Modeling People-AI Interaction: A Case Discussion with Using an Interaction Design Language

Juliana Jansen Ferreira(✉), Ana Fucs, and Vinícius Segura

IBM Research, Av. Pasteur 146, 22290-240 Rio de Janeiro, Brazil
{jjansen, anafucs, vboas}@br.ibm.com

Abstract. Artificial Intelligent (AI) system development is the current challenge for all areas related to software development practice and research, including Human-Computer Interaction (HCI). Most AI systems' research has been focused on the performance and accuracy of Machine Learning (ML) algorithms. Recently, new research questions concerning people in the loop of AI systems development and behavior have been emerging such as bias, reasoning, and explainability. In this new people and AI systems scenario, humans and computers collaborate, using their unique and powerful capabilities in a kind of symbiosis. In this new setting, AI systems are now real social actors as they are active players in the interaction with people. Defining and understanding the behavior of an AI system and its motivation for suggestions and reasoning are definitely a complex endeavor. HCI and Software Engineering communities, with their designers and developers, use models to represent, discuss and explore different domain scenarios in different stages of the software development process. In this paper, we present and discuss a scenario represented in an interaction modeling representation and how it can enable the representation and discussion of the people-AI symbiosis.

Keywords: Artificial Intelligence · Interaction model · User models · Software models

1 Introduction

The development of Artificial Intelligence (AI) systems is a current challenging scenario for all people involved in the software development process. Most AI systems' research has been focused on the performance and accuracy of Machine Learning (ML) algorithms [1, 15]. Recently, new research questions concerning people in the loop of AI systems development and behavior have been emerging such as bias, reasoning, and explainability [3, 4, 12, 13]. Defining and understanding the behavior of an AI system and its motivation for suggestions and reasoning are definitely a complex endeavor.

In this new people and AI systems scenario, humans and computers collaborate, using their unique and powerful capabilities in a kind of symbiosis. On the one hand, computers bring their capability to deal (organize, relate, present, etc.), in different ways, with large sets of data, which is not possible for the human brain. On the other

© Springer Nature Switzerland AG 2019
A. Marcus and W. Wang (Eds.): HCII 2019, LNCS 11584, pp. 379–388, 2019.
https://doi.org/10.1007/978-3-030-23541-3_27

hand, humans provide their capability to judge, make decisions, to assess situations considering their intuition and knowledge (structured, unstructured, tacit, etc.), and their ability to innovate in a given domain. The human cognitive abilities and computer processing prowess can now be coupled very tightly, and the resulting partnership will present new ways for the human brain to think and computers to process data [9].

AI Systems are now real social actors as they are active players in the interaction with people. They can learn and change their behavior while they interact with users. This new context presents new experiences and challenges not only for users but also for technology developers [6]. As developers cannot foresee all the possible scenarios, the User eXperience (UX) design becomes even more important in the dialogue between people and AI Systems. Considering this new characteristic of having a dynamic behavior, the AI system can have different roles in the interaction with users: (a) it may behave as a traditional system – a black-box with a fixed set of inputs/outputs, or (b) it may behave as an active player in the interaction with humans in the context of their work practices – an "AI-powered user" with its own set of biases and the ability of establishing a cooperative work with end-users, generating new knowledge pieces, and adapting to the user's profile, preferences, and context (goal, environment, emotional status). That last role can have several possibilities still not explored. The possible AI system's different roles should be taken into account throughout development and use times of an AI system.

The communities of Human-Computer Interaction (HCI), with designers, and Software Engineering (SwEng), with developers, use models to represent, discuss and explore different domain scenarios in different stages of the software development process. For this human-computer symbiosis we did not identify any specific models to support the representation of this new symbiotic interaction. In the early stages of development, identify and explore AI systems interaction "what-if" scenarios, with and without users, can enable designers and developers to cover a broader space for problem solving. We believe that it can also enable both communities to identify possible undesirable outcomes and handle them very early in the development process.

In this paper, we present a case scenario with people-AI interaction modeled with an interaction design language and discuss if the notation provides the resources to represent and explore the people-AI interactions possibilities for the development of an AI system. Our discussion scenario is a knowledge-intensive domain where people and the AI system need to interact to execute a set of tasks part of a decision-making process. Our goal is to start an investigation to assess if the existing modeling approaches can be applied or adjusted to represent the people-AI symbiosis scenarios or if we need new modeling approaches for those cases.

