Chatbot Design - Reasoning about design options using i* and process architecture

Zia Babar[1], Alexei Lapouchnian[2], Eric Yu[1,2]

[1]Faculty of Information, University of Toronto
[2]Department of Computer Science, University of Toronto
zia.babar@mail.utoronto.ca, alexei@cs.toronto.edu,
eric.yu@utoronto.ca

Abstract. Software systems are often designed without considering their social intentionality and the software process changes required to accommodate them. With the rise of artificial intelligence and cognitive services-based systems, software can no longer be considered a passive participant in a domain. Structured and methodological approaches are required to study the intentions and motives of such software systems and their corresponding effect on the design of business and software processes that interact with these software systems. This paper considers chatbots as domain example for illustrating the complexities of designing such intentional and intelligent systems, and the resultant changes and reconfigurations in processes. A mechanism of associating process architecture models and actor models is presented. The modeling and analysis of two types of chatbots - retrieval-based and generative - are shown using both process architecture and actor models.

Keywords: Enterprise Modeling, Goal Modeling, Actor Modeling, Software Processes, Chatbot, Adaptive Enterprise.

1   Introduction

Rapid advancement in software and emerging technologies is influencing the design of enterprises while promoting continuous and ongoing cycles of innovation and transformation [1]. Software systems and software processes are enablers of such enterprise transformation [2] and structured methodologies need to be developed that would aid in selection of suitable designs and configurations of such software systems and processes. However, such investigations cannot be done from just one perspective and process models and actor models need to be collectively aiding in the design of enterprise software while the enterprise itself is undergoing rapid cycles of transformation and innovation. Integrating these modeling approaches together would allow for meaningful and multi-dimensional analysis for determining alternative configurations of processes and the analysis of requirement satisfaction. This promotes deeper domain reasoning and a judicious understanding of the implications of design actions.

Chatbots are an appropriate selection for introducing and illustrating some of the concepts in this paper due to their apparent intentional nature. A chatbot is a software

system that converses with human users using natural language, either in verbal or textual form [3]. Recent advances in cognitive services, machine learning, natural language processing and artificial intelligence have seen chatbots become ever more pervasive and being put to use in commercial applications [4]. Generally, they are being developed as an additional (conversational) interface to existing software applications. Chatbots work by producing a response to an inquiry from a human user. Despite being software systems, chatbots appear to exhibit intentionality and are active (as opposed to passive) participants in the overall domain and their interaction with the user. They have defined functional goals, such as conversing with the user, processing information intelligently, sourcing all available data while processing a response, etc. These functional goals are accompanied by non-functional goals like providing a natural conversational experience, ensuring the incorporation of appropriate context into the response, being coherent and have sensible responses, providing a diverse set of responses, etc. Thus, it would be appropriate to model chatbots as social actors and utilize the i* framework [5] for analyzing their strategic intentions, rationales, and dependencies, in conjunction with the process of designing and developing these chatbots, while considering bi-directional influences between actor and process perspectives.

Chatbots can be classified in one of two ways depending on their sophistication and the complexity involved in producing the response (to the inquiry):

- *Retrieval-based* chatbots are simple in design and are generally used for situations where a simple response or action is required to a user inquiry or command.
- *Generative* chatbots are more complex and are generally used in situations which require the construction of a unique and contextually appropriate response to user inquiries.

2 Analyzing Social Implications of Process Architecture Reconfigurations

The visual depiction of the social aspect is done using the i* framework while the process related elements and activities are modeled using a process architecture model that was previously introduced in [6][7][8][9]. A process architecture model is needed to be able to model relationships among multiple processes, such as between software development processes and the software usage process. The primary reason for focusing on process architectures (and not on individual business processes) is to investigate inter-process relationships and configurations; in particular, to identify variations of how functionality can be distributed across processes in the enterprise, to explore the different possible process relationships, and to model how these choices affect enterprise objectives. Following the above-mentioned process architecture modeling framework, *Process Elements* (PEs) are basic process activity units that produce some output or outcome. A collection of PEs that are be executed collectively as part of the same execution cycle is referred to as a *Process Stage* (PS).

