Planning for Automated Composition of Aggregated Assistants

Tathagata Chakraborti · Shubham Agarwal · Krissy Brimijoin · Prerna Agarwal
Yara Rizk · Dario Silva Moran · Scott Boag · Yasaman Khazaeni

IBM Research AI

Abstract

An aggregated assistant is realized as an orchestrated set of individual capabilities called skills. In this demo, we will show how complex behaviors of such an assistant can be composed on the fly using automated planning.

Video link: https://youtu.be/K7FPcl-IYgE

Conversational assistants such as Siri, Google Assistant, and Alexa have found increased user adoption over the last decade and are adept at performing tasks like setting a reminder or an alarm, putting in an order online, control a smart device, and so on. However, the capability of such assistants remain quite limited to episodic tasks that mostly involve a single step and do not require maintaining and propagating state information across multiple steps.

A key hurdle in the realization of sophisticated AI assistants is the complexity of the programming paradigm – at the end of the day, end-users and developers who are not necessarily subject matter experts have to be able to build and maintain these assistants. A particular architecture that has emerged recently to address this issue is that of an "aggregated assistant" where the assistant is build out of individual components called skills. Skills are the unit of automation and they perform atomic tasks that can be composed together to build the assistant capable of performing more complex tasks. Prominent examples of this include IBM Watson Assistant Skills[1] and Amazon Alexa Skills[2].

This setup is not particularly confined to personal assistants either. An increasingly popular use of assistants is in enterprise applications. Here also, examples of aggregated assistant can be seen in offerings from Blue Prism[3], Automation Anywhere[4], and others. These individual skills belong to the class of Robotic Process Automation (RPA) tools that automate simple repetitive tasks in a business process.

In recent work, we have been exploring the prospect of using automated planning techniques in the context of aggregated assistants. (Sreedharan et al. 2020) The typical role of an automated planner here is to look at an interpretation of the user utterance (or, in general, a system event), cast

[1]IBM Watson Assistant Skills https://ibm.co/33f58Hc
[2]Amazon Alexa Skills: https://amzn.to/35xK2Xv
[3]Blue Prism Digital Exchange: https://bit.ly/2Ztzdla
[4]Automation Anywhere Bot Store: https://bit.ly/33hcr12

