



aCHAT-WF: Generating conversational agents for teaching business process models

Donya Rooein¹ · Devis Bianchini² · Francesco Leotta³ · Massimo Mecella³ · Paolo Paolini¹ · Barbara Pernici¹

Received: 30 November 2020 / Revised: 18 August 2021 / Accepted: 6 September 2021 / Published online: 8 October 2021
© The Author(s) 2021

Abstract

This paper proposes a general approach for using conversational interfaces such as chatbots to offer adaptive learning of business processes in an environment involving different actors. Adaptivity concerns both the content being proposed, the sequence of learning items, and the way the conversation is conducted. The original approach allows the development of sustainable chatbots and empowers various non-technical actors (authors, teachers, publishers, and learners) to control the chatbot features directly. The aCHAT-WF framework (adaptive CHATbot for WorkFlows), proposed in this paper for managing conversational interfaces, conceptually represents all the aspects related to a conversation about business processes, with different facets for the user, the conversation flow, and the conversation contents, combining them to obtain a flexible interaction with the user. The paper focuses on the different preparation phases for instructional material based on Business Process Modeling Notation (BPMN) models, separating the different roles involved in the construction of a chatbot for teaching business processes and with the possibility of defining different styles for the interaction with the users. The proposed method is configuration-driven, to facilitate the separation of the different aspects of the control of the interaction and the delivery of contents.

Keywords Chatbot · Business process · BPMN · Digital transformation · Configuration driven · Educational conversational agent

Communicated by Selmin Nurcan and Pnina Soffer.

The work of D. Rooein is supported by EIT Digital and IBM. The work of Devis Bianchini has been supported by Smart4CPPS (Smart Solutions for Cyber-Physical Production Systems) Lombardy Region project (<https://www.smart4cpps.it/>). The work of F. Leotta and M. Mecella has been partly supported by the European Commission through the projects FIRST (virtual Factories Interoperation supporting buSiness innovaTion – #734599) and DESTINI (Smart Data Processing and Systems of Deep Insight – #857420)..

✉ Donya Rooein
donya.rooein@polimi.it

Devis Bianchini
devis.bianchini@unibs.it

Francesco Leotta
leotta@diag.uniroma1.it

Massimo Mecella
Mecella@diag.uniroma1.it

Paolo Paolini
paolo.paolini@polimi.it

Barbara Pernici
barbara.pernici@polimi.it

1 Introduction

Business process emerged in the last years as a powerful concept for improving the organizations performances and supporting digitalization. As a consequence, improving the design, implementation, and control of business processes is necessary for organizations across all industries. The topic of education in business processes has many aspects, which require effectively training employees to select and use specific methods, tools, and technologies. The overall challenge of teaching a complex multidisciplinary topic such as business processes is the variety of involved roles in an organization, and the need for essential requirements on all levels, including top executives, workers associations, and other stakeholders. Another challenge according to J. Moormann et al. is the spreading of knowledge about a business

¹ Politecnico di Milano, Milan, Italy

² Università di Brescia, Brescia, Italy

³ Sapienza Università di Roma, Rome, Italy

process within different levels of organizations [20]. In particular, with the widespread diffusion of digitalization in organizations, human actors act as decision makers relying on IT technologies to support their business processes. According to the Human-In-The-Loop paradigm [9], different roles in businesses and in modern factories (ranging from shop-floor operators to data analysts and business managers) may benefit from the acquisition of knowledge about the underlying production, logistic and business processes at different granularity levels. For an effective use of digitalization it is essential that the actors gather a good understanding of the business processes underlying their work.

From the education point of view, it is important to provide a comfortable experience to students who need to study remotely, alone, or in a blended situation. Conversational interfaces like chatbots have the potential to play an important role within the realm of technology-enhanced learning environments and can help generating a presence feeling which is not obtainable through traditional interactive interfaces. A conversational approach can facilitate the learning process. In fact, teaching a business process is a complex task and learners can get confused or bored through the learning experience.

Chatbots can provide a powerful tool to teach about processes in the form of flexible conversations, tailored to the roles of specific users who are interacting for monitoring the processes and taking decisions based on process data. Conversations based on chatbots may abstract from technological details that could hamper non IT people from improving their knowledge about business processes. Most of the current chatbots are designed for specific scenarios and without providing extensive configurations to give an adaptive conversation or handle a complex task. Although the idea of “educational chatbot” is not new, we believe that the direct involvement of chatbots in learning activities and specifically their application in the Business Process Management (BPM) field is rather limited, due to several factors, among which: (i) conversations are mainly perceived as “playful and entertaining”, not completely useful for complex tasks; (ii) the development and maintenance of the conversational technology for complex tasks requires IT specialists, high costs and a considerable amount of time, whereas teachers and process designers have problems at directly controlling the features of the chatbots; (iii) conversations are very often intended to satisfy requests by the user; this can be a part of a learning process, but only a marginal part. In addition, teaching business processes models requires to develop contents which have specific features and needs a specific consideration within a chatbot architecture.

This paper provides an original contribution in a few directions: (i) assigning a specific (original) role for conversational interfaces in teaching a business process; model; (ii) proposing an (original) effective architecture that makes

the development of educational chatbots for different actors in education a sustainable task; (iii) empowering process designers to almost directly shape and monitor the learning experience of a business process for learners; (iv) emphasizing the opportunity of adapting learning to the profiles and context of users who interact with the chatbots, and the role that chatbots can play for it. Specifically, we present an approach that takes a business process modeled with BPMN (Business Process Modeling Notation¹) as an input and generates learning items from the process description to feed the chatbot. Learners can be guided step-by-step through learning pathways to improve their knowledge about the process.

This work is an extension of the paper [28], where we firstly introduced the main features of a framework for creating educational chatbots for teaching business processes. In this paper, we illustrate the modeling process that allows supporting the different stakeholders in the production of educational material for teaching business processes, we detail the procedure for describing and automatically generating a conversation from a process description in BPMN, and we illustrate a running prototype for the system and its validation with a group of teachers and students.

The paper is structured as follows. In Sect. 2, we discuss the state of the art in using chatbots to teach business processes. In Sect. 3, we discuss the challenges and opportunities in educational chatbots for teaching business processes. In Sect. 4, we introduce the aCHAT-WF framework (adaptive CHATbot for WorkFlows), proposed in this paper for managing conversational interfaces for teaching business processes and in Sect. 5 we propose a model for modeling flexible and personalized conversations. In Sect. 6 we describe the model for learning contents creation in general, focusing in Sect. 7 on creating learning contents from a BPMN process. In Sect. 8, we introduce a running use case. Section 9 presents the results of a systematic evaluation of the framework with the running use case. Finally, Sect. 10 presents concluding remarks and possible future work.

2 Related work

The use of chatbots presents numerous advantages for organizations. Barakat et al. [2] investigated on key internal and external factors that contribute to the sustained use of conversational user interfaces. Among the mentioned factors, multi-stakeholder approach, continuous improvements, technological advances and compliance with regulations are the most related to this study. Moreover, a clear separation between different roles in a learning environment is a challenge also mentioned by work of S. Tegos et al. [29], in order to obtain a configurable chatbot. They presented a teacher-

¹ <https://www.omg.org/spec/BPMN/>

configurable architecture for a conversational agent with limited customization in collaborative activities in MOOCs².

During the last years, several attempts have been made to introduce chatbots in different contexts, aiming at querying data available in unstructured and structured formats. Here, we focus on the use of chatbots for learning purposes, with particular attention to their application in the Business Process Management (BPM) field.

Chatbots for learning purposes. In [22], 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 [11]. The matching patterns, that are essential to the chatbot answering capability, are defined through Artificial Intelligence Markup Language (AIML). The main drawback of traditional chatbots, implemented for example through the AIML language as mentioned above, is the fact that the knowledge base has to be constructed ad-hoc by handwriting thousands of possible responses.

OntBot [1] employs a mapping technique to transform an ontology into a relational database and then uses that knowledge to construct answers. 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 to answer questions.

In [13], the author declares that chatbots are the proper solution to provide personalized learning in MOOCs. iMOOC [5] is a novel methodology for designing customizable MOOCs, that brings adaptivity into the learning experience. According to this approach, various users of MOOCs may have different purposes, therefore 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.

Notably, in the technology-enhanced learning (TEL) literature, Intelligent Pedagogical Agents (IPAs) have been designed for pedagogical purposes to support learning. Chatbots like the ones proposed in this paper are basically a form of IPA, and here we demonstrate their applicability for teaching business processes.

Chatbots in industrial applications. The concept of digital factories is built on the idea of a decentralised production system, in which “human beings, machines and resources communicate with each other as naturally as in a social network” [12]. The introduction of chatbot-based services can be con-

ceived as an advanced family of factory-integrated services [10,30] to enable a new degree of control, surveillance, transparency, and efficiency. In particular, these services might contribute to change the role of human beings in the digital factory landscape, expecting employees to enjoy greater responsibility, to act as decision makers and to take on process supervision tasks (e.g., supplier selection, site selection, monitoring activities), according to the Human-In-the-Loop Data Analysis [9] paradigm, while the operational level (e.g., picking, loading) will be characterised by autonomously acting entities (i.e., the Cyber-Physical Systems). In this scenario, chatbots can be used for different purposes (e.g., monitoring, learning, supporting decision makers), focusing on different artifacts (e.g., business processes, documents) and targeting different categories of users (e.g., employees, data analysts, business managers).

