Towards Continual Learning in Interactive Digital Assistants for Process Automation

(Short Paper)

Praveen Venkateswaran, Vinod Muthusamy, Yara Rizk and Vatche Isahagian

IBM Research AI, Cambridge, MA, USA

1. Introduction

Business process automation (BPA) has recently emerged as a multi-billion dollar industry, revolutionizing how businesses manage their processes [1]. It develops solutions for automated decision making, data management, and workflow optimization. Many of these solutions leverage digital assistants that can speak the language of business users to make these automations more accessible to non-tech-savvy end-users [1, 2, 3]. In enterprise settings, conversational assistants may be used by teams of people to automate role-based tasks (e.g., in HR, sales, or accounting) and they may want to refine them over time to meet changing business needs.

However, existing commercial offerings of digital assistants for process automation from companies like IBM, Microsoft, and UIPath provide a fixed set of automation capabilities. Expanding their functionality is often restricted to domain experts with programming skills. Hence, most end-users are unable to customize or create new process automation capabilities to suit their process management needs. This, in-turn, can result in user frustration and decreased adoption of process automation tools. One possible approach to increase adoption is to empower end-users to intuitively and naturally develop customized and improved automation capabilities from existing automations within these digital assistants by teaching them. Unlike the personal chatbots in [4], digital assistants in business automation settings may be more role-driven. Hence, they would be used by multiple end-users who need to customize them based on company processes, roles and team preferences.

In this paper, we posit that the next generation of digital assistants within enterprise settings will be iteratively taught in-situ through intuitive *multi-modal interactions* dependent on the type of task and skill of the teacher(s). These assistants should be characterized by their ability to continually learn, customizability by end-users with minimal coding expertise, evolution with changing business processes, and generalizability to new process automation tasks by understanding their context. We also discuss some of the challenges of realizing these assistants, focusing on challenges in machine learning, process management, and interactive task learning.

PMAI@IJCAI22: International IJCAI Workshop on Process Management in the AI era, July 23, 2022, Vienna, Austria

[1] https://www.gartner.com/en/newsroom/press-releases/2021-04-28-gartner-forecasts-worldwide-hyperautomation-enabling-software-market-to-reach-nearly-600-billion-by-2022

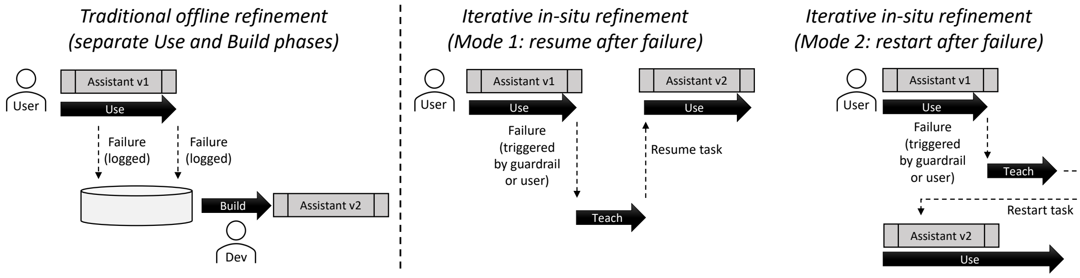
Traditional offline refinement
(separate Use and Build phases)
User
Assistant v1
Use
Failure (logged)
Failure (logged)
Build
Assistant v2
Dev
Iterative in-situ refinement
(Mode 1: resume after failure)
User
Assistant v1
Use
Failure (triggered by guardrail or user)
Teach
Resume task
Assistant v2
Use
Iterative in-situ refinement
(Mode 2: restart after failure)
User
Assistant v1
Use
Failure (triggered by guardrail or user)
Teach
Restart task
Assistant v2
Use

Figure 1: An improved iterative in-situ approach for process automation by teaching digital assistants.