2 Models for Understanding and Communication

Models are partial or simplified representations of reality that present a set of objects, with properties (*i.e.*, relations, functions, associated operations, etc.) accurately and consistently defined. Models are abstractions of some object or aspect of the real world. The idea of creating abstractions of the world to understand complex issues is inherent in human behavior. To investigate a complex phenomenon, we may use various models and each of them has a different approach to represent the context. [7] A model should present three main characteristics [8]:

1. **Reduction characteristic** – the model reflects only a relevant selection of properties of modeled object, so that it focuses on certain aspects of interest in the object.
2. **Mapping characteristic** – the model is based on an object in the real world and taken as a reference in relation to some property of this object.
3. **Pragmatic characteristic** – the model must replace the original object, that is, be usable, in some aspect or purpose.

Models, with their diagrammatic representations, are resources that help us solve problems. By modeling, representing the problem domain, we may have different views and consider parts or the whole, simulating different scenarios in search of the solution for a problem [5]. Through interaction with model diagrams, those who seek to solve the problem, have an object that can be manipulated during discussions and experimentation to represent different situations and proposed solutions that continue to foster discussions of possible solutions.

Regardless of the methodology of software development, models are always present as artifacts produced in several stages of the process. Their use is recognized and presented in the most traditional references of the Software Engineering [pp. 126–127, 11] and Human-Computer Interaction [10] and occurs in practically all stages of development: from requirements specification, with domain models and use case model, passing the specification of architecture, to the physical model of the final software database.

Models have different abstraction levels and visions about the software to be developed. The semantics of the model varies according to the need of the development stage. There are various notations defined as standards and used for modeling during software development. The models used for problem contextualization and understanding are powerful communication instruments between all people involved in the process. In Software Engineering [pp. 175–180, 11] and HCI [pp. 380–381, 10], we use models that represent tasks performed by users and how users may interact with the system under development. Those conceptual models help all people involved to express and discuss the scenarios that users and system may work together to achieve their goals.

There are different models and notation for modeling software and interaction aspects. To start our investigation, we chose MoLIC (Modeling Language for Interaction as Conversation) as our modeling language. MoLIC is an interaction design language [2] that perceives the interaction as a conversation between designers (represented by their proxy: the system) and users. It allows the representation of the interaction as a set of conversations that the user can have with the designer's proxy (system), expecting that the designer present the metamessage clearly. The language also serves as an epistemic tool, leading designers to improve their understanding about the problem to be solved and the artifact to be created. The described characteristics, particularly the focus on user-designer (system) interaction, made MoLIC a pertinent modeling language to explore human-AI interaction.

3 People-AI Interaction Modeling Case

We defined a case scenario for building a model to represent and discuss people-AI interactions. We aim to investigate the possible scenarios of people-AI interaction by representing it in the form of a diagram. We decided to use an existing model (MoLIC) to assess if we can use the same notation for this new necessity (represent people-AI interaction), if we can adapt the notation, or if we need to define a new notation to support the people-AI interaction representation and discussion.

3.1 Case Scenario

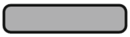
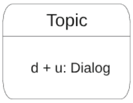
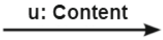
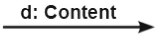
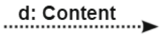
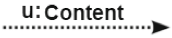
The people-AI interaction case scenario, inspired by CoRgI [14], is of a personal assistant (pA) used by a PhD student (Mary) while studying a paper. This pA has access to Mary's university annotated papers database, built for students to share their notes about papers. Considering the process of "studying a paper", we can list a few tasks performed to achieve the goal (*e.g.* look for other publications from the paper's authors, check the paper's references, etc.). For our scenario, we will focus on the task "annotate paper". There are some premises that need to be defined to contextualize the pA on helping Mary in achieving her goal of learning by annotating the paper. We can define a set of initial questions that may provide contextual information for the task in hand. This contextualization process is done by Mary every time she reads a paper. Therefore, the pA can help her by speeding the task and also making the relations Mary could not do (*e.g.* identify that the 2nd author of the paper works with the most prestigious researcher on a given field and he has a publication with that prestigious researcher). Other questions from this contextualization process that could be considered are:

- Which is the conference of the paper? Is this conference relevant in a specific theme area? Are there other papers in the same conference related to the same topic? Which are the most referenced papers in this conference?
- Which is the main theme of the paper? Which are the keywords of the paper? This impacts on reference suggestion, similar papers.
- Who are the authors? Are they reference author on some research area? Which are their most recent publications?