Romero et al. [27] explored a set of application domains where humans can be supported to supervise cyber-physical systems in digital factories. According to the authors in [25], in these domains humans can be supported by using chatbots. They introduced many use case scenarios, where softbots bring proactive insights over the production planners to optimize and support all operational demands. The authors mentioned that in smart supply networks softbots provide monitoring (e.g., track and trace) to empower companies against any disruption. In [7], a chatbot was developed to aid new hires through their onboarding process and meets related information needs as if it were a human assistant. In [17], a chatbot has been integrated within a MES (Manufacturing Execution System) to provide technical assistance for shop floor operators, for predictive maintenance purposes. The paper [31] proposes the idea of chatbots as virtual assistants to help shop floor workers in learning assembly tasks in the manufacturing domain. This chatbot is under development and still the conversation is currently still very limited and needs enhancements to allow for more detailed questions and answers.

The dialogue management component of the chatbot is one of the key parts in designing a conversational interface. In [21], they discussed the requirements, specifications, and tuning that are typically used for dialogue management in the industry. They also highlighted the importance of a balance between automation and control if research is to effectively influence the design and practice of dialogue management in the industry. A dialogue manager for a goal-oriented chatbot to conduct a proper conversation with user is illustrated in [14].

Chatbots and BPMN processes. The aim of this paper is to propose a framework to teach processes. The interest in the employment of chatbots in the context of Business Process Management (BPM) is quite recent. Authors in [15] propose a way to take a business process flow as input and to produce a Watson conversation model as output. Differently from our

² Massive Open Online Courses

approach, here authors focus on the execution of the process instead of teaching. In [4], an approach is proposed to use chatbots to learn business processes from input data. Here, the focus is on data analysts in charge of mining business processes. In [16], the use of chatbots to help a process actor via conversation to perform tasks of a process model is presented.

Chatbots architecture and configurable design. Most of the chatbots have generic concepts to represent their architectures. In general, a chatbot architecture, according to [26], is composed of intent classification, entity recognition, candidate response generator, and response selector. Each of these modules can be developed by different approaches. The architectures of modern chatbots like Amazon Alexa and Apple's Siri use retrieval processes and machine learning to provide advanced information retrieval processes for dialogue generator and selector modules. In the present work, we propose an original approach, based on conceptual modeling based on state diagrams and relational data structures (i.e., tables), to separate the different aspects of a conversation, and the generation and management of the learning items from a business process model conceived by a process designer.

3 Adaptive learning technology: challenges and requirements

The introduction of chatbots for teaching a business process paves the way to many opportunities, such as the adaptation to the users' language, enabling iterative questions with incremental complexity and facing incomplete knowledge of users about the process. Nevertheless, to attain sustainability of such an incremental approach, new emerging challenges must be addressed, namely the implementation of a systematic methodology that supports the creation of learning pathways about processes, ensuring the separation of roles between technology and chatbot service providers on one hand, and process designers, content providers and conversation designers on the other hand.

Figure 1 illustrates the separation of roles in the aCHAT-WF platform. Tech and chatbot service providers are distinguished from content creators (e.g., process designers, conversation designers, authors) and teachers. Tech and chatbot service providers create the infrastructure to design and run chatbots. *Tech providers* integrate the AI powered technologies, providing Natural Language Processing (NLP) features and speech synthesis, while *chatbot service providers* create the infrastructure for user intent management. Contents creation is managed by *publishers* through their *contents creators*, which can be, in general, different from *educational institutions* dispensing the course with the assistance of *teachers*.

The availability of a chatbot creation technology as a mean of separation between content creators and tech providers, in order to relief content creators from the technicalities, requires to separate different concerns [24]. The technical chatbot infrastructure provided by aChat-WF acts as a chatbot creation technology, allowing to separate different concerns in the design of a lecture for teaching a business process based on instructional material provided by content creators. In particular, our approach allows focusing on three aspects, which are addressed in separate stages: the design of the business process itself, the design of conversational aspects suited for the different types of users and their learning preferences, and the provisioning of single learning items that can be used in creating learning paths at different levels of complexity.

Process designers know the modeling notation, the process, and its context and target users. Conversation designers work on the chatbot dialogues and on improving the quality of the conversation, focusing on the design of the conversational aspects, to control the way the chatbot interacts with its users. The goal is to create well-defined interaction styles depending on the audience. Authors and teachers focus on the contents of the learning items, identifying items that must be associated to the business process, and adapting them to the target users' profiles by combining them in different learning paths. The authors create the content material and basic learning paths, while teachers can personalize them to their specific teaching goals. Process designers, conversation designers, authors and teachers are empowered to directly control the main features of chatbot related to their job, without recurring to IT experts. The aCHAT-WF framework provides support for content creation from process models and also many adaptive features for controlling the conversation for learners. This separation of roles is designed to make the chatbot creation process adaptive and sustainable.

We discuss in the following the main goals and challenges considered to develop the approach that relies on the aChat-WF framework.

Common solutions for dealing with conversations. Introducing a chatbot creation technology allows creating a family of chatbots that behaves in a uniform way, with a common conversation strategy, regardless of the specific business process to be taught. The aChat-WF framework provides, in this sense, a common schema for preparing and tagging the content.

High level customization of the chatbot. The customization of the chatbot features normally is a technical level task and it is performed by IT persons. Separating different actors who are working with the chatbot and preparing a high level customization access to update the chatbot has the goal to empower them and reduce the fraction of the cost due to IT experts.

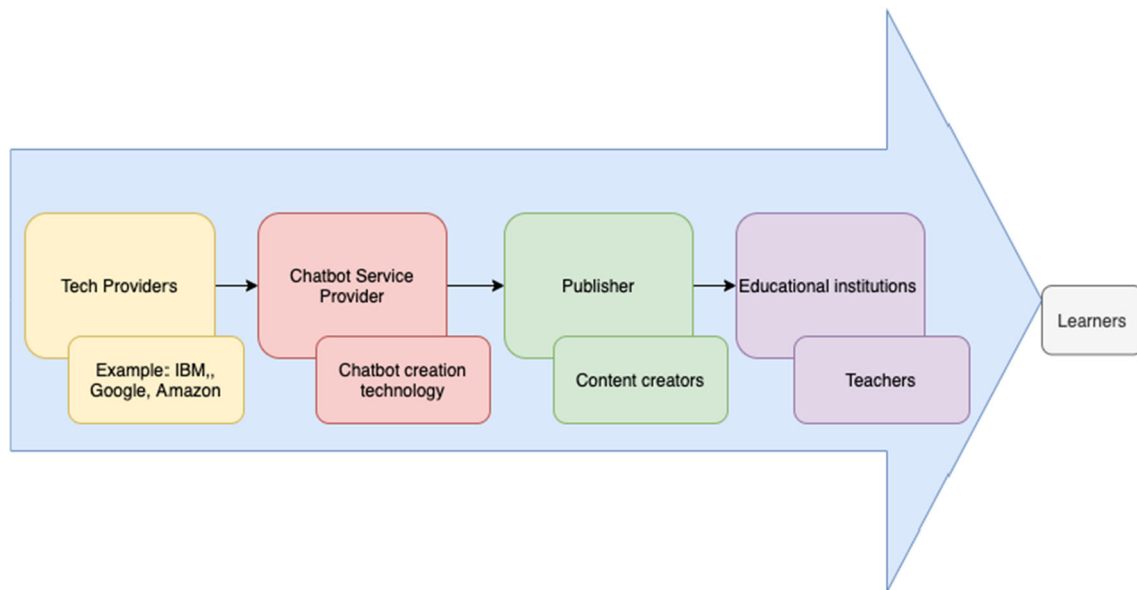


Fig. 1 Role of aCHAT-WF in the education domain

Adoption of the users' language. Formal process modeling notation might not be suitable or understandable for all the users who are interested in learning about the 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.

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 for them and relying on their choices to infer their skill about the business process.

Creation of adaptive conversations. Conversation features for adaptivity (style and wording) can be tuned to the specific needs concerning the user's profile and usage context. This also allows skipping undesired details and proposing dialogues that are more suitable for users and their operating context. Besides, an adaptive chatbot can control the number of turns and the length of the utterances.

Dealing with incomplete knowledge about the process. Companies document operational processes to study, execute and improve them. Several abstraction methodologies have been offered over past years for the process abstraction [23]. Chatbots, by generating conversations, provide exploration of different abstractions of a process model in teaching a business 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.

In the following, an original architecture is proposed, clarifying the differences between the various components of the chatbot: interface, conversation, interpretation, content and a generalized "meta-solution" for the conversations, controlled via state machines and several configuration data and tables. This approach is original and crucial for sustainability (i.e., lowering chatbot development and maintenance costs). We introduce a conceptual approach for content creation and for defining learning pathways from a BPMN model, due to the complexity of teaching processes models and content preparation.

4 The aCHAT-WF approach to chatbot design and implementation

In this section, we introduce the aCHAT-WF architecture (Sect. 4.1) and the main features to control the conversations (Sect. 4.2).

4.1 Chatbot architecture

The aCHAT-WF framework aims to create "flexible and adaptive" conversational interfaces, capable of supporting complex tasks and going beyond question/answering paradigms.

