.oreilly.com/library/view/ designing-bots/9781491974810/

[38] da Silva, B.S., Barbosa, S.D.J.: Modelando a Interação do Nita: um estudo de caso e extensões ao MoLIC. In: Anais do VI Simpósio sobre Fatores Humanos em Sistemas Computacionais, pp. 181–184, Sociedade Brasileira de Computação - SBC, Curitiba, PR (2004), URL http://www2.sbc.org. br/ce-ihc/Anais2004/AnaisIHC2004-completos.pdf

[39] de Souza, L.G., Barbosa, S.D.J.: Extending molic for collaborative systems design. In: Kurosu, M. (ed.) Human-Computer Interaction: Design and Evaluation, pp. 271–282, Springer International Publishing, Cham. (2015), ISBN 978-3-319-20901-2 https://doi.org/10.1007/978-3-319-20901-2_25

[40] de Souza, L.G., Barbosa, S.D.J., Fuks, H.: Evaluating the expressiveness of molicc to model the hci of collaborative systems. In: International Conference of Design, User Experience, and Usability, LNCS '16, vol. 9746, pp. 255–265, Springer International Publishing, Toronto, Canada— (Jun 2016) https://doi.org/10.1007/978-3-319-40409-7_25

[41] Svenningsson, N., Faraon, M.: Artificial Intelligence in Conversational Agents: A Study of Factors Related to Perceived Humanness in Chatbots. In: Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference, pp. 151–161, AICCC 2019, Association for Computing Machinery, New York, NY, USA (Dec 2019), ISBN 978-1-4503-7263-3, https://doi.org/10.1145/3375959.3375973,

[42] Svikhnushina, E., Pu, P.: Key Qualities of Conversational Chatbots — the PEACE Model. In: 26th International Conference on Intelligent User Interfaces, pp. 520–530, Association for Computing Machinery, New York, NY, USA (Apr 2021), ISBN 978-1-4503-8017-1 https://doi.org/10.1145/3397481.3450643

[43] Teixeira, B.R., Barbosa, S.D.J.: Investigating the integration of user values with design rationale and its effects on HCI design artifacts. In: Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems, pp. 1–10, IHC '20, Association

for Computing Machinery, New York, NY, USA (Oct 2020), ISBN 978-1-4503-81727, https://doi.org/10.1145/3424953.3426488,

[44] Valtolina, S., Neri, L.: Visual design of dialogue flows for conversational interfaces. Behaviour & Information Technology 40(10), 1008–1023 (Jul 2021), ISSN 0144-929X, https://doi.org/10.1080/0144929X.2021.1918249,

[45] Valério, F.A.M., Guimarães, T.G., Prates, R.O., Candello, H.: Here's What I Can Do: Chatbots' Strategies to Convey Their Features to Users. In: Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems, pp. 1–10, IHC 2017, Association for Computing Machinery, New York, NY, USA (Oct 2017), ISBN 978-1-4503-63778, https://doi.org/10.1145/3160504.3160544,

[46] Wachtel, A., Fuchß, D., Schulz, S., Tichy, W.F.: Approaching Natural Conversation Chatbots by Interactive Dialogue Modelling amp; Microsoft LUIS. In: 2019 IEEE International Conference on Conversational Data Knowledge Engineering (CDKE), pp. 39–42, IEEE, San Diego, CA, USA (Dec 2019), https://doi.org/10.1109/CDKE46621.2019.00013

[47] Meyer von Wolff, R., Hobert, S., Schumann, M.: How May I Help You? – State of the Art and Open Research Questions for Chatbots at the Digital Workplace. In: Proceedings of the 52nd Hawaii international conference on system sciences, pp. 95–104, HICSS, Hawaii (Jan 2019), ISBN 978-0-9981331-2-6, https://doi.org/10.24251/HICSS.2019.013, URL http://scholarspace.manoa.hawaii.edu/handle/10125/59450