



# Chatting About Processes in Digital Factories: A Model-Based Approach

Donya Rooein<sup>1(✉)</sup>, Devis Bianchini<sup>2</sup>, Francesco Leotta<sup>3</sup>, Massimo Mecella<sup>3</sup>,  
Paolo Paolini<sup>1</sup>, and Barbara Pernici<sup>1</sup>

<sup>1</sup> Politecnico di Milano, Milan, Italy  
{donya.rooein,paolo.paolini,barbara.pernici}@polimi.it

<sup>2</sup> Università di Brescia, Brescia, Italy  
devis.bianchini@unibs.it

<sup>3</sup> Sapienza Università di Roma, Rome, Italy  
{leotta,mecella}@diag.uniroma1.it

**Abstract.** Using chatbots in digital factories, to interact with devices through instant messages and voice commands, can make the understanding of underlying manufacturing, logistic and business processes easier for workers. Intelligent chatbots can provide flexible conversations and tailor them to the specific users who are interacting. The iCHAT framework conceptually represents all the aspects related to a conversation, with different facets for the user, the conversation flow, and the conversation contents, and combining them to obtain a flexible interaction with the user. In digital factories, flexible production is driven by processes combining different services. In this paper, we present an original approach extending iCHAT to be able to chat about processes, aiming at instructing a worker about a process.

**Keywords:** Conversational user interface · Chatbot modelling · Digital factory · Industry 4.0

## 1 Introduction

Given the dynamic nature of today's business world, organizations should quickly respond to a wide range of possible changes in their environment. The changes can have either short-term effects, like disruptions in the supply or production chain, or long-term effects, like changes in one part of the production system that affect other subsystems and processes. Changes may also have an impact on controlling and monitoring essential parameters, that refer to the situation in which a production process is being executed. The introduction of data-driven solutions and enabling technologies, to transform collected data into actionable insights, has raised the complexity of monitoring and controlling activities. In addition, different roles, including business managers, data analysts, machine operators and designers, need to be informed on the processes and their changes to perform their job. These workers: (a) need tools to find the right information

within a complex ecosystem of information, configurations, workflows, sensors and devices; (b) should be enabled to use their own language, that is far from the technical notation used to store and manage these kinds of information; (c) might have different levels of knowledge about the processes.

Conversational interfaces, such as the ones implemented by chatbots, might adapt to the new structure(s) and behavior(s) of the production processes. They may help workers, who have not exactly in mind what they are searching for or have partial knowledge about the process, to properly refine and focus questions, moving across many pathways in the production process. Using chatbots can mitigate the gap between a worker’s language and formal notation used to model processes and data. Moreover, when the production is running, and some activities have already been carried out, a conversational interface like a chatbot may support a worker in learning the process, by making explicit for worker’s the current context.

The goal of this paper is to present a flexible approach to generate conversations with a chatbot taking into account the structure of the process.

In this paper, we focus on the adoption of chatbots to *teach* a worker about the process structure. The present work is based on a conversational approach to generic education [3], proposing new general conversational structures for process teaching. The proposed approach is able to adapt to the context of the interaction and the context of the environment as it evolves during the process execution, going beyond the traditional question-answering mechanisms used for chatbots. The chatbot helps a learner in a possibly complex learning pathway for navigation over the process tasks. We aim to support the changing role of human beings in the digital factory landscape, according to the human-in-the-loop paradigm [8], where workers are expected to enjoy greater responsibility, to act as decision makers and to take on supervising tasks, while relying on chatbots for more operative levels (product quality control, process control and productivity, anomaly detection and health assessment). To make our contribution clear in the paper, we use an example of MyMuffin factory from [2], providing a conversational interface in a small factory to support the digital transformation.

The paper is structured as follows. In Sect. 2, we introduce challenges and motivations about using chatbots in digital factories. In Sect. 3, we introduce the iCHAT framework which is the basis for the rest of the paper and how the generic domain conversations by chatbots are managed, based on a data-driven approach. In Sect. 4 we discuss how different types of pathway structures about a process can be represented. In Sect. 5, we describe the process in MyMuffin factory and present an example of conversation showing how it can be dynamically generated by our data-driven approach. In Sect. 6, we discuss the state of the art in chatbots and in process- and service-based digital factories.

## 2 Challenges for Chatbots in Digital Factories

The introduction of a chatbot in digital factories aims to address the following challenges.

**Adaptation to the Users’ Language.** Formal process modeling notation might not be suitable or understandable for all the users who are interested in learning about the business process. Chatbots using natural language, possibly adopting general-purpose and domain-specific terminology, may mitigate the gap between the business process notation and the users’ language. The chatbot can also act as a mediator between raw data coming from sensors and the users, presenting information in such a way to ease the decision making about process instances.

**Iterative Questions About the Process.** The conversation style of the chatbot might enable a progressive refinement of the information to be provided to users, also proposing different options to users and relying on their choices to infer their skill about the business process. This also allows skipping undesired details and proposing dialogue pathways that are more suitable for users and their operating context.