An original feature of the architecture is a clear separation among different concerns (see Fig. 2): interface, conversation, interpretation of the situation and content to be delivered through the chatbot. Different engines are embedded inside the architecture to deal separately with each one of these concepts. The engines are the *Conversation engine*, the *Action engine* and the *Interpretation engine*. The approach

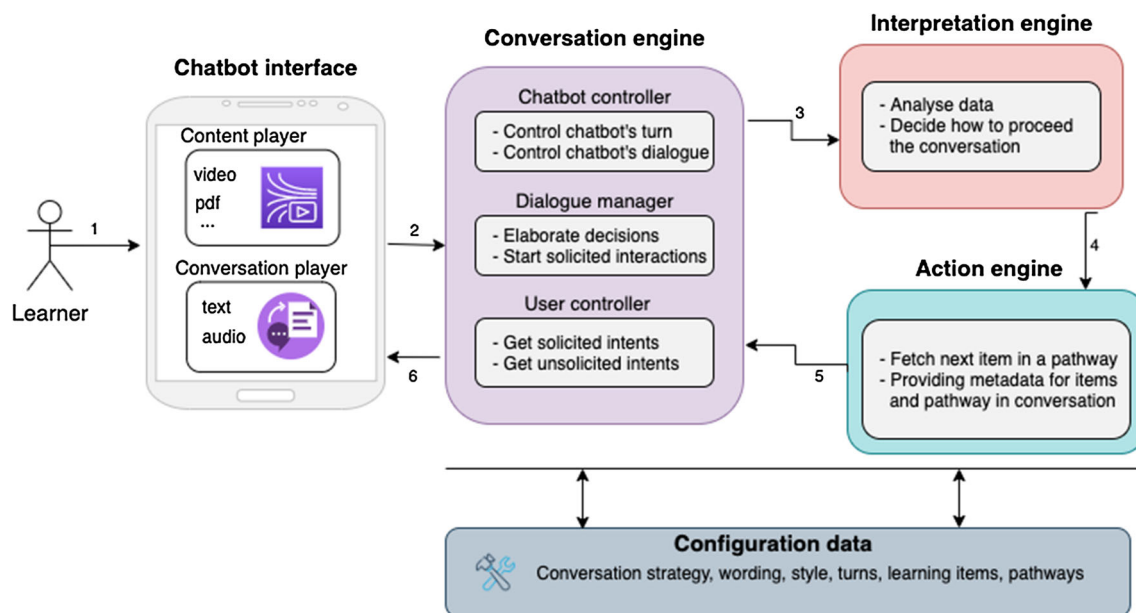


Fig. 2 The architecture for aCHAT-WF

at the basis of all the engines is table-driven, following a configuration-driven approach [28] to system development (*Configuration Data*). These engines interact with each other to deliver an engaging conversation.

The *Conversation engine* allows a user “to speak to” or “to be spoken by” an application, taking into account the user’s profile to tailor the conversation. It performs three tasks: it generates the conversation turns of the chatbot; it understands the user’s turns of conversation and it organizes the flow of turns. When it is necessary, the conversation engine takes instructions from the interpretation engine.

The role of the *Action engine* is to control the progress on a learning *pathway*, i.e., a sequence of learning items. When a request is received, the Action engine takes care of the transitions across the learning items, i.e., determining which one is the most suitable “next item”. It is driven by a database (storing the learning items) and tables describing the “pathways” (defined by the author of the contents) that allow traversing the content.

The *Interpretation engine* is responsible of maintaining a number of parameters describing the dynamic situation of the user from the 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 Action engine about what to do at the next step. The Interpretation engine is controlled via a set of rules stored in interpretation tables.

Finally, the *Chatbot Interface component* is responsible for UI tasks, such as managing the various input/output devices, managing the content delivery (e.g., showing a video or a pdf file related to a learning item), managing possi-

ble media conversions (e.g., text-to-speech, speech-to-text), transferring user’s turns to the Conversation engine, transferring learning items and chatbot turns to the user.

Starting from the proposal in [28], we developed a new modular architecture to achieve the decoupling between events and their handling. In aCHAT-WF, we designed a communication model between the main components to avoid asynchronous communications, which was the main drawback of the previous version of the framework. Therefore, the single components are synchronous and expose APIs in the whole architecture, thus reducing the queue management overhead.

4.2 Main features to control the conversation

To accomplish the educational activities, aCHAT-WF aims at satisfying the following requirements:

- allowing learners and teachers to connect by using their own devices;
- providing a general solution to manage several processes by separating content and conversation;
- adhering to the teaching approach, through a natural interaction with the chatbot (meant as a support tool to teach business processes).

The aim of educational chatbots is to deliver learning content intertwined with the conversation. In addition to separating contents production and conversation design, we aim to make available learning content reusable in different courses, tailored to different types of learners. In this way dif-

Table 1 Actors and their roles for aCHAT-WF

Actor	Roles
IT expert	- creates families of chatbots - modifies the state machine to control the chatbot behavioral model
Conversation designer	- designs the conversation strategy of the chatbot families - shapes the dialogue features of the chatbot - creates default conversation styles for chatbot
Publisher	- puts together different actors - provides requirements from the field (for BPM, specifically) - prepares the infrastructure of chatbot to the market - performs quality checking
Author	- creates the content (BPM specifically) - defines the base pathways - creates the specific chatbots (belonging to family)
Teacher	- coordinates the use of a chatbot to group of learners - customizes the content (optional) - creates the customized pathways (optional) in specific chatbots - adds their own wording style to specific chatbots (optional) - assigns pathways to a class of learners (and/or to subgroups) - defines default conversation features for a class of learners (and/or for subgroups)
Learner	- uses chatbot, as customized by teacher - is enable to create further customized pathways - customizes the conversation settings (change style, speed, talkativeness and loquacity)

ferent families of chatbots can be created, for teaching similar contents, and also different courses can be created, tailored to the needs of specific groups of learners. In the following, we focus on the main activities related to course design for teaching processes, starting from a general infrastructure that allows this separation of concerns

In Table 1, different actors are shown, with their roles in providing the technological infrastructure and creating a chatbot for teaching a business process with aCHAT-WF. The *IT expert* is responsible for creating a family of the chatbots. Each family of chatbots is based on a state machine, structuring the interaction in terms of states and in possible actions in each state and each transition. In the present paper, we discuss in Sect. 5.1 a state machine which provides a generic model for structuring interaction in a course in which learning material is presented, which provides a suitable generic structure for teaching business processes. The *Conversation designer* defines the conversation strategy of the chatbot, i.e., the way the chatbot interacts with the learners, based on the automatic recognition of the intents of the learners during the conversation. The conversation model is described in detail in Sect. 5 and its adaptation to teaching BPMN processes is presented in Sect. 8. The *publisher* puts different actors together to offer the chatbot service for teaching a business process, performing coordination and quality control activities.

Once the environment is set up, content providers (authors and teachers) can use it to create courses on specific business processes. First, after learning items are extracted from the process description, *authors* can associate to them additional learning material (e.g., multimedia contents). The quality of the content is completely separated from the conversation and the chatbot features. Authors, by creating adequate media, can improve the quality of the contents. In addition, they define the basic *learning pathways*, i.e., the description of the order of presentation of the learning material. A learning

pathway can be a simple playlist or a more complex topology, conceived as an organization of items. Different pathways can be created for the same material to provide different types of courses, as illustrated in Sect. 6.1 and detailed for business process teaching in Sect. 7. After authors, teachers receive the contents and may customize it. Therefore, the teachers' role is to refine the learning items and define a customized pathway for a personalized way of teaching. Teachers are also able to add their own wording for the chatbot dialogues. At the end, a *learner* selects one of the existing learning pathways and starts the conversation. Learners are able to customize the conversation settings (change style, speed, talkativeness and loquacity), thus adapting the conversation.

The chatbot configuration is completely separated from the content preparation. This separation is quite novel, as most of the chatbots, which we are aware of, merge conversation and content structures; this separation of concerns is conversely crucial for development, as the authors can concentrate on contents creation. Contents change much more often than the general conversation strategy.

5 Conversation modeling and generation

Designing the Conversation engine is crucial to separate the chatbot dialogues into two domains: the *generic* and the *specific* application domains. The generic domain is responsible for the chatbot chitchat to peruse the conversation naturally. Besides, the chatbot should be domain-specific to deliver learning items tailored to the specific process model. For instance, if the chatbot is used in a production process, it needs to support a particular conversation about its contents. These two domains are managed by the Dialog Manager component in the Conversation engine through different dialogue categories, which are defined to model a conversation. Later in this section, we will delve into details about these

categories, to demonstrate how a conversation is made by combining these parts.

In the Conversation engine, the *chatbot controller* is responsible for chatbot turn in the conversation and the *user controller* manages user's intents from both solicited and unsolicited interactions. 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, which provides the basic structure for the interaction. In the first case, it passes the control and parameters to the chatbot controller to continue the general conversation. In the second case, it passes control to the user controller to manage domain-specific conversation. In the latter case, it passes the control and parameters to the Interpretation engine for any further action in the state machine following the learning path.

5.1 State machine

A state machine models the conversation between a human and a chatbot, providing a basis for a flexible generic conversation during the learning process. This state machine allows controlling the utterances that the chatbot produces when reaching each state. The goal of this state machine is to be editable by an IT expert to customize the chatbot behavior and also a conversation designer to modify the sequence of the chatbot dialogues. Figure 3 shows the graphical representation of the state machine which provides the basic interaction structure used for teaching processes, consisting of a set of states and transitions between them. The transition from one state to the next depends on the user's utterance. A conversation goes on as a sequence of "turns" by the chatbot and the user. According to the state machine in Fig. 3, when a conversation starts (either starting a new conversation in state *presenting_lx* or continuing a previous conversation in state *resuming_lx*), it can be continued in *consuming_item*, *permission_for_item* (asking the user's permission to continue) and *getting_item*. Moreover, in the learning experience it is possible to have a long break and later resume the conversation (*asking_suspension* state), leave the conversation to stop the learning pathway (*asking_stopping* state), or reach the end of the pathway (*completed* state).