3.2 Selected Model Notation – Interaction Model with MoLIC

MoLIC (Modeling Language for Interaction as Conversation) is an interaction design language proposed by Barbosa and Paula [2]. As the theory perceives the interaction as a conversation between the designer's deputy and the user, MoLIC allows the representation of the interaction as a set of conversations that the user can have with the designer's deputy (system), expecting that the designer presents the metamessage clearly. The language also serves as an epistemic tool, leading designers to improve their understanding about the problem to be solved and the artifact to be created. The MoLIC models and their descriptions are presented in Table 1.

Table 1. Elements of the MoLIC language

Visual representation	Description
●	**Opening point**
	**Ubiquitous access:** the opportunity for user to change the topic of the conversation, starting from any other scene
	**Conversation scene:** stage of the conversation between user and designer's proxy about a specific subject (topic)
	**User utterance:** exchange of turn, in which the user passes the control of the conversation to the designer's proxy (system)
■	**System processing:** hidden moment for the user, where s/he expects the system is performing the operation that the user requested from the proxy
	**Designer utterance:** change of turn, in which the designer's proxy informs the user on the result of a processing, and can return control of the conversation to the user – if the speech destination is a scene – or proceed to other processing
	**Breakdown utterance (designer):** change of turn, the designer tells the user about an unexpected system processing, and can return control of the conversation to the user – if the speech destination is a scene
	**Breakdown utterance (user):** change of turn, the user tells the designer about an unexpected system processing, and can return control of the conversation to the designer's proxy
◉	**Closing point**

The core element in the MoLIC model is the scene. The scene is composed by (i) the topic of the scene, which is the subject of the conversation caught in the scene, represented by a phrase in the infinitive, which can be read as a talk from the designer and user; and (ii) conversation units that focus on different aspects of the topic of the scene.

4 Modeling People-AI Interaction for "Annotate Paper" Process

The scenario model was built by an experienced modeler and researcher in the modeling notations area. The goal was to use the elements provided by the selected modeling language (MoLIC) to represent and express people-AI interaction for the chosen scenario and report the modeling process.

After the modeler responsible to the model construction (model producer) finished the model, she presented and discussed it with other two people with experience in design and development of systems. They also have experience in developing system with AI features. These two professionals acted as model consumers and helped the model producer to assess and discuss the use of an existing model to represent people-AI interaction. The model built is shown in Fig. 1.