**Incomplete Knowledge About the Process.** The level of user’s knowledge about the business process might be unknown, nevertheless it is of paramount importance to focus the process exploration on the relevant activities, flows and data, also targeting the right level of detail. The conversational structure of chatbots and AI models behind them might enable to infer this information from the interactions with the users, thus improving the exploration experience.

**Managing Data and Process Complexity.** A human being is mainly unable to perceive and collect massive streaming data about processes and to create the best insights and actionable decisions. For example, considering the multiple configurations and product personalisation on the MyMuffin production process, how can users select the best solution with minimum cost? How do some data correlate to other data? In a digital factory based on digital twins [6], what is the cause-and-effect relating to the twin data differently from the real sensors data? According to the Human-In-the-Loop Data Analysis [8], experts must act as decision makers, while relying on chatbots for more repetitive and operative operations due to the data complexity and volume.

### 3 The iCHAT Framework

The iCHAT framework is our foundation for applying chatbots in digital factories. We first introduce the previous version of iCHAT, developed targeting educational chatbots (iChat-v1), then we illustrate the main extensions proposed in this paper (iCHAT-v2), that will be presented in detail in next sections.

#### 3.1 Background: iChat-v1

The goal of iCHAT is to create “flexible and adaptive” conversational interfaces, capable of supporting complex tasks and going beyond Question Answering paradigms. Designing a conversation application in iCHAT (besides a more or less standard interface component) is committed to three separate main engines

that are embedded inside a chatbot. These engines interact with each other to deliver an engaging conversation with the user via control tables. The engines are the *Conversation Engine*, the *Interpretation Engine*, and the *Transition Engine*, and all of them are table-driven, following a data-driven approach [3].

The *Conversation Engine* allows a user “to speak to” (“to be spoken by”) an application, taking into account the user’s profile to tailor the conversation. The role of the *Transition Engine* is to control the progress on a *pathway*, i.e., a sequence of content items for teaching. When a request is received, the Transition Engine takes care of the transitions across content items, i.e., determining which one is the most suitable “next item”. It is driven by a database (storing the content items) and tables describing the “pathways” (defined by the author of the contents) that allow traversing the content. A conversational interface interacts with the user through different devices (e.g., mobile phone, tablet, Alexa device) through a variety of media (e.g., text and videos).

The *Interpretation Engine* is responsible to maintain a number of parameters describing the dynamic situation of the user (from cognitive, emotional, psychological points of view) and, more importantly, to interpret them properly. Using the parameters, in fact, the engine should “interpret” the situation and instruct the Transition Engine about what to do at the next step. The Interpretation Engine is controlled via a set of rules stored in “interpretation tables”.

In addition, the *Interface Component* is responsible for technical jobs, like managing the various input/output devices, managing possible media conversions (e.g., text-to-speech, speech-to-text), transferring user turns to the conversation engine, transferring “content items” and chatbot turns to the user.

### 3.2 New Features Introduced in iCHAT for Digital Factories

In this paper, we propose a new architecture for iCHAT and a modified state transition machine for the chatbot, to develop it in the broader scope of processes in digital factories and tailor it to specific users who want to use a chatbot as an interaction interface to learn about a process. The overall architecture of this new version of iCHAT (iCHAT-v2) is shown in Fig. 1.

To re-design the Conversation Engine, it is crucial to separate chatbot’s dialogues in two domains: the generic domain and the specific application domain. The generic domain is responsible for the chitchat of the chatbot to peruse the conversation in a natural way. Besides, this chatbot should be domain-specific to deliver those conversations which are tailored for the specific purpose. For instance, if the chatbot is used in the production process, it needs to support a specific conversation on each component in the production process. These two domains are managed by the dialog manager component in the conversation engine through controlling tables and configuration data. The dialogue manager is a “decision-maker” in the conversation engine. It can take one of the following decisions: i) the chatbot has to say something; ii) the chatbot must wait for the user to say something; iii) the learning process should move on based on the state machine. In case “i” it passes control (and parameters) to the chatbot controller to continue the general conversation; in case “ii” it passes control to

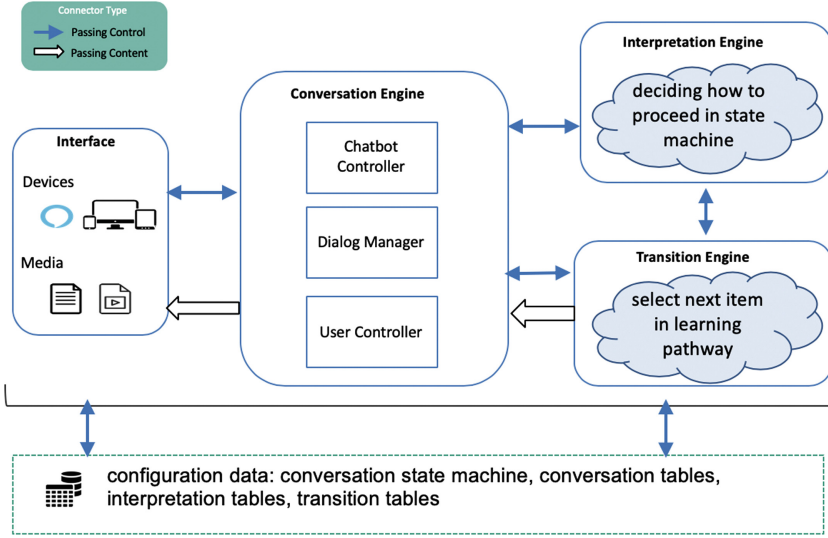


Fig. 1. The extended architecture of iCHAT.

the user controller to manage specific domain conversation; in case “iii” it passes control and parameters to the interpretation engine for any further actions on both state machine and learning path.