The two perspectives -- actor and process -- can be associated by considering common constructs that appear in both models, represented as a process segment in the process-based model and a task (or operationalized goal) in the actor-based model. The models can be associated together at one of two levels,

- **PS Level:** PS are single-purpose process structures and are meant to accomplish certain enterprise functional requirements (goals) while adhering to non-functional requirements (softgoals). PS are executed by participants (or actors).
- **PE Level:** In an actor model, a goal is eventually achieved through some task(s), which map to a PE in the process model (assuming both models are decomposed to the same granularity), thus allowing for model linking at a more granular level.

Thus, process goals and process activities would be mapped to actor goals and actor tasks respectively, with mechanisms and data flows being represented as dependent resources in the actor models. Existing actor modeling techniques (such as i*) can be used (with extensions and annotations) to signify the association between models from both modeling perspectives while allowing for traceability and movement of changes in one modeling perspective to the other. A PS may have multiple associated actors to ensure the attainment of the process stage goal. Although in simple cases, fulfilling the goal of a process stage maybe entirely self-contained within a single actor boundary. In general, this approach is not limited to a one-to-one mapping between the elements of the different modeling approaches, nor are they expected to be at the same level of granularity and abstraction.

Variation points (VP) in the process architecture model can signify alternative process configurations (also called *variants*). An actor model can be associated at this VP by having alternatives shown as one of several means-to-end for an actor goal. Actor goal satisfaction techniques will be employed to determine the optimum alternate (based on softgoal tradeoff analysis) which maps to a process architecture configurations. Care must be taken when proposing an alternate process configuration at a VP as the proposed alternate should not result in goal non-satisfaction for other process stages in other parts of the process architecture.

2.1   Retrieval-Based Chatbots

Retrieval-based chatbots are configured to select a response from a predefined set of responses based on the response's suitability to the inquiry. As the responses are pre-constructed, there is no chance of grammatical mistakes and the primary intelligence required on the chatbot's side is to retrieve an appropriate response to match the inquiry. This ensures that the chatbots are simple in nature, and as a result they are unable to handle advanced queries from the user. There are several third-party frameworks and solutions available that are able to provide message parsing and response selection capabilities, thus retrieval-based chatbots are quick to construct but have limited flexibility and use in cases where domain specific responses are required.

Fig. 1 shows the dependency relationships that exist between the retrieval-based chatbot on other actors (including the internal chatbot rationales). The i* model allows the determination of the type of chatbot that needs to be developed, and the nec-

essary process structure that should be formed, to allow the satisfaction of various goals. In this case, the simplicity of the intentions and motives of the retrieval-based chatbot are reflected in a less complex process architecture (Fig. 2). The two tasks, **Understand Request** and **Construct Response**, are attained by corresponding process stages **Understand User Request** and **Construct User Response**.
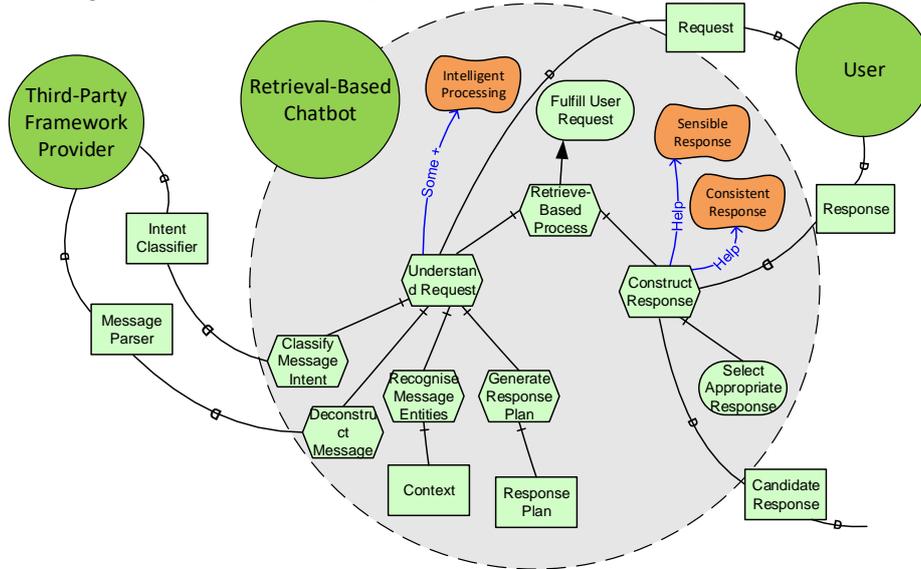