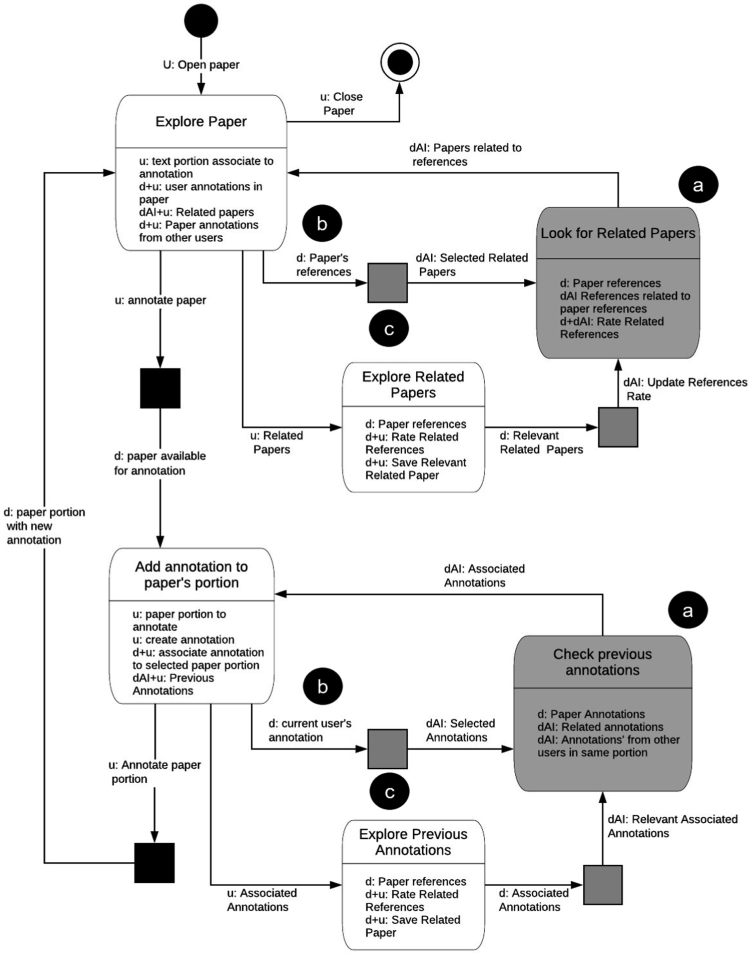
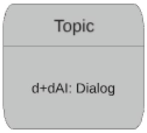
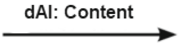
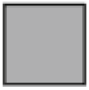

U: Open paper
u: Close Paper
Explore Paper
u: text portion associate to annotation
d+u: user annotations in paper
dAI+u: Related papers
d+u: Paper annotations from other users
dAI: Papers related to references
a
Look for Related Papers
d: Paper references
dAI References related to paper references
d+dAI: Rate Related References
b
d: Paper's references
dAI: Selected Related Papers
c
u: annotate paper
Explore Related Papers
d: Paper references
d+u: Rate Related References
d+u: Save Relevant Related Paper
u: Related Papers
d: Relevant Related Papers
dAI: Update References Rate
d: paper available for annotation
d: paper portion with new annotation
Add annotation to paper's portion
u: paper portion to annotate
u: create annotation
d+u: associate annotation to selected paper portion
dAI+u: Previous Annotations
dAI: Associated Annotations
a
Check previous annotations
d: Paper Annotations
dAI: Related annotations
dAI: Annotations' from other users in same portion
b
d: current user's annotation
dAI: Selected Annotations
c
u: Annotate paper portion
dAI: Relevant Associated Annotations
Explore Previous Annotations
d: Paper references
d+u: Rate Related References
d+u: Save Related Paper
u: Associated Annotations
d: Associated Annotations

Fig. 1. "Annotate Paper" MoLIC model

To build the model with MoLIC we considered as our premise its characteristic of perceiving the interaction as a conversation between the designer's deputy (system) and the user. In that direction, we needed to represent the dialog between user and AI system. In the modeled scenario, we notice that some actions from the AI system would not be presented to or started by the user. But, considering our need to contemplate scenarios where the AI system can interact with the user, we needed to represent it, so it could be considered in the whole system construction. For that, we added three elements in the MoLIC notation. They are not new elements, but adapted ones to express the AI system participation in the communication. MoLIC already represents the system's proxy, but we felt the need to separate it from the AI portion of the system and also represent the dialog between the "traditional system" and the "AI portion" of the system. For that we created an AI Conversation scene, an AI Designer utterance and an AI System processing (Table 2). The explicit representation of the AI portion of the interaction, with user or system, expresses points of turn changing from one part and another in the dialog. According to MoLIC definition [9], the scene change can only be done through a user utterance, but in the dialog between the system and the AI portion the system, a system utterance (d) flows (Fig. 1b) to an AI system processing (Fig. 1c) so it can flow to an AI Conversation scene where there are dialog only between d (system) and dAI (AI portion of system) (Fig. 1a)

Table 2. AI related elements

Visual representation	Description
	**AI Conversation scene:** stage of the conversation between main system designer's proxy and AI designer's proxy about a specific subject (topic). **This scene is not visible to the user.**
	**AI Designer utterance:** change of turn, in which the AI designer's proxy informs the system designer's proxy on the result of a processing.
	**AI System processing:** hidden moment for the user, where he expects that the system is performing the operation that the user requested from the proxy