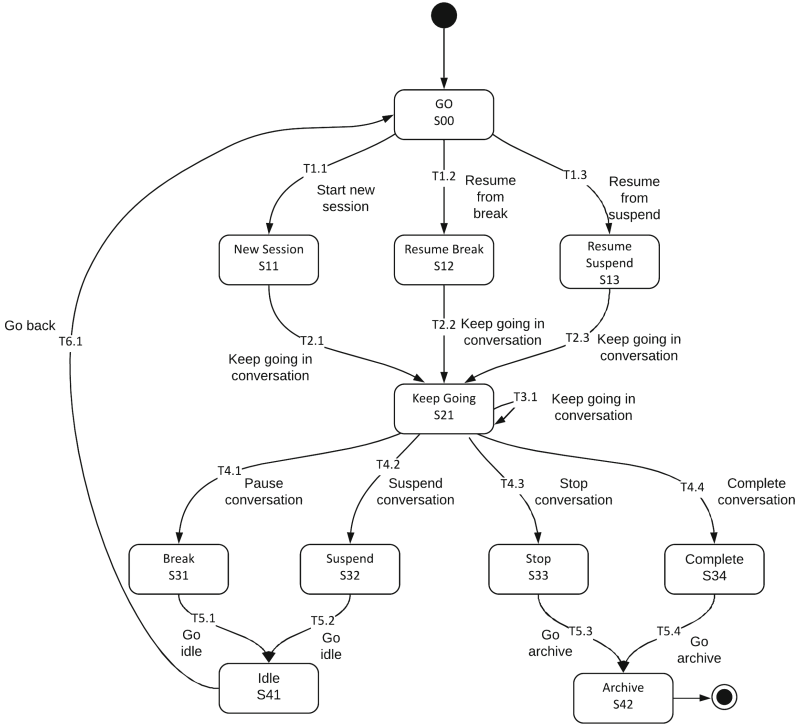
### 3.3 Generating Dialogues

Starting from the reference state diagram in iCHAT [3], we built a new state diagram in Fig. 2 to consider more situations in the conversation between a human and a chatbot. The state diagram allows controlling the utterances that the chatbot produces when reaching each state. This graph consists of a set of states and transitions between them. The transition from one state to the next depends on the user’s utterance. The state diagram allows controlling the utterances that the chatbot produces when reaching each transition or a state.

This finite state machine is defined as a tuple,  $(Q, \Sigma, \tau, q_0, F)$  consisting of a finite set of states  $Q$ , a finite set of input messages  $\Sigma$ , a transition function  $\tau : Q \times \Sigma \rightarrow Q$ , an initial state  $q_0 \in Q$ , and final states  $F \subseteq Q$ . The transition function here is a set of predefined data-driven rules in the conversation table (Fig. 3a<sup>1</sup>) that activate a message for the chatbot.

A conversation goes on as a sequence of “turns” by the chatbot and the user. According to the state machine in Fig. 2, a conversation can be *paused* (for a short break and later resumed), *suspended* (for a long break and later resumed), *stopped* (leaving the conversation), or *completed* (at the end of the pathway).

<sup>1</sup> ‘\*’ in message ID and ‘?’ before the category name indicate enquiring from the user. For readability, states and related transitions are separated by lines.



**Fig. 2.** The state transitions for the generic domain of conversation.

The number of consecutive turns and other chatbot’s behaviours<sup>2</sup> are controlled by tables.

The dialogue of the chatbot is designed by a set of categories. For each state in Fig. 2, different sequences of categories are associated. There are some dialogues associated to each state and transition. The dialogues are generated by considering different categories regarding the situation. The sentence(s) for each turn belongs to one out of seven different categories: ‘*greetings*, *support*, *preview* (of content), *summary* (of what has been done), *forecast* (of what is needed to complete the pathway), *action* (asking the user for taking action), and *reinforcement* messages.

For example, in state “S00”, the chatbot performs a dialog from “greetings” category; in transition “T3.1” the chatbot first says a “summary” message of what is done by user and a “forecast” message for what the user has to learn next. All information for generating the chatbot dialogues is derived from controlling tables.

<sup>2</sup> The behaviour can be customized on the basis of the user’s profile or context information. The chatbot, for example, could take several turns or a few ones; the chatbot could take long verbose turns, or could speak very briskly; the chatbot could use different wording styles (e.g., “professional”, “friendly”, “soft”).