2. The Next Generation of Digital Assistants

The traditional development cycle, in Figure 1 (left), splits the use and build phases of digital assistants for process automation. Run-time failures encountered by users are logged, and used by expert developers to eventually build and deploy new versions of the assistants with improved automation skills. This is a relatively longer process, which due to lack of developer resources, can result in developers prioritizing the creation of some skills and resolution of certain issues over others, leaving behind frustrated and disconnected users.

We instead envision an *iterative in-situ* refinement of digital assistants, as in Figure 1 (right), where users are empowered to address automation shortcomings by teaching the assistant new capabilities. There are three aspects to this approach that differ from the traditional way that assistants are built. First, failures while using a digital assistant interrupt the user's task and the system enters a teaching mode. These failures may occur due to developer or business-defined guardrails that limit the execution bounds for the assistant, or because of the user explicitly indicating they were dissatisfied with the assistant's actions. For example, an assistant used to book a trip may only know what to do for domestic trips, and a guardrail can be triggered when a user tries to book an international trip. Second, the user is repairing a specific failure in the context of the task they were in. Since teaching happens in the context of the failures, it enables the user to quickly address the issue. We believe this is an easier way for non-developers to teach and refine the assistant. Third, the same user interacting the assistant is responsible for teaching the assistant how to handle the failure. This empowers users to evolve and customize the assistant according to their needs.

There can be two variants of the iterative in-situ refinement approach: *Mode 1* where the user resumes their task after repairing the failure; and *Mode 2* where the user restarts their task with the new version of the automation. The latter may be appropriate if there have been no side effects so far, or if the automation can roll back or compensate the steps taken so far.

This in-situ teaching mode needs to be intuitive to non-tech-savvy end-users by leveraging various input modalities tailored to automation tasks and users' teaching abilities. For some tasks or users, it may be easier to "show" the assistant how to perform the task and thus requires screen recording capabilities. Other tasks may be more easily taught through natural language instructions. While [5] discuss multi-modality in the context of basic interactions, many of the concepts can be adapted for our envisioned teaching mode. Furthermore, some users may be more articulate than others in communicating their automation needs and hence, the assistant must identify when more iterations with the user are required to fill information gaps.

3. Challenges

Realizing this next generation of digital assistants with multi-modal, iterative, in-situ teaching requires addressing various research challenges and integrating multiple technological advancements before successful deployment is possible.

Metacognition: An important pre-requisite for learning in digital assistants is their ability to recognize and understand what they do, and do not know [6]. End-users may trigger multiple automation skills, often in specific sequences, making it critical for assistants to identify gaps in their capabilities. Without this, users may have to repeatedly teach the same skill, or assistants may try to execute tasks that they are incapable of.

Natural language understanding: Understanding the underlying intentions of user inputs is critical for assistants to execute the right skills or determine when to get more information and learn. Analyzing inputs in isolation can result in misinterpretation since information may be provided through different modalities, combined across multiple inputs, or be missing altogether. Prior work [7] limits end-user vocabulary, which is not practical for process automation settings where users have different backgrounds, business needs, and expertise. For instance, the terminology to automate the same skill can differ between IT support and HR, and novice users may not articulate their needs as clearly as experts. Improving existing language models (e.g., BERT) is required to realise learning-enabled automation assistants, including the ability to interpret domain specific terminology and process model concepts, learn ontologies, interpret task signatures and inter-dependencies, etc.

Interactive and informative experience: The teaching process can result in assistants behaving differently from user expectations (e.g., learning incorrect skills or misinterpreting key information), which can prove costly. Hence, it is important to provide informative responses reflecting the assistant's understanding at every stage of the teaching process. This includes "failing gracefully" by defining mechanisms to provide interpretable error messages and seek appropriate user feedback, when users' expectations are violated. The vast space of multi-modal responses makes generating the appropriate response a difficult problem.