In order to give freedom to conversation designers to control the behavior of the chatbot, we designed two separate configuration structures for chatbot state machines. The first is the *implementation table*, used to define states and actions for the chatbot to implement possible complex behaviors. In addition, there is a *conversation table* for conversation designers, who can modify the conversation turns. Both structures are configuration-driven and will compile easily into executable code to control the conversation. In the follow-

ing, we will provide details about the implementation of the state machines.

The chatbot behaviour follows according to these configuration tables for the state machine (see Figs. 4 and 5). In the implementation table (Fig. 4), it possible to create and remove states, add or remove transitions, use events as inputs for firing transitions, solicit an interaction when entering a state. All these possibilities make the state machine capable of defining unlimited conversation shapes. We designed a simplified configuration table for designers to control output dialogues of the chatbot sentences at each transition. Defining new states, transitions and conditions to fire transitions (all the black elements in Fig. 3) are fixed by IT expert and cannot be changed by designers at this level. The chatbot output interactions like dialogues and questions (blue elements in Fig. 3) can be modified as the conversation designer wants.

Each state in Fig. 3 consists of one or more sub-states that are not included in this simplified graphical representation. The list of the sub-states is presented in Fig. 4. A state in the implementation table is defined by:

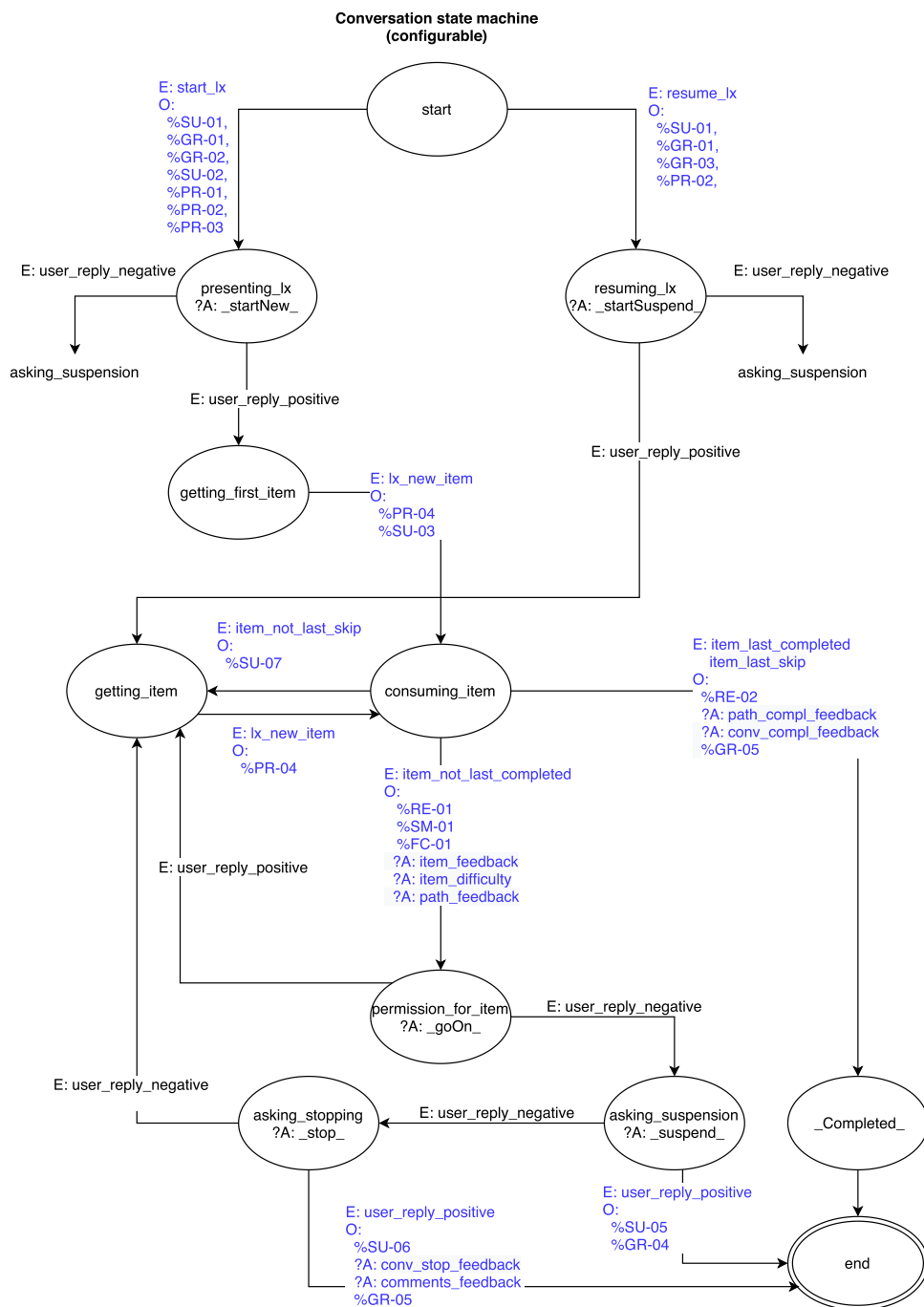
- *id*: an identifier that must start with 'S' and only contains digits (0-9) or capital letters (A-Z);
- *state_name*: an optional longer name for describing the state, with a maximum length of 20 characters;
- *exceptional*: a Boolean variable to denote whether the state is exceptional or not (default value is *false*);
- *solicit*: an optional solicited interaction to start when reaching the state.

A transition is defined by:

- *transition_event*: the event that fires the transition (optional); these predefined events are useful to model conversation in teaching situations; they are in a limited number and defined also in a table;
- *transition_id*: an identifier that must start with 'T' and only contains digits (0–9), capital letters (A–Z) or dots (.);
- *to_state*: the destination state of the transition, it must be a valid state defined in the configuration table.

Exceptional states are special states that keep track of the previous state. When being in an exceptional state, only some events are allowed: if the first detected event is allowed, then the corresponding transition is fired. Otherwise, if the event is not allowed, the system rolls back to the last non-exceptional state, as nothing happened. This is particularly useful for dealing with the unsolicited request of stop or suspension, that, if confirmed, would change the state machine branch but, if not approved, should go back to the conversation. Every non-ending state must have at least a transition attached. Only one ending state is allowed. The organiza-

Fig. 3 The graphical representation for state diagram in aCHAT-WF



tion of the chatbot dialogues is defined by the conversation table (see Fig. 5). For each transition in the state machine, a conversation designer can attach one or more dialogues.

5.2 Modeling conversations

Modeling the conversation is still a dilemma for building chatbots and their conversational abilities despite recent voice recognition and natural language process features. For

example, making the dialogue patterns to create a desirable conversation between human and chatbot raises some efforts for conversation designers. On the other hand, for more than five decades there are studies on modeling a conversation that are ongoing on the structure of the conversation, the sequential mechanism and the conversation technology [18]. Moore has presented a new approach to implement the multi-turn dialogues known as the Natural Conversation Framework (NCF), that provides a library of the generic conversation

Fig. 4 Implementation table for controlling the state machine

state_id	state_name	exceptional	solicit	transition_event	transition_id	to_state
S00	start			conv_start_lx	T0.0	S02
				conv_resume_lx	T0.1	S03
S99	end					
S02	presenting_lx		_startNew_	user_reply_positive	T2.0	S01
S01	getting_first_item			user_reply_negative	T2.1	S09
				lx_new_item	T1.0	S04
S03	resuming_lx		_startSuspend_	user_reply_positive	T3.0	S15
				user_reply_negative	T3.1	S09
S04	consuming_item			item_not_last_completed	T4.0	S05
				item_last_completed	T4.1	S13
				item_not_last_skip	T4.2	S15
				item_last_skip	T4.3	S13
S05	item_feedback		_likeItem_	user_reply	T5.0	S06
S06	item_difficulty		_difficultItem_	user_reply	T6.0	S07
S07	path_feedback		_interestPath_	user_reply	T7.0	S08

transition_id	throw_lx_event	output_sentence
T0.0		%SU-01
		%GR-01
		%GR-02
		%SU-02
		%PR-01
T0.1		%PR-02
		%PR-03
		%SU-01
		%GR-01
		%GR-03
T2.0	lx_fetch_item	%PR-02
T2.1		
T1.0	lx_consuming_first	%PR-04
		%SU-03
T3.0	lx_fetch_item	
T3.1		
T4.0		%RE-01
		%SM-01
T4.1		%RE-02
T4.2	lx_fetch_item	%SU-07
T4.3		%RE-02
T5.0		
T6.0		
T7.0		

Fig. 5 Conversation table (used by designers to customize the state machine)

UX patterns by following the natural human conversation patterns [19].