In iCHAT-v2, an acceptable conversation to teach a production process, is the one leading the learner to complete a learning pathway in a proactive way. As it is illustrated in the following, beyond the state transition graph to manage a generic conversation, we introduce a new transition graph to handle the conversation on the different learning pathways defined to teach about processes. We develop a new structure for pathways beyond the sequential one originally used in iCHAT, to be able to deal with the complex structure of processes and, in addition, we develop conceptual models for transition graphs for teaching processes. The dialogue graph for this chatbot provides the structure for generating sentences, while the contents depends on the conversation tables. In Fig. 3b, we show part of a simulator for our chatbot to generate chatbot’s messages in the states and transitions that are defined in Fig. 3a. This is a sustainable solution to empower the instruction designers to build an adaptive learning process and adaptive learning conversation via chatbot and It is important to depict that the workers in the process do not need any additional background more than what is required for online learners to get trained by this chatbot.

Acronym	NAME	WHEN	CATEGORY	MSG ID	EXAMPLE
	GO	ENTER	GREETINGS	GR-01	Thank you for using DONYA, the learning chatbot
S00	GO	ENTER	SUPPORT	GR-01	If you have any problem interacting with the chatbot, please ask for support
T1.1		BEFORE	SUPPORT	SU-02	These are a few preliminary instructions <short-help>
T1.3		BEFORE	SUPPORT	SU-01	Do you need to refresh instructions for using the chatbot?
S11	FIRST TIME (FOR THIS PATHWAY)	ENTER	GREETINGS	GR-02	Welcome in this new exciting learning experience
S11	FIRST TIME (FOR THIS PATHWAY)	ENTER	PREVIEW	PR-01	You have selected the course <course.name> and the pathway <pathway.name>. The course is about <short course description>. The pathway <short pathway description>
S11	FIRST TIME (FOR THIS PATHWAY)	IN	FORECAST	FC-01	This is what you will accomplish <short pathway forecast>
S11	FIRST TIME (FOR THIS PATHWAY)	EXIT	ACTION	AC-01	Are you ready to start?

Variables - Iteration: 0  
Chat Panel - ITERATION: 0  
Variables - state: S00  
Chat Panel - C: <GR-01> Base Thank you for using DONYA, the learning chatbot  
Chat Panel - C: <SU-01> Base If you have any problem interacting with the chatbot, please ask for support  
Variables - Transition: T1.2  
Variables - ITERATION: 1  
Chat Panel - ITERATION: 1  
Variables - state: S12  
Chat Panel - C: <GR-02> Base Welcome back from your short pause  
Chat Panel - C: <PR-02> Base We remind you that you are in the pathway <so>.  
Chat Panel - C: <SH-01> Base This is what you have done so far <short-summary>  
Chat Panel - C: <FC-02> Base This is what you still have to do <short-forecast>  
Chat Panel - C: <AC-01> Base Are you ready to start?  
Chat Panel - U: No  
Variables - Transition: T2.2  
Chat Panel - C: <PR-04> Base We start with the first item of your learning path  
Variables - ITERATION: 2  
Chat Panel - ITERATION: 2  
Variables - state: S21  
Chat Panel - C: <SU-04> Base When you have completed the item or when you stop the content, we will resume our conversation  
Chat Panel - C: <SU-04> A1 We will resume after you have finished with the content.  
Chat Panel - C: <AC-01> Base What is your opinion about this content item?  
Chat Panel - U: No  
Chat Panel - C: <AC-02> Base What is your opinion about this learning experience(pathway)?  
Chat Panel - U: Yes  
Chat Panel - C: <AC-03> Base What is your opinion about this conversation?  
Chat Panel - U: Yes

(a) Conversation table

(b) Simulator for conversation tables

**Fig. 3.** Data-driven simulator for generating dialogues from tables for the chatbot

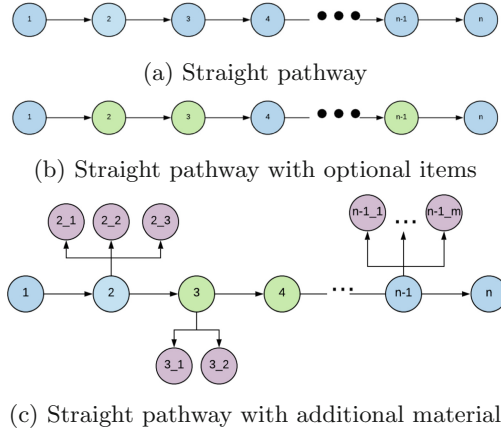
## 4 Chatbot as a General Process “Navigator”

A chatbot as a general process “navigator” is a finite state machine. It is driven by a graph, describing the current pathway. Given the “current position” it determines which is the most suitable next item for the learner. Nodes and arcs of the graph are enriched by colors and metadata to semantically recognize the characteristics of the items. Colors and metadata are basically used to “preview” what is coming or (more important) to help the user with “adaptivity in the small”, i.e., skipping unwanted items of the pathway, or looking for specific ones. For instance, in order to demonstrate the adaptivity in the small, nodes in the pathway are labeled by colors (blue for mandatory items in the path, green

for optional ones, and purple for additional materials). These content items are stored in table “Item DB” and organized in the learning pathways. Besides, users can select the pathway topology most suitable in a given context, which is called “adaptivity in the large”.