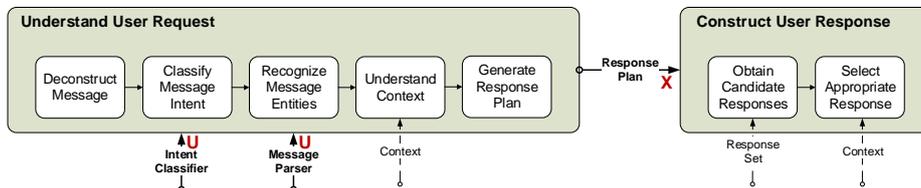**Fig. 1.** i* Strategic Rationale Model for Retrieval-Based Chatbots

Fig. 2. Process Architecture Model for Retrieval-Based Chatbots

2.2   Generative Chatbots

Generative chatbots attempt to construct a response by tracking the ongoing conversation, utilizing the history of exchanges with the users, and by matching appropriate responses based on a semantic understanding of the inquiry. As the responses are constructed by the chatbot in real-time, there is a chance of the chatbot making mistakes in sentence construction. Additionally, the responses as constructed by a generative chatbot may be domain specific (for example, queries about medical conditions) which require access to a domain-specific knowledge base. Due to their specialized and complicated nature, often such chatbots have to be developed within an organization itself without having the benefit of using pre-built third-party frameworks.
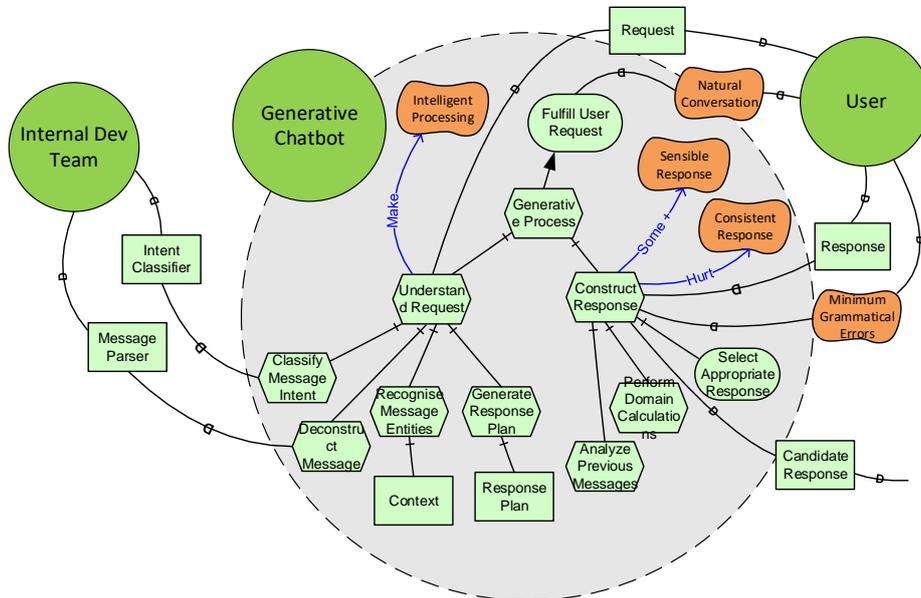
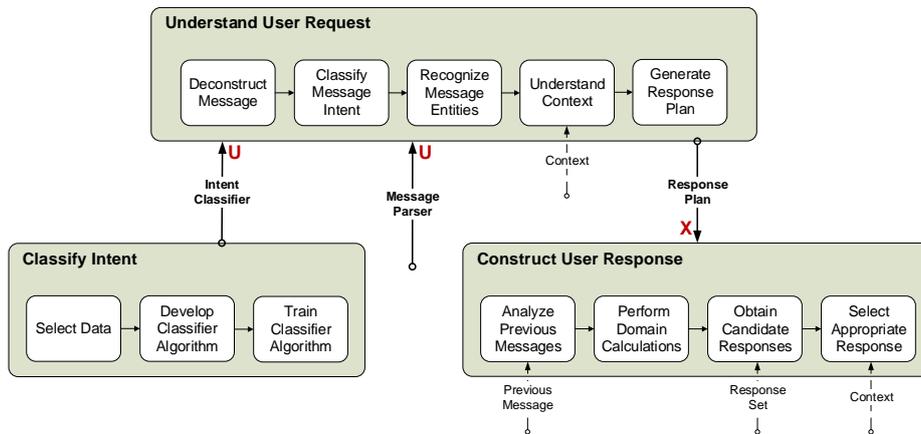**Fig. 3.** i* Strategic Rationale Model for Generative Chatbots