Learning new skills: Automation skills vary in complexity, scope, and usage, challenging assistants to learn new skills. Some may require incrementally updating to existing skills, or learning a model through user-provided input/output examples, or even synthesising new code from specifications, making it important to identify what and how to learn. Large teams of users may teach overlapping skills, requiring assistants to disambiguate. Also, new skills must be immediately available for use. Analyzing interaction logs to provide capabilities such as auto-complete, and continually updating skills with changing business and user needs are important capabilities that can make the assistant's teaching model more intuitive.

Customization and generalization: Existing assistants often require users to provide individual inputs to execute each skill and its parameters, incurring significant overhead for repeated executions of large workflows. The ability to teach customizations such as macros to execute groups of skills with a single input, pre-filling process parameters, and modifying process execution should be enabled in assistants using efficient representations (e.g., user profiles). However, since skills will be used by multiple users with different business needs, assistants should be able to generalize the taught instructions and extrapolate to similar tasks. This will enable them to perform an ever-increasing number of tasks, require fewer instructions, and learn continually from subsequent interactions. Furthermore, new skills may be taught to the assistant by multiple users (with different "teaching" abilities) which requires identifying similar instructions and generalizing from them. While previous work focused on generalization in specific domains [8], expanding the scope is challenging to achieve and requires identifying entities, their values and the tasks that they can be incorporated in.

Catastrophic forgetting: Teaching digital assistants new tasks runs the risk of conflicts between new capabilities and old ones or even "forgetting" how to perform previously known tasks. Existing literature [9] has studied this phenomenon in applications like transfer learning where neural networks evolved their learnt weights to a point where they were no longer able to perform previously learned tasks. In essence, the assistant would be replacing previous knowledge with new knowledge as opposed to combining and augmenting their understanding of the tasks. This would create a frustrating experience for end-users who may find themselves re-teaching the system previously learned tasks. Hence, to create a digital assistant that can truly learn from user interactions, we need to address this problem.

References

[1] J. Sen, F. Ozcan, A. Quamar, G. Stager, A. Mittal, M. Jammi, C. Lei, D. Saha, K. Sankaranarayanan, Natural language querying of complex business intelligence queries, in: Proc. Int. Conf. Management of Data, 2019, pp. 1997–2000.

[2] S. Mani, N. Gantayat, R. Aralikatte, M. Gupta, S. Dechu, A. Sankaran, S. Khare, B. Mitchell, H. Subramanian, H. Venkatarangan, Hi, how can i help you?: Automating enterprise it support help desks, in: Proc. AAAI Conf. Artificial Intelligence, volume 32, 2018.

[3] Y. Rizk, V. Isahagian, S. Boag, Y. Khazaeni, M. Unuvar, V. Muthusamy, R. Khalaf, A conversational digital assistant for intelligent process automation, in: Int. Conf. Business Process Management, Springer, 2020, pp. 85–100.

[4] F. Daniel, M. Matera, V. Zaccaria, A. Dell'Orto, Toward truly personal chatbots: On the development of custom conversational assistants, in: Proc. 1st Int. Workshop on Software Engineering for Cognitive Services, 2018, pp. 31–36.

[5] J. O. Kephart, Multi-modal agents for business intelligence, in: Proc. 20th Int. Conf. Autonomous Agents and MultiAgent Systems, 2021, pp. 17–22.

[6] M. T. Cox, Metacognition in computation: A selected research review, Artificial intelligence 169 (2005) 104–141.

[7] A. Azaria, S. Srivastava, J. Krishnamurthy, I. Labutov, T. Mitchell, An agent for learning new natural language commands, AAMAS 34 (2020) 1–27.

[8] M. Nicolescu, M. Mataric, Natural methods for robot task learning: Instructive demonstrations, generalization and practice, in: AAMAS, 2003, pp. 241–248.

[9] R. M. French, Catastrophic forgetting in connectionist networks, Trends in cognitive sciences 3 (1999) 128–135.