In this paper, a conversation model for a teaching activity has been designed. Through this conversation model and the configuration-driven structure of the aCHAT-WF design, a conversation designer can create and configure a conversation to get many various conversation experiences for the users. This model has been conceptualized in Fig. 6. According to this model, each conversation starts from a state and, based on each state, there are a set of actions (it can be zero action or many actions). It is possible to move from one state to another through a transition. Both of the actions or transitions begin a solicited dialogue turn by the chatbot. In the solicited turn, the chatbot solicits the user and then waits for the user’s reply to continue the conversation. Chatbot inter-

prets the user’s message and response with an immediate reply to the user and throws the corresponding event to transit to the next state. In addition, at any state of the conversation, the chatbot listens to the user for any upcoming unsolicited turns. In this case, the chatbot, after understanding the intent, provides an immediate reply to the user, changes the behavior accordingly, if it is requested by the user, and updates the state of the conversation.

5.3 Personalizing conversations

Besides modeling the conversation, to make it configurable for different actors, we need a separated design for conversation designers and IT experts to design or modify the state machine. These separated designs have been implemented by configuration tables.

In Fig. 3, the output messages attached to transitions can be distinguished in two types. The first type collects messages whose name starts with a “%” symbol, followed by a string that corresponds to one of the possible message categories like greeting (GR), support (SU), preview (PR), reinforcement (RE), forecast (FC) and summary (SM). The other types of output messages attached to the transitions are those whose name starts with the “?” symbol. These messages solicit the user to take an action and later are translated by the Interpretation engine to decide on the next step. All these messages are controlled by the conversation table (Fig. 5) handled by a conversation designer to modify the state machine and instantiate different versions of the chatbot. Both of these messages are used to model the conversation of the chatbot and the user and each of them is a conversation activity as part of the conversation modeling for tutoring. This is an abstract model of the general pattern for modeling the tutoring as inspired by the natural conversation. Conversation tables are designed to support conversation designers about creating different conversation patterns for teaching.

conversation model.pdf

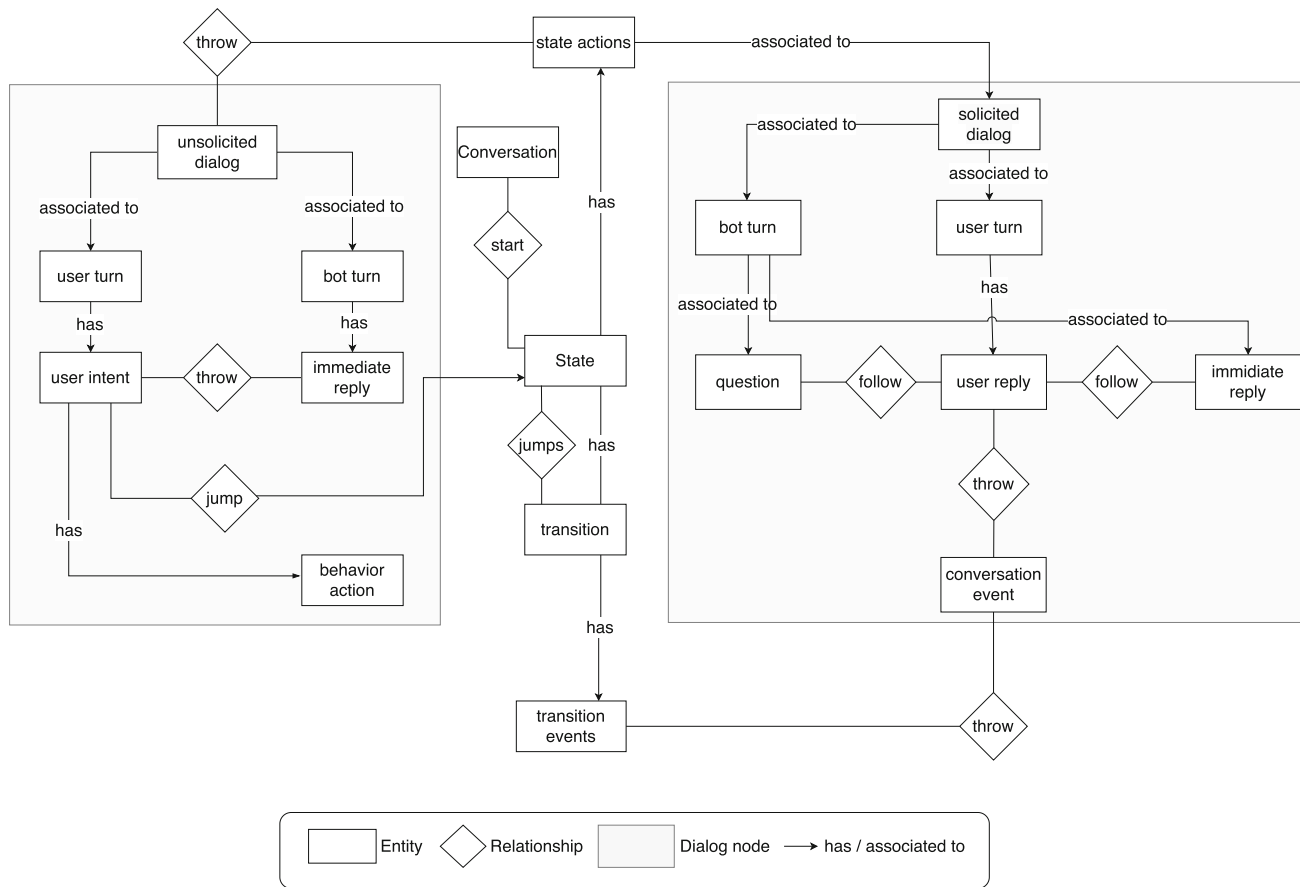


Fig. 6 Conceptual model for conversation modelling

Table 2 Schema for configuration data for chatbot dialogues

Message id	style 1 (empathic)	style 2 (professional)
%GR-01	Hi buddy! Hi < \$T_USER> ! :) ...	wording 1 wording 2 ...
%PR-01	wording 1 wording 2 ...	wording 1 wording 2 ...
%SU-01	wording 1 wording 2 ...	< \$V_BOT-NAME> is the name of the chatbot that will accompany you in this learning experience. You are using < \$V_BOT-NAME> , the innovative chatbot to support your learning experience. ...
...	wording 1 wording 2 ...	wording 1 wording 2 ...

5.4 Dynamic conversation settings

The number of consecutive turns and other chatbot behaviours (having a different style of the conversation or being talkative or taciturn) are controlled by conversation settings. These behaviours 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”). These flexible aspects for the conversation with the chatbot are also designed and implemented via configuration tables. A conversation designer can easily update the wording of the conversation table and also add new styles for her own chatbot. Here in Table 2, the general schema for bot sentences in configuration data is shown. Some utterances are self-contained (e.g., “Hi buddy!”); other utterances require the use of templates and variables (e.g., “You have selected \$T_USER”). Variables and templates are derived from the table. These features assist the chatbot to be able

to talk in diverse ways and customize it to the user's variables according to the profile of the user. For example, if the learner prefers a friendly conversation rather than a dry one, the chatbot marks this in the user's profile and changes it later according to the user's request. It is also extended to the other preferences, like length of the chatbot sentences or number of the turns.

In next section, we will delve into the content preparation and we introduce an automated approach for generating learning content from a business process model that is integrated with our chatbot.

6 Learning contents and pathways

A *learning item* is the smallest part of a complex body of contents. As mentioned above, learning items can be of different types and are defined by the authors and new items can be added by the teachers. In Sect. 7, the contents creation in terms of learning items for a BPMN process is illustrated in detail. A *learning pathway* is a collection of learning items, properly arranged in a topology. Multiple pathways can be created from the same learning material, with different topologies (see Sect. 6.1).

In addition to the topology structure, a pathway has a traversing algorithm to move across it. Pathways can be obvious in teaching a book or a course, like "chapter 1", "chapter 2" and so on, in sequence. Pathways can also be related to what is required in teaching that content, e.g., "core", "recommended", "optional" items. Another criteria could be the level of difficulty. Criteria can be combined, creating for example different pathways: (a) chapter 1 and chapter 2 - core and recommended items - "elementary" and "basic" items; (b) chapter 1 and chapter 2- optional items - excluding "advanced" items. Traversing of a pathway can be "adapted" to the user's profile, specific or temporary needs and conditions. Additional metadata can be used to define the characteristics of a learning item depending to the subject. In teaching a business process each learning item can be any type like "overview", "definition", "FAQ" and so on. These additional metadata are defined directly by the owner of the content (authors of the process model).

A learning session for learners starts by selecting a pathway. Once the pathway has been selected, it cannot be modified within the conversation, but additional adaptation is possible. An example could be skipping "elementary" items. Another possible adaptation is to select or to exclude nodes of a given color. For example, teachers can create pathways including items with color 'overview' or 'FAQ'.

```

filter_include = [
  {
    metadata1 = values, # and
    metadata2 = values # and
  }
  , # or
  {
    metadata1 = values, # and
    metadata2 = values # and
  }
]

filter_make_skippable = [
  {
    metadata1 = values, # and
    metadata2 = values # and
  }
  , # or
  {
    metadata1 = values, # and
    metadata2 = values # and
  }
]

```

Fig. 7 Data structure for defining filters with included and skippable items for pathway creation

6.1 Topology of a learning pathway

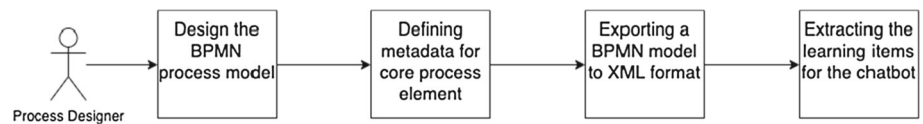
A learning pathway can be created by manual selection of learning items or by using a predefined topology. A topology is a definition of a structure that can be used to compose items of a course into a learning pathway. The structure may be defined by including items of certain metadata in the pathway and by choosing which items are mandatory and which ones are optional and can be skipped. The simplest topology is a linear playlist of mandatory items and the simplest algorithm to traverse the pathway may include only next/previous actions. A more complex topology, traverses the learning items with more actions (e.g. user asks for next 'overview' or 'FAQ' item in the pathway).