Fig. 4. Process Architecture Model for Generative Chatbots

Fig. 3 shows the dependency relationships that exist between the generative chatbot on other actors with the process architecture model depicting the processes involved in processing a user request shown Fig. 4. The generative chatbot requires an additional level of intelligence in order to be able to carry out a natural conversation with a human user. While the tasks (**Understand Request** and **Construct Response**) remain the same, the softgoals **Intelligent Process** and **Natural Conversation** indicate a different approach needs to be adopted to ensure their satisfaction. This requires the addition/modification of new tasks decompositions and dependencies on **Internal Dev Team**. The corresponding process architecture model (Fig. 4) reflects this by including a new process stage, **Classify Intent**, whose output is used by the

Chatbot associated processes. While these diagrams are simplistic in nature and illustration, they do indicate the nature of propagation and traceability analysis that can be performed by associating actor models with process architecture models.

3 Conclusions and Future Work

The above is part of our ongoing research [6][7][8][9] for developing a framework for characterizing fundamental types of software-enabled enterprise transformations where we consider the upstream factors in the design of software systems and processes which can be traced to enterprise business objectives and social contexts, and the downstream effects of selecting alternative design and configuration options for software systems on software processes. In this paper, actor models are shown as a means of determining optimum process architecture configurations based on the unique strategic motives and intentions of those actors. They extend and complement the process architecture models by allowing for the inclusion of social and agent relationships with the process participants (involved in the enactment of process activities) being treated as actors from a social modeling perspective. Association points between the models allow for navigating between the modeling notations, with the models themselves consistently reflecting the same domain reality. In future work, we plan to continue to explore various association points (including possible modeling patterns) between the two kinds of models and how change on one modeling perspective can be traced to, and be influenced by the other. Additionally, we'll investigate possible extensions to the i* framework for attaining the above.

4 References

1. Wilkinson, M.: Designing an "adaptive" enterprise architecture. BT Technology Journal, 24(4), pp. 81–92 (2006)
2. The Economist: Organisational Agility: How Business can Survive and Thrive in Turbulent Times. A report from The Economist Intelligence Unit (2009)
3. Surmenok, P.: Chatbot Architecture, March 10, 2017, http://pavel.surmenok.com/2016/09/11/chatbot-architecture/
4. Augello, A., Gentile, M., Weideveld, L., Dignum, F.: A model of a social chatbot. In Intelligent Interactive Multimedia Systems and Services, pp. 637-647. Springer (2016)
5. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering. MIT Press (2011)
6. Lapouchnian, A., Yu, E., Sturm, A.: Redesigning process architectures towards a framework of design dimensions. In 9th International RCIS Conf., pp. 205-210, IEEE (2015)
7. Lapouchnian, A., Yu, E., Sturm, A.: Design Dimensions for Business Process Architecture. In International Conference on Conceptual Modeling, pp. 276-284, Springer (2015)
8. Lapouchnian, A., Yu, E.: Exploiting emergent technologies to create systems that meet shifting expectations. In Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, pp. 371-374, IBM Corp (2014)
9. Babar, Z., Lapouchnian, A., Yu, E.: Modeling DevOps Deployment Choices using Process Architecture Design Dimensions. In The Practice of Enterprise Modeling, Springer (2015)