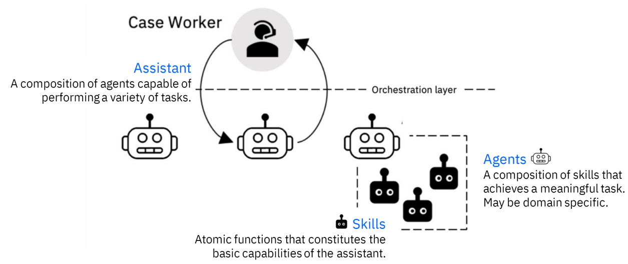


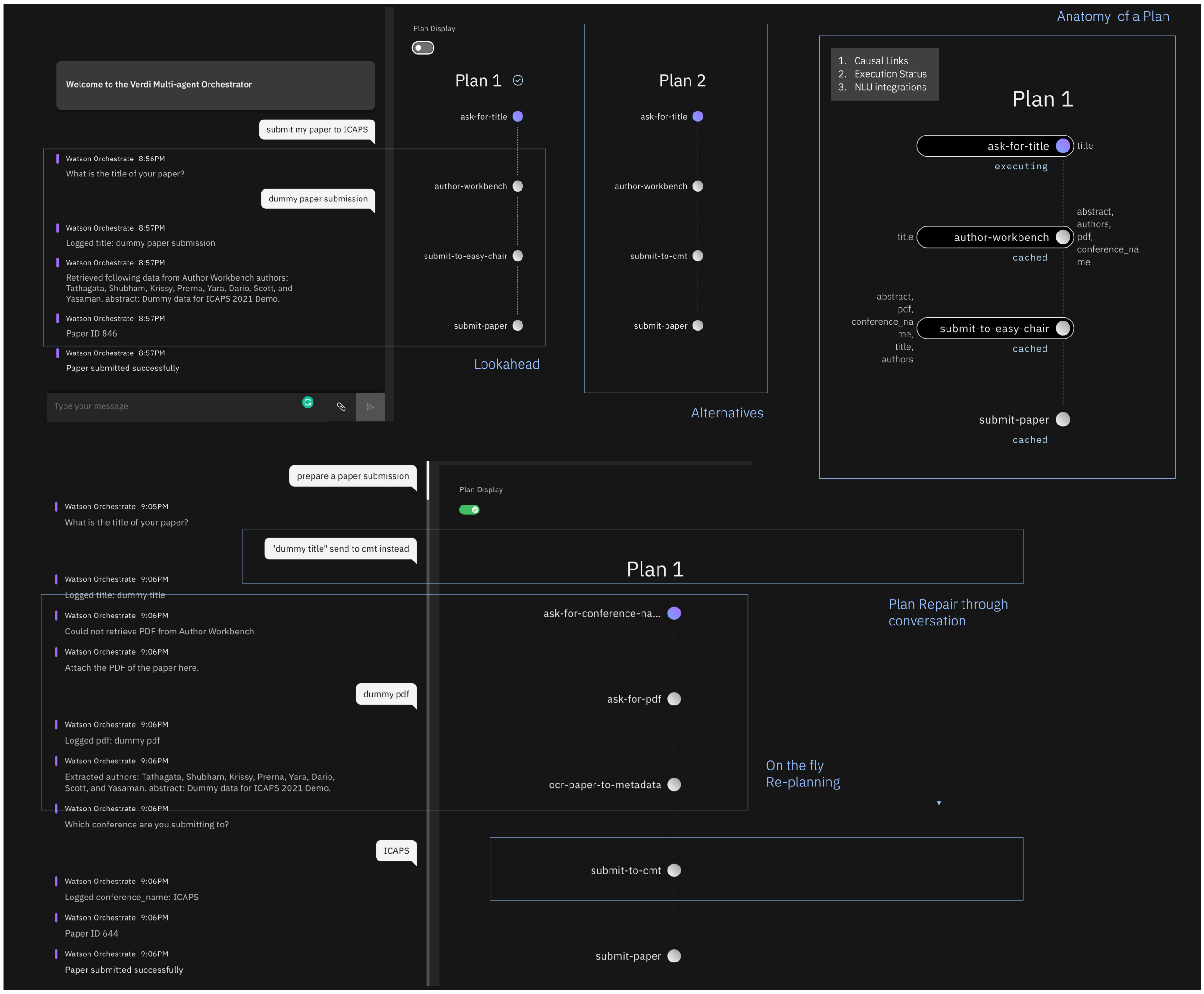
Figure 1: Simplified architecture diagram of an aggregated assistant (Rizk et al. 2020) illustrating the compositional nature of the assistant-agent-skill hierarchy.

that interpretation into a planning problem, and generate a sequence of agents or skills to solve the task at hand. The interpretation here can be a set of agents that have self-selected themselves as relevant to the user utterance, a set of relational constraints derived from the utterance (e.g. *"do this then that"*), or in general, some higher level abstract goal (e.g. *"apply for a loan"*).

System Overview

Our aggregated assistant, as shown in Figure 1, is an orchestrated set of agents. The agents are themselves a composition of skills. A skill is the unit of automation. Each skill and agent has specified with it a signature that tells the planner what information it consumes and what information it produces. Apart from the conversational aspects of it, this should remind the reader of web service composition. The rest of the constraints are derived from the user utterance. In this demo, we will focus on the orchestration layer, and thus the automated composition of agents only, and not on the composition of skills into agents. The flow of control at the orchestration layer proceeds as follows:

- Every time the user says something, the natural language processing components of the agents evaluate the utterance and declare their relationship or relevance to it.
- The planner consumes this information along with the signature of all the available agents and other constraints that may be derived from the user utterance and produces a sequence of agents that can satisfy all the constraints and the



agent selection as per the signatures. It sends the first step in the sequence for execution.

- The agent in the first step (may be different from the selected ones) executes and goes into one of four modes – synchronous or asynchronous hold indicating that they need to interact with the user or need some time to finish; or success and failure, which are both finish states.
- Based on the status of execution, the planner either replans or continues onto the next step.

Live Demonstration

During the demo, we will go through the simple process of submitting a paper to a conference. When the user types in their intent to do this, they get to see all the agents under the hood, that can possibly support this task. You also get to see what the system has planned ahead. This view not only shows you the plan under execution but also alternatives that the system considered. They can hover over individual steps in the plan to see what each of these steps consume and produce, and what are their sources and destinations. This establishes the provenance of the data being exchanged among members of the aggregate constructed on the fly.

We will then explore how the planner responds to internal failures and user interactions on the fly. In this scenario, the user looks at the rest of the plan and recognizes that the system is thinking of pinging EasyChair with the paper information, although it should go to CMT instead. In order to change the system's plan, the user types in more information. This is an example of collaborative plan repair with the assistant. Furthermore, the plan evolves as the assistant encounters failures and replans to adjust. Modeling of such replanning and repair is essential to user interaction, since the planner is always working with an incomplete model.

References

Rizk, Y.; Boag, S.; Khalaf, R.; Isahagian, V.; Khazaeni, Y.; and Muthusamy, V. 2020. Conversational Digital Assistant for Intelligent Process Automation. In *BPM RPA Forum*.

Sreedharan, S.; Chakraborti, T.; Rizk, Y.; and Khazaeni, Y. 2020. Explainable Composition of Aggregated Assistants. In *ICAPS Workshop on Explainable AI Planning*.