The topology can be implemented as multiple filters that select only items having certain properties. Two groups of filters are needed: one will select which items to include in the pathway and the other will choose which of them are optional. The filters can be represented as shown in Fig. 7. The metadata inside the filter components can be considered as conditions connected by a logical AND, while filter components are in logical OR with each other. For the first group, only items for which the resulting formula is true will be inserted in the pathway. In the same way, the second group will select which of the included items can be skipped. In Sect. 7.2 we illustrate in detail a set of topologies for pathways for teaching BPMN processes.

7 Generating learning contents from BPMN models

Content creation from a BPMN model includes several steps. Figure 8 provides a description of the steps to be performed by a *process designer*, who plays here the role of author of the content (business process), according to Table 1. First, the *process designer* prepares the BPMN model to describe the process. Then, as a mandatory step, s/he needs to define the learning items.

Fig. 8 The high level stages for content creation for the chatbot



7.1 BPMN learning items

In order to define a learning item from a process model, we defined a set of guidelines to help the process designer³. A simple way to extract the learning items from a process model description is to consider each element of the process model as a single learning item.

On the other end, nowadays, business process modeling tools are much more than simple diagrammatic tools for business processes. It is possible to visualize and document business processes to develop better organizational understanding. Some tools give the user the opportunity to take business process modeling to another level with additional features to add more metadata to a process. There are many platforms for designing and editing a process like Signavio⁴, CAMUNDA⁵ and Eclipse BPMN2 Modeler⁶. The Signavio Process Manager facilitates adding of metadata to business process models. Diagrams are created in the Signavio Process Manager using BPMN 2.0 modeling language and they can be exported, including additional metadata, according to an XML format. In this paper, for designing a BPMN model and crafting the teaching content, we are using the Signavio platform.

For translating a business process into actionable learning contents, we consider the following key requirements: (i) completeness, i.e., considering processes with arbitrary topologies; (ii) automation, to produce the teaching content for the chatbot automatically and without human intervention. The output generated by this translation is readable for the chatbot to create the executable learning items needed to teach a process.

There are several possible strategies for extracting a single learning item from a process model. The fragmentation of contents into small items is an essential component for adaptive learning, since it makes it possible to create learning pathways oriented to specific needs. The simplest strategy is to consider each element in the process model as a single learning item. This strategy makes the automated extraction easier; however, it probably makes each learning item too short.

³ The guidelines are available at the URL <https://github.com/donyaaaaam/aCHAT-WF>.

⁴ <https://www.signavio.com/bpmn-2-0-for-efficient-process-design/>

⁵ <https://camunda.com/bpmn/tool/>

⁶ <https://www.eclipse.org/bpmn2-modeler/>

In aCHAT-WF, in order to extract the initial learning items directly from process models, we start from core elements as they are defined in Signavio for BPMN models: tasks, pools/lanes, data objects, start and end events, etc. For each core element, we associate in Signavio a set of metadata as shown in Table 3, that are the basis of the composition of learning items in learning pathways as described in the next section. These metadata define the properties for each element with the process as follows: *Level* defines the level of difficulty of a learning item for learners; *Color* refers to concepts associated to learning content; *Role* of learning item in teaching in terms of its importance for the learning task. Due to the nature of the process, the ordering of the learning items to teach a process is important. For each process, the default ordering comes from drawing the elements during the design process. The aCHAT-WF framework automatically parses the BPMN XML file extracted from the process editor to extract the learning contents from the BPMN model, annotated with the requested metadata for creating learning items. A learning item consists of text data for the chatbot to play in the conversation and visual content like video, pdf, or any external link for further information about that item to be shown in the content panel. This media is provided by the author or teacher separately from the automatic generated text from the BPMN model.

7.2 Predefined pathways for teaching a process model

Teaching a BPMN process model entails providing useful description data for learning items to create meaningful pathways. Consequently, we define four pathway topologies and their properties to be used in our chatbot:

- **Full learning:** this pathway includes all the items of the course; the chatbot does not ask for permission to the user to play the items and all the items are mandatory for the learner; clearly, using the adaptive features, a user can ask for skipping an item during the learning path;
- **Overview learning:** the pathway with “core”⁷ items; the chatbot filters just the “core” items and does not ask for permission to play the content;
- **Regular learning:** this pathway contains items with “core” and “optional” items; the chatbot does not ask

⁷ The term “core” here refers to role of learning item in the pathway not the type of the process element in BPMN.

Table 3 The list of the metadata for teaching a process model

Attribute Name	Attribute Description	Attribute Values
Level	The level of a learning item for learners	[elementary, basic, advanced]
Color	The color defines concepts of learning content	[red (overview), green (definition), yellow (FAQ)]
Role	The role of learning item in teaching	[core, optional, recommended]
Order	The position of an item in the learning pathway	[0, 1, ..., N]. N: the maximum number of learning items
Element_Type	The type of element in the process model	[Tasks, Events, ..., e]. $e \in [core\ process\ elements]$

for permission regarding the “core” items, but an user is requested to confirm the “optional” items;

- **Advanced learning:** the pathway has items with “core” and “recommended” items; the chatbot asks for permission for playing “core” items, which are not mandatory ; in addition, the chatbot does not ask user’s confirmation about the “recommended” items, as they can be considered mandatory for this learning experience.

A suitable pathway allows further selection according to learner’s requests or perceived problems, and adapting the conversation features to the learners’ preferences and to contextual situations (e.g., running out of time). Using metadata in order to arrange a proper pathway gives learners a better adaptivity in education. In addition to the general topologies that can be applied to most teaching domains (e.g., also in a STEM course), we can consider specific pathways considering the “Element_Type” metadata related to business models. This enriches the filters to be more process-aware. For example, in a predefined pathway, a student can learn all “Task” elements as mandatory learning items and skip all the “Events”.

As introduced before, pathway creation is separated from the content creation. We designed a tool for authors and teachers to define their filters, as shown in Fig. 7 and to create a pathway. Teachers can also modify the existing pathways (adding more learning items or reordering the position of the items) and create their own pathways. When learners select a course, they can activate one of the existing pathways for their learning experience and start the chatbot on that pathway.

8 Teaching a process in the MyMuffin factory

In this section, we illustrate an example of teaching a process in a production factory, starting from a process firstly described in [3]. The overall production process in MyMuffin is shown in Fig. 9. The client orders online box(es) (each one containing 4 muffins), by choosing among different possible variants, such as: (a) chocolate bits 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 muffins with yogurt, icing sugar on top, pink baking paper, while another client asks 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 and capacity of 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 preparing 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.

Considering the BPMN process model from MyMuffin factory, the learning items are automatically extracted from the BPMN XML description, as described in Sect. 7. We created a course named “Business Process for MyMuffin factory” based on extracted learning items, for a total time of 38 minutes and 56 seconds of video, including some metadata like name and description of a learning item. As we mentioned earlier in Sect. 7, the media for a learning item are just author and teacher’s concern. In the MyMuffin use case these are videos, and all the videos for this course were generated from image-slides coupled with audio generated from text provided by conversation designer as metadata during design of the process model by using Watson text-to-speech