#### 4.1 Conversations for Teaching About a Process

To teach a process, three learning pathways structures have been defined:



**Fig. 4.** Pathway structures

- **Quick overview.** The learner can go through the overview items of the process. The goal is to improve the quality time of an employee to perceive the outline of the production process. In this regard, just items with proper metadata and tags are used to create a conversation on the highest abstraction of the process. The learning topology regarding to this pathway can be expressed as a straight pathway as shown in Fig. 4a. A straight pathway contains a list of items that have the nodes with “Overview” flavor type and the role is “Core” that is demonstrated by blue color in the graph. The user can go next through items to complete the learning path. Jumping back through items is not possible in this pathway. The Transition Engine keeps a list of items for each pathway in the straight pathway.
- **Full learning.** The full learning pathway shows more details about the process by considering all optional nodes for each activity. Figure 4b is a straight pathway with optional items and users have control to select optional items or skip them. All blue items in the pathway are coming from “Core” role and they are mandatory for the user to follow. The optional items are depicted with green node color and the Transition Engine informs the chatbot if the next item is optional and the chatbot gives the user a choice to select the

optional item or not during her learning path. If the user takes the items, the Transition Engine marks that item “visited” otherwise “skipped”.

- **Advanced learning.** Advanced learning pathways declare a straight pathway with core, optional and additional material items, which include all information (e.g., process, sub-processes, and variables related to each process). An example of this topology is shown in Fig. 4c. Additional material in item DB are tagged by “Recommended” and they are purple in the graph. If a user wants go deep on one item, this pathway structure is applicable. From the learning path viewpoint, these recommended materials are for an advanced learning experience. In a process, these additional materials can be inserted by the author which is designing the pathway to add information about the gateways, flow conditions, task data, and so on.

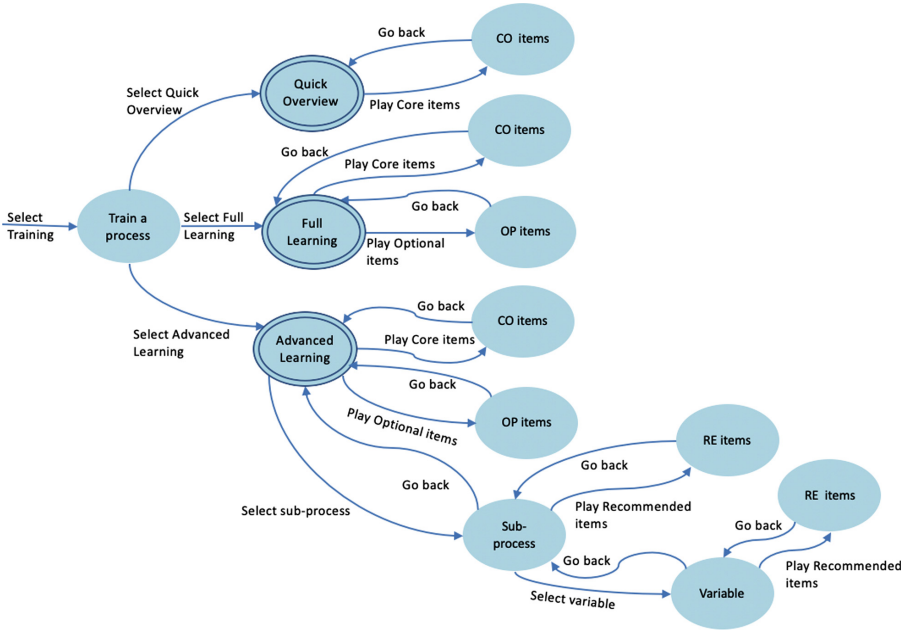


Fig. 5. State transitions for teaching on a process

## 4.2 State Transitions for Teaching a Process

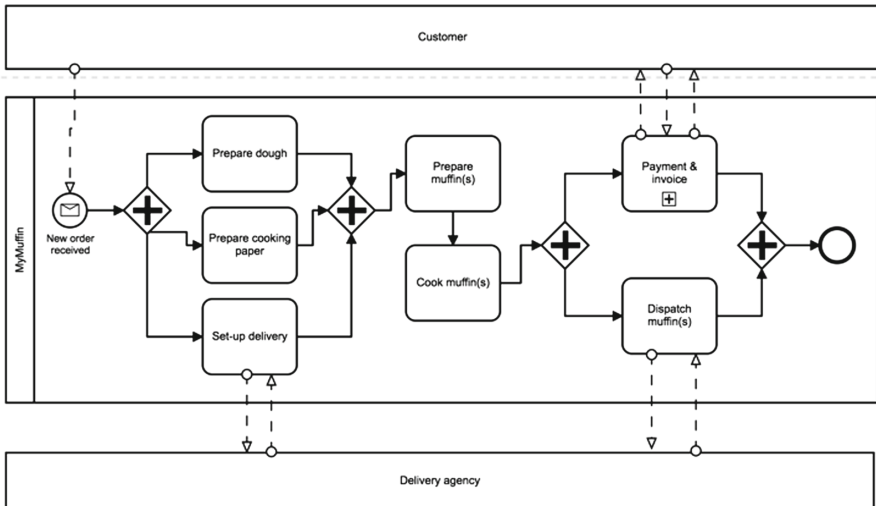
Figure 5 displays the state diagram for accessing an item in teaching a process. When a user selects process training, the chatbot goes in the initial “Train a process” state. From this state, it is possible to go deep in one of the possible topologies. As it is shown, items are tagged in three different roles (Core: CO, Optional: OP, and Recommended: RE). From the “Quick Overview” state, the