In our scenario, the dialog between user and dAI led to a feedback cycle (highlighted in Fig. 2). In the AI Conversation scene, some AI-based processing was performed and then presented to user for feedback. In the feedback scene, the user provided input to be processed by dAI in the same AI Conversation scene in a posterior interaction. For example, when the user opens a paper, 'd' informs 'dAI' about that paper references. In the "Look for Related Papers", AI can locate in public datasets other papers related to the current open paper – its theme, its authors, etc. – that may be relevant for the user. After identifying relevant papers, 'dAI' informs the user about relevant papers and the user may "Explore Related Papers" to provide feedback to 'dAI' about those papers for that user.

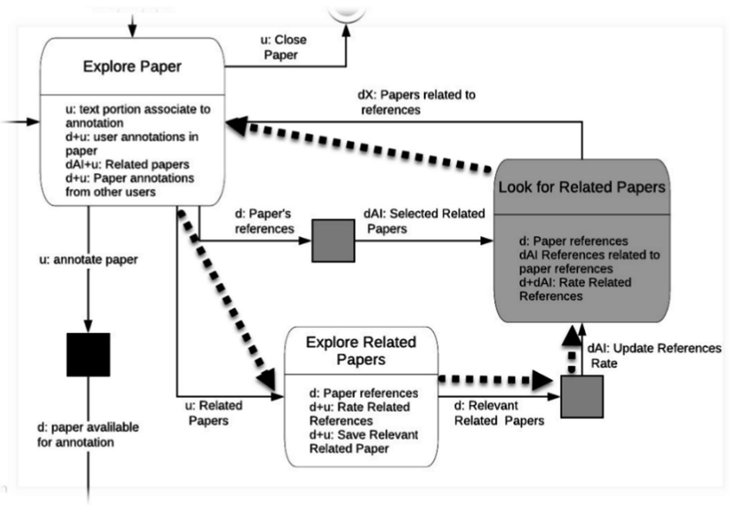
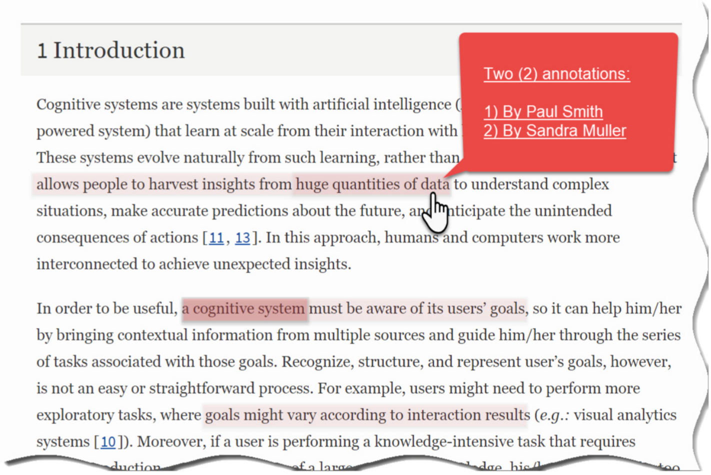

u: Close Paper
Explore Paper
u: text portion associate to annotation
d+u: user annotations in paper
dAI+u: Related papers
d+u: Paper annotations from other users
dX: Papers related to references
Look for Related Papers
d: Paper references
dAI References related to paper references
d+dAI: Rate Related References
d: Paper's references
dAI: Selected Related Papers
u: annotate paper
Explore Related Papers
d: Paper references
d+u: Rate Related References
d+u: Save Relevant Related Paper
dAI: Update References Rate
u: Related Papers
d: Relevant Related Papers
d: paper avalilable for annotation

Fig. 2. dAI-User feedback cycle

An analogous feedback cycle was represented from the scene "Add annotation to paper's portion". The user can see other users' annotation while going over the paper (as illustrated in Fig. 3), but an AI scene can be designed to "Check previous annotations" and relate to users' annotations. The AI system can deal with more data and make relations that the user cannot make alone. Therefore, 'dAI' show the user the associated annotations and the user will tell which ones are relevant for him in that context.