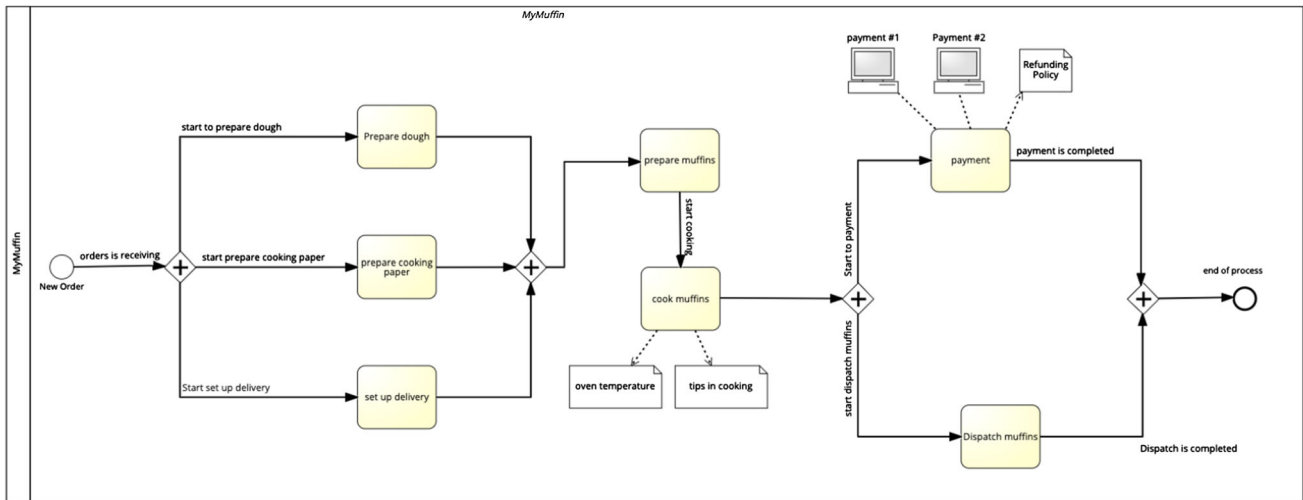


Fig. 9 The process at MyMuffin factory [3], BPMN diagram created in Signavio

intent	description	wording-1	wording-2	wording-3	wording-4	wording-5
#command	Command (special intent)	_command_	_solicited_	_unsolicited_		
#ans-01	Very negative answer	absolutely not	not at all	very bad	terrible	terribly
#ans-02	Negative answer	no	nope	nop	nah	n
#ans-03	Neutral answer	so so	so and so	I don't know	idk	maybe
#ans-04	Positive answer	yes	sure	yep	yeah	y
#ans-05	Very positive answer	very well	great	amazingly	super	very good
#uGR	Unsol. greetings	hello	hi	hola	how are you	how was your day
#uSU-01	Unsol. support	Can you help me?	can you do me a favor	I need help	do this for me	help me with this
#uPR	Unsol. preview	What's the next item?	I would like a preview	I want a preview	preview	preview please
#uSM	Unsol. summary	Summary, please	i don't remember what we talked	please, can you briefly recap	recap, please	remind be about my path
#uFC	Unsol. forecast	How much do I still have to do?	E.T.A. please	eta ?	forecast	how long it will take for this

Fig. 10 Configuration table for user's intents (the first line is occupied by the #command)

technology. The essential content metadata for conversation is extracted automatically from the process description; on the other hand, the course designer adds multimedia related to each learning item. This approach has several advantages: (i) it lowers the costs of production, (ii) it reduces cost and time for maintenance, since modifying text is much simpler than editing video, and (iii) it allows additional delivery media (pdf files for text and images) that can also be printed.

The conversation designer adjusts the chatbot dialogues and sentences to set up the chatbot for the MyMuffin use

case. Then, to enhance the conversation engine and expand the user's understanding for the chatbot, the conversation designer adds a set of domain dependent intents to the general intents. These intents are in the configuration data for each chatbot and they can be updated easily before the chatbot execution. The table of intents is a specific table containing only the possible words or sentences from the user. The table structure includes an intent name, a description, and several columns for the different wordings, as shown in Fig. 10. Not all the intents need to contain the same amount of

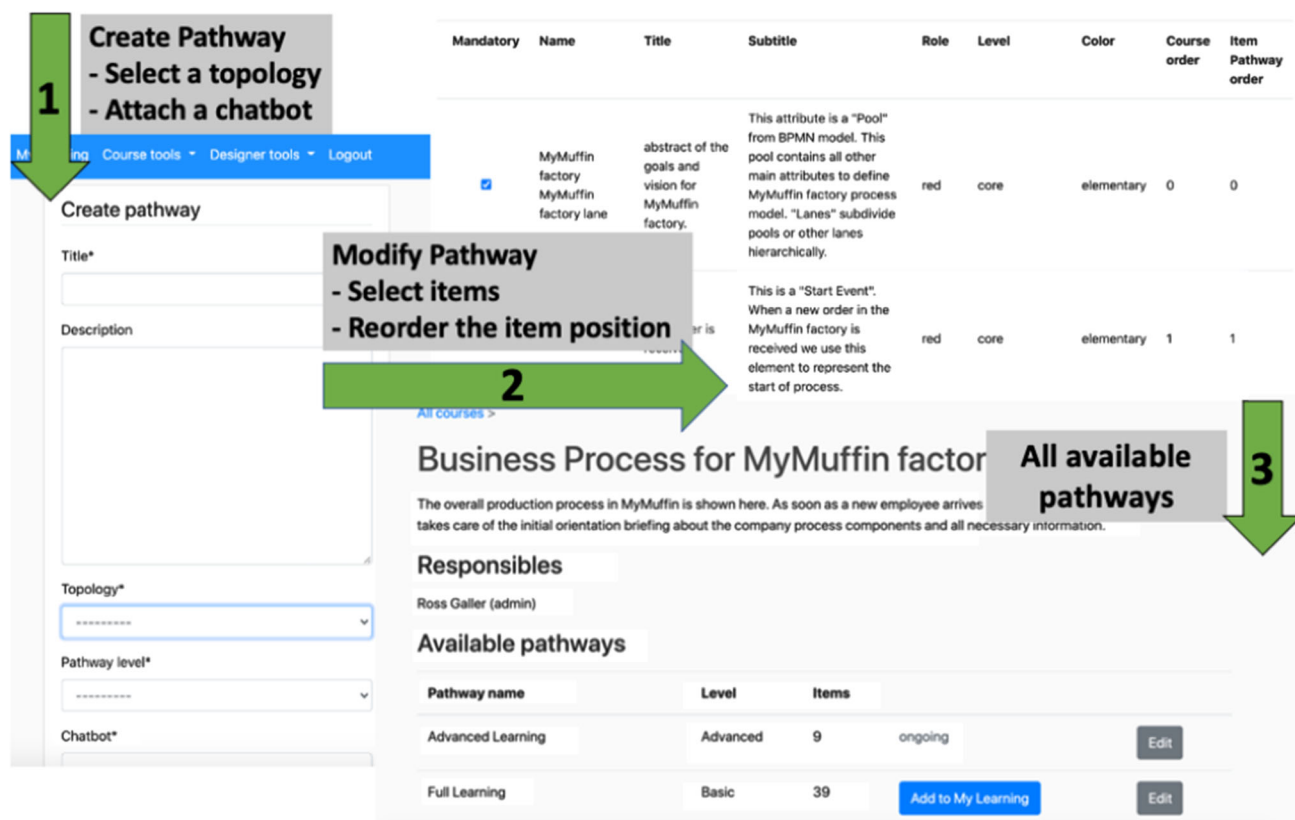


Fig. 11 A snapshot of the authoring tool for pathway creation in aCHAT-WF

wordings, but it would be more accurate with more training words. Later, to decode the user's intents, we rely on an IBM Watson Assistant, which will read the user's input and interpret its meaning. However, this architecture is quite (but not completely) independent of the underlying technology (it is also possible to use other chatbot technologies as Dialogflow, Amazon, Facebook).

An authoring tool has been developed in aCHAT-WF separately from the chatbot platform for authors and teachers to create and select a course and add their learning pathways. A snapshot of main steps on this authoring tool is demonstrated in Fig. 11.

The teachers in the MyMuffin use case applied all the predefined pathways mentioned in Sect. 6. Also, we defined an additional learning pathway related to the process properties. The data structure for the filter of this different pathway is presented in Table 4. This filter implies a learning pathway that includes all the learning items: tasks, events, data objects, attributes from all levels, roles and colors. However, with a skipable filter, we put more adaptivity on the learning experience and give learners the possibility to skip some learning items (e.g., pools and event items).

A short video showing the chatbot in action for students and the interface for the editor/teacher is uploaded for more consideration⁸.

9 Evaluation

We conducted a number of tests in order to assess the effectiveness⁹ of conversational interfaces for supporting learning, dealing with complex content as business processes. Secondary goals were detecting usability issues and comparing users' reactions to two different interfaces for the same content, i.e., a chatbot vs an interactive interface.

Figure 12 shows four steps for this assessment: (i) a preliminary survey was conducted to identify expectations; (ii) a survey was used for collecting the reactions to the use of the conversational interface; (iii) a survey was used for comparison between two different interfaces using the same content; (iv) an interview-focus group was used to dig into pedagogical issues. Surveys were used as a practical tool, not so much

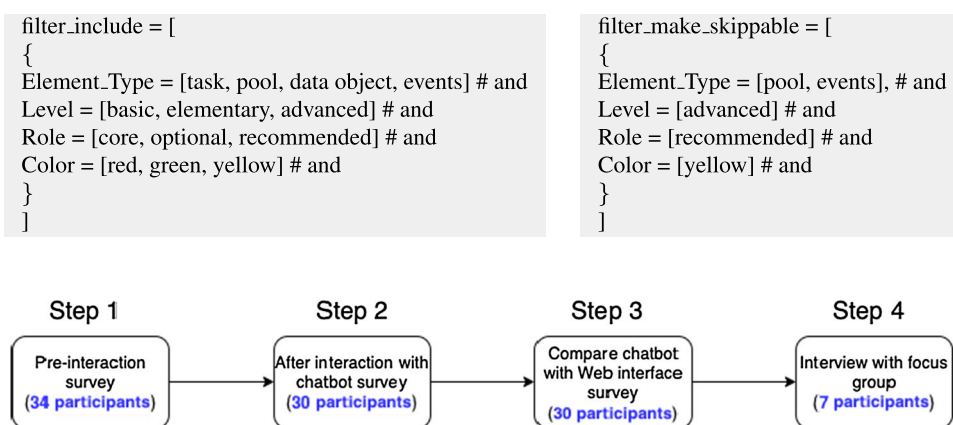
⁸ <https://youtu.be/7-NdqYGcunQ>

⁹ Effectiveness here refers to a self perception of the participants after using chatbots and the web interface for the same content.