Transition Engine derives core items from “Item DB”. From the “Full Learning” state, going to core or optional items is achievable, based on the underlying structure of the pathway. From the “Advanced Learning” state, users can also go further to a component of the process to retrieve recommended information for available sub-processes and variables in a component.

## 5 The Chatbot in the MyMuffin Factory

It is important to consider two criteria for using iCHAT in your application scenario. Firstly, the content must be manageable by “pathways” to be followed of some complexity. Secondly, the user should accept a “constrained” conversation within the limit of the application domain.

Now we describe how we proceed by using a chatbot in a teaching a process in a factory, as the key contribution of this work, based on an example of a production factory first described in [2]. The overall production process in MyMuffin is shown in Fig. 6. The client orders box(es) (each one containing 4 muffins) online, by choosing among different possible variants, such as: (a) chocolate chips vs. blueberry vs. apricot bits vs. carrot bits vs. nothing as additional ingredient; (b) butter cream vs. hazelnut cream vs. icing sugar vs. nothing as topping; (c) yogurt vs. honey vs. nothing in the dough. The client can also customize the colors of the baking paper (wrapping the single muffin) as well as the colors of the box. The MyMuffin factory collects orders and organizes batches of muffin dough for production. As an example, if a client asks for 3 boxes of carrot



**Fig. 6.** The process at MyMuffin factory, BPMN diagram [2]

muffins with yogurt, icing sugar on top, pink baking paper, and another client for 2 boxes of carrot muffins with yogurt, nothing on top, yellow baking paper, the same dough can be used for both orders. Clearly, this scheduling service is based on the number of (and capacity of each) dough mixers, the stream of received orders, etc. The factory has a pool of dough mixers, of different capacity. The fact that the number of different combinations is finite guarantees that such a scheduling can be performed. When an order is received, in parallel to the dough preparation, the baking paper should be set up as well. In addition to prepare a set of the requested paper baking cases, a QR-code should be printed on each of them and used as a unique identifier of the specific order. The identification of the single muffin is crucial for customization. After the dough has been prepared, the muffins are placed in the baking paper cases and sent to the oven (connected to a QR code reader) for cooking. Muffins are cooked in batches of about 1,000 items and the length of this step is equal for all of them. After the baking has been performed, the cart is operated in order to route the different muffins to the right boxes, after putting the right topping, and then to the proper delivery station. Depending on the order, different delivery agents can be used.

The objective of introducing the chatbot here is to create a conversation to teach the process. As soon as a new employee arrives in the MyMuffin factory, the chatbot takes care of the initial orientation briefing about the company process components and all necessary information. Once the information regarding the initial training, operational knowledge, and other processes are handled by the chatbot, neither the employees and the trainer have to depend on each other. In the MyMuffin use case, the semantic tag related to each learning item is defined by four flavors (“Overview”, “Definition”, “Subdivision”, and “FAQ”) that are dependent to the specific process domain and three roles (“Core”, “Optional”, and “Recommended”) which they are general for any learning content related to process navigation.

The conversation is driven by tables governing how the chatbot speaks and how it understands the user’s utterances (Fig. 7). Below the tables a simple example of a conversation between chatbot and user is shown. As shown in Fig. 7, in state “S13” in the state machine, there are predefined categories for the chatbot turn when resuming a suspended conversation. Each sentence from the conversation table calls contents from “Item DB” table to create chatbot’s dialogues, selecting items for the categories specified for the “S13” to create a complete dialogue.