1 Introduction

Cognitive systems are systems built with artificial intelligence (
powered system) that learn at scale from their interaction with
These systems evolve naturally from such learning, rather than
allows people to harvest insights from huge quantities of data to understand complex situations, make accurate predictions about the future, and anticipate the unintended consequences of actions [11, 13]. In this approach, humans and computers work more interconnected to achieve unexpected insights.

Two (2) annotations:

1) By Paul Smith
2) By Sandra Muller

In order to be useful, a cognitive system must be aware of its users' goals, so it can help him/her by bringing contextual information from multiple sources and guide him/her through the series of tasks associated with those goals. Recognize, structure, and represent user's goals, however, is not an easy or straightforward process. For example, users might need to perform more exploratory tasks, where goals might vary according to interaction results (*e.g.*: visual analytics systems [10]). Moreover, if a user is performing a knowledge-intensive task that requires

Fig. 3. Annotations from different users in the same paper

5 Final Remarks and Future Work

The people-AI interaction modeling with MoLIC pointed relevant investigation paths about on how a model can enable the representation and discussion of the people-AI symbiosis. As an initial study, we start the investigation about the application or adjustment of existing modeling approaches to comport the people-AI symbiosis scenarios and the possible need for new modeling approaches for those cases.

The model presented in this paper was a first investigation of one case with a modeling notation to represent and discuss the people-AI interaction. It presented interesting findings to be further explored with other scenarios, more discussion, with diverse people involved in the software development process.

Although the MoLIC notation offers an interesting approach to model the interaction as a dialog between user and system' proxy, it is not a well-known language. The experience with this language can be used as input for future experimentation with other languages, for example, explore UML (Unified Modeling Language) diagrams that are broadly used in the Software Engineering community.

References

1. Artificial Intelligence Journal. https://www.journals.elsevier.com/artificial-intelligence
2. Barbosa, S.D.J., de Paula, M.G.: Designing and evaluating interaction as conversation: a modeling language based on semiotic engineering. In: Jorge, J.A., Jardim Nunes, N., Falcão e Cunha, J. (eds.) DSV-IS 2003. LNCS, vol. 2844, pp. 16–33. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39929-2_2
3. Biran, O., Cotton, C.: Explanation and justification in machine learning: a survey. In: Aha, D.W., et al. (eds.) IJCAI 2017 Workshop on Explainable AI (XAI), Melbourne, Australia, pp. 8–13 (2017)
4. Floridi, L., et al.: AI4People - an ethical framework for a good AI society: opportunities, risks, principles, and recommendations. Minds Mach. **28**(4), 689–707 (2018)
5. Funt, B.V.: Problem-solving with diagrammatic representations. Artif. Intell. **13**(3), 201–230 (1980)
6. Hill, C., et al.: Trials and tribulations of developers of intelligent systems: a field study. In: 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 162–170. IEEE (2016)
7. Lieberman, B.A.: The Art of Software Modeling. Auerbach Publications, Boca Raton (2006)
8. Muller, P.-A., et al.: Modeling. Softw. Syst. Model. **11**(3), 347–359 (2012)
9. Nass, C., et al.: Computers are social actors. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Celebrating Interdependence - CHI 1994, pp. 72–78. ACM Press, Boston (1994)
10. Preece, J., et al.: Interaction Design: Beyond Human-Computer Interaction. Wiley, Chichester (2015)
11. Pressman, R.S.: Software Engineering: A Practitioner's Approach. Palgrave Macmillan, Basingstoke (2005)
12. Rossi, F., Mattei, N.: Building Ethically Bounded AI (2018). arXiv preprint: arXiv:1812.03980
13. Samek, W., et al.: Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models (2017). arXiv preprint: arXiv:1708.08296

14. Segura, V., Ferreira, J.J., Fucs, A., Moreno, M.F., de Paula, R., Cerqueira, R.: CoRgI: cognitive reasoning interface. In: Kurosu, M. (ed.) HCI 2018, Part II. LNCS, vol. 10902, pp. 398–409. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91244-8_32
15. The Journal of Artificial Intelligence Research (JAIR). https://www.jair.org/index.php/jair
16. Unified Modeling Language (UML). http://www.uml.org/