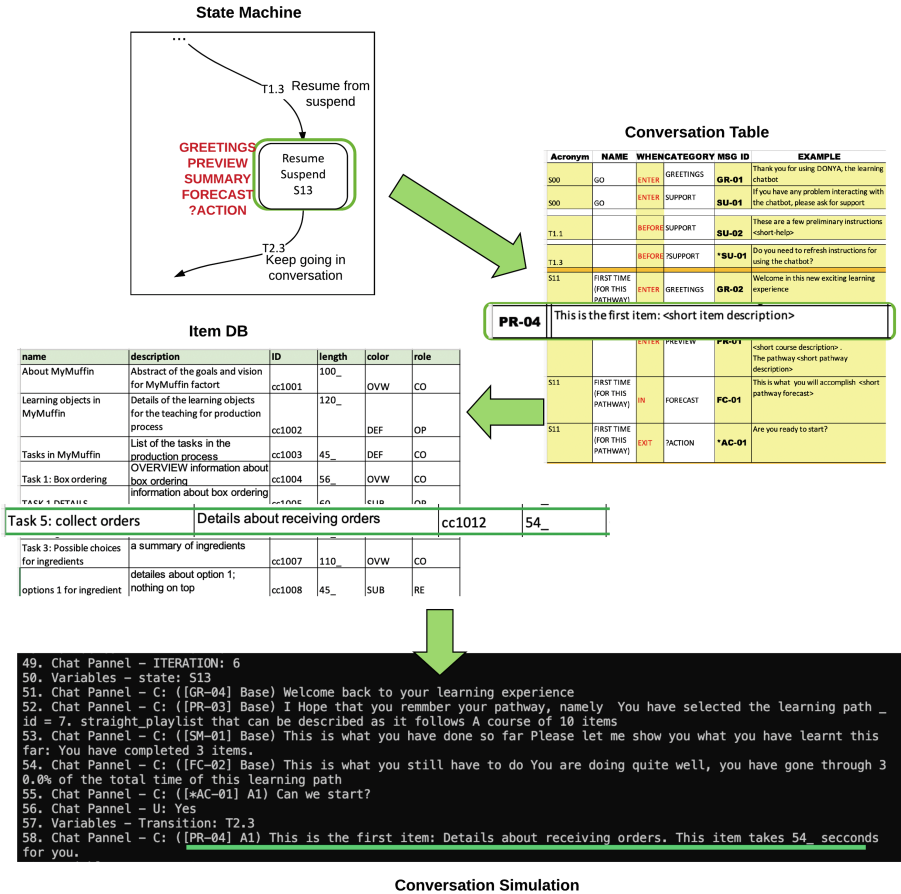


Fig. 7. Conversation simulation for the MyMuffin use case

## 6 Related Work

During the last years, some attempts were made to ease the creation of chatbots, aiming at querying data available in unstructured and structured formats.

Romero et al. [18] explored a set of application domains that can support human to supervise cyber-physical systems in digital factories. According to the authors in [16], these possible domains can be supported by using chatbots. They introduced many use case scenarios, where softbots can bring proactive insights over the production planners to optimize and support all operational demands. The authors mentioned that in smart supply network softbots provide monitoring (e.g., track and trace) to empower companies against any disruption in a supply network. In [7], a chatbot was developed to aid new hires through their onboarding process and reach related information needs as if it were a human assistant.

In [15], a chatbot was constructed on top of some open data. Here the first step is to extract plain text from documents stored as PDF files by employing an optical character recognition (OCR) software. At this point, a set of possible questions about the extracted contents were constructed using a “Overgenerating Transformations and Rankings” algorithm, which was implemented using the question generation framework presented in [9]. Finally, the matching patterns, essential to the chatbot’s answering capability, are defined through Artificial Intelligence Markup Language (AIML).

OntBot [1] employs a mapping technique to transform an ontology into a relational database and then uses that knowledge to construct answers. The main drawback of traditional chatbots, implemented for example through AIML language, is the fact that the knowledge base has to be constructed ad-hoc by handwriting thousands of possible responses. Instead of providing answers by looking for a matching one inside a database, OntBot retrieves information from the database, which will be then used to build up the response. Therefore, likewise our solution, OntBot does not need to handwrite all the knowledge base that stands behind the system.

In [10] authors declare that chatbots are the proper solution to provide personalized learning in MOOCs (Massive Open Online Courses). iMOOC [5] is a novel methodology for designing customizable MOOCs that brings adaptivity into the learning experience. iMOOC supports that various users of MOOCs may have different purposes, some learners may want to get personalized content, not learning the entire material. Here a chatbot is introduced to support the learner in choosing the most appropriate path. In the TEL (Technology-Enhanced Learning) literature, since several years Intelligent Pedagogical Agents (IPAs) are proposed as embodied intelligent agents designed for pedagogical purposes to support learning. Chatbots as those ones proposed in this paper are basically a form of IPA, and here we demonstrate their applicability to learning manufacturing processes.

The dialogue management component of the chatbot is one of the key parts in designing a conversational interface. In [14], dialogue management design and management in industry is discussed. A dialogue manager for a goal-oriented chatbot to conduct a proper conversation with user is illustrated in [11]. In [17], a model based on a Finite-State Turn Taking Machine is introduced in order to select an action any time.

Our aim is to propose a framework to query and teach processes. The interest in the employment of chatbots in the context of Business Process Management (BPM) is quite recent. Authors in [12] propose a way to take a business process flow as input and to produce a Watson conversation model as output. Differently from our approach, here authors focus on the execution of the process instead of teaching or monitoring it. In [4] an approach is proposed to use chatbots to learn business processes from input data. Here the focus is not on process participants. Instead the final users are data analysts in charge of mining business processes. In [13] the use of chatbots to help a process actor via conversation to perform tasks of a process model is presented.

## 7 Concluding Remarks and Future Work

In this work, we have introduced a framework to represent conversations to support instruction about manufacturing/production processes.

In future work, we will develop further the idea of connecting a conversation with a service-based process in a factory, in order to be able to control flexible process executions, with the chatbot as an interface for monitoring and controlling a running process in a digital factory using the same structures we presented for teaching as a basis for controlling conversations.

A robust validation in industrial settings is also foreseen, to measure the acceptability of the approach by workers and its effectiveness over current production processes.

**Acknowledgements.** The work of Donya Rooein has been supported by EIT Digital and IBM. The work of Francesco Leotta and Massimo Mecella has been supported by the EU H2020-RISE project FIRST. The work of Devis Bianchini has been supported by Smart4CPPS Lombardy Region project. This work expresses the opinions of the authors and not necessarily those of the funding agencies and companies.

## References

1. Al-Zubaide, H., Issa, A.A.: OntBot: ontology based chatbot. In: International Symposium on Innovations in Information and Communications Technology, pp. 7–12. IEEE (2011)
2. Bicocchi, N., Cabri, G., Mandreoli, F., Mecella, M.: Dynamic digital factories for agile supply chains: an architectural approach. *J. Ind. Inf. Integr.* **15**, 111–121 (2019)
3. Di Blas, N., Lodi, L., Paolini, P., Pernici, B., Renzi, F., Rooein, D.: A new approach to conversational applications. In: SEBD, Data Driven Chatbots (2019)
4. Burattin, A.: Integrated, ubiquitous and collaborative process mining with chat bots. In: 17th International Conference on Business Process Management, BPM Demo Track, Vienna, Austria, pp. 144–148. CEUR-WS (2019)
5. Casola, S., Di Blas, N., Paolini, P., Pelagatti, G.: Designing and delivering MOOCs that fit all sizes. In: Society for Information Technology & Teacher Education International Conference, pp. 110–117. (AACE) (2018)
6. Catarci, T., Firmani, D., Leotta, F., Mandreoli, F., Mecella, M., Sapio, F.: A conceptual architecture and model for smart manufacturing relying on service-based digital twins. In: IEEE International Conference on Web Services, ICWS, Milan, Italy, pp. 229–236 (2019)
7. Chandar, P., et al.: Leveraging conversational systems to assists new hires during onboarding. In: Bernhaupt, R., Dalvi, G., Joshi, A., Balkrishan, D.K., O’Neill, J., Winckler, M. (eds.) INTERACT 2017. LNCS, vol. 10514, pp. 381–391. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-67684-5\\_23](https://doi.org/10.1007/978-3-319-67684-5_23)
8. Doan, A.: Human-in-the-loop data analysis: a personal perspective. In: ACM International Workshop on Human-In-the-Loop Data Analysis (HILDA 2018), pp. 1–6 (2018)
9. Heilman, M., Smith, N.A.: Question generation via overgenerating transformations and ranking. Technical report, Carnegie Mellon University Pittsburgh PA Language Technologies Institute (2009)

10. Holotescu, C.: MOOCbuddy: a chatbot for personalized learning with MOOCs. In: Iftene, A., Vanderdonckt, J. (eds.) RoCHI-International Conference on Human-Computer Interaction, pp. 91–94, Bucarest (2016)
11. Ilievski, V., Musat, C., Hossmann, A., Baeriswyl, M.: Goal-oriented chatbot dialog management bootstrapping with transfer learning. arXiv preprint [arXiv:1802.00500](https://arxiv.org/abs/1802.00500) (2018)
12. Kalia, A.K., Telang, P.R., Xiao, J., Vukovic, M.: Quark: a methodology to transform people-driven processes to chatbot services. In: Maximilien, M., Vallecillo, A., Wang, J., Oriol, M. (eds.) ICSOC 2017. LNCS, vol. 10601, pp. 53–61. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-69035-3\\_4](https://doi.org/10.1007/978-3-319-69035-3_4)
13. López, A., Sànchez-Ferreres, J., Carmona, J., Padró, L.: From process models to chatbots. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 383–398. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-21290-2\\_24](https://doi.org/10.1007/978-3-030-21290-2_24)
14. Paek, T., Pieraccini, R.: Automating spoken dialogue management design using machine learning: an industry perspective. *Speech Commun.* **50**(8–9), 716–729 (2008)
15. Pichponreay, L., Kim, J.-H., Choi, C.-H., Lee, K.-H., Cho, W.-S.: Smart answering chatbot based on OCR and overgenerating transformations and ranking. In: 8th International Conference on Ubiquitous and Future Networks (ICUFN), pp. 1002–1005. IEEE (2016)
16. Rabelo, R.J., Romero, D., Zambiasi, S.P.: Softbots supporting the operator 4.0 at smart factory environments. In: Moon, I., Lee, G.M., Park, J., Kiritsis, D., von Cieminski, G. (eds.) APMS 2018. IAICT, vol. 536, pp. 456–464. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-99707-0\\_57](https://doi.org/10.1007/978-3-319-99707-0_57)
17. Raux, A., Eskenazi, M.: A finite-state turn-taking model for spoken dialog systems. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 629–637 (2009)
18. Romero, D., et al.: Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial revolution technologies. In: International Conference on Computers and Industrial Engineering (CIE46) (2016)