Table 4 Example filter for a learning pathway for MyMuffin use case

<pre>filter_include = [{ Element.Type = [task, pool, data object, events] # and Level = [basic, elementary, advanced] # and Role = [core, optional, recommended] # and Color = [red, green, yellow] # and }]</pre>	<pre>filter_make_skippable = [{ Element.Type = [pool, events], # and Level = [advanced] # and Role = [recommended] # and Color = [yellow] # and }]</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------

Fig. 12 Evaluation steps



aiming at quantitative assessment, rather in order to identify issues and problems.

The study was conducted with students and teachers¹⁰ in Engineering in Computer Science and Human computer interaction (HCI) invited by emails from different universities in Italy¹¹.

In the pre-interaction survey (Step 1), we asked participants about: (i) their age group, (ii) if they are familiar with business process models and (iii) if they have an experience with any chatbot. The result from this survey showed that the majority of the participants are among early practitioners in BPM and interested in chatbots and digital transformation in general. Overall, 34 participants participated in the preliminary pre-interaction survey (see Fig. 12).

In order to assess the impact of a conversational interface, we offered the same contents in two ways: with a chatbot interface and with an interactive Web interface. The same learning items are offered on the process diagram, including speech items and their graphical representation. The chatbot was developed with aChat-WF, while the interactive Web interface was created with a specific tool developed by Politecnico di Milano¹² for creating an interactive web from text and video contents [8].

In Step 2, we evaluated the contents delivery with conversational interface built by aChat-WF. A snapshot of the conversational interface is shown in Fig. 13a. The interface has two main panels, the left one is used for delivering the learning contents to the learner, using most of the media types previously mentioned in Sect. 6 and the right panel is for delivering the chat. It is also possible to activate a voice conversation and listen to the speaking chatbot. One of the features of the *aCHAT-WF interface* is that a learner can also customize the conversation with adaptive features of the chatbot. Notably, in this test version, we designed a set of APIs to be able to control conversation adaptativity (change style, wording policy and talkativeness level) of the chatbot.

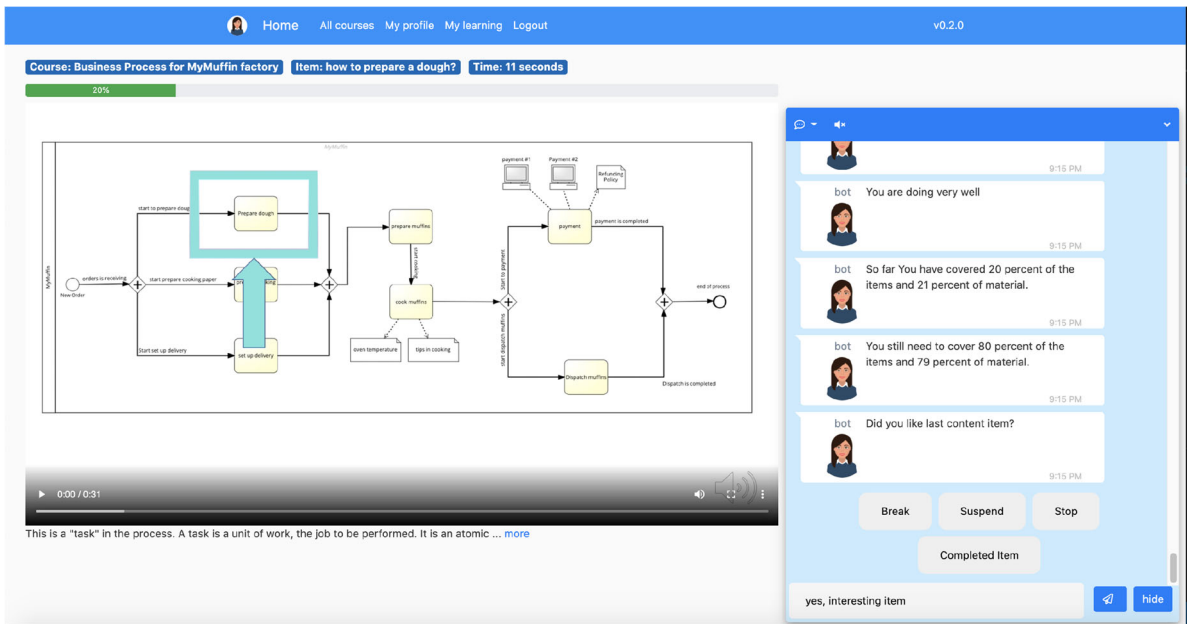
In Step 3, we compared the conversational interface with the interactive Web interface. The interactive Web interface shows the same learning materials managed with a more traditional point-and-click interface instead of a conversation. The learner instead of selecting the learning items engaging in a conversation, selects them from a sequential menu on the right and there are no conversational elements, neither to drive the conversation nor to add details on the learning item being displayed in the media panel (see Fig. 13b).

Finally, the final focus group had the role of further digging into relevant issues and to perform an in-depth analysis (Step 4). In this step, semi-structured interviews were conducted with some of the participants, and different opinions compared. Interviews with teachers were used to refine the overall requirements of the chatbot production process and

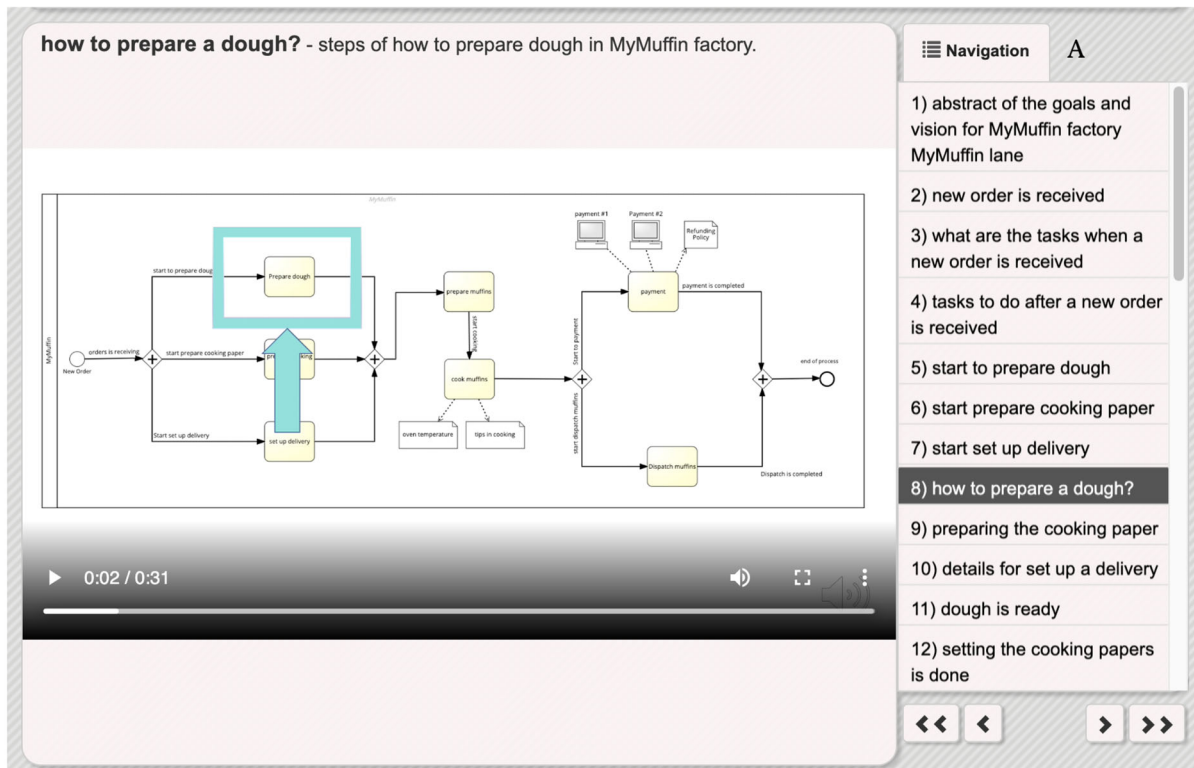
¹⁰ In this study, teachers refer to instructors and professors in the university.

¹¹ Mainly from Politecnico di Milano, Università di Brescia, and Sapienza Università di Roma.

¹² Cf. <http://www.1001storia.polimi.it/1001login/>.



(a) A snapshot of the chatbot and learning content in the aCHAT-WF interface.



(b) A snapshot of the content in the interactive Web interface.

Fig. 13 Teaching a process model with the interactive Web version vs. aCHAT-WF

design efforts. Interviews with students focused on the usability issues.

We have performed a set of empirical tests, with the following goals:

- to assess the overall “value of conversations” to enhance the learning experience;
- to improve the usability and acceptability of the chatbot;
- to assess the effectiveness (of the chatbot with respect to the interactive Web interface based on complex content, as in the case of business processes);
- to identify relevant features of conversational interfaces to be improved.

To address the mentioned goals, we considered a few qualitative assessments:

1. Is the conversation a valuable paradigm for online learning vs a traditional point-and-click version?
2. Are the adaptive features of the chatbot easy, usable, and effective for learners?

In the appendix, we illustrate the evaluation surveys proposed to the users in the first three steps, while interviews in the last step were held in a free format.

Over the total of 34 participants, 30 participants finished all 3 surveys (namely the pre-interaction survey, the one for interaction with the chatbot and the one for comparing the chatbot with the Web interface). At the end, 7 participants were interviewed and took part in a focus group for further analysis.

From the pre-interaction survey in Step 1, the volunteer participants are 82.4% under 35 years old and almost 80% of them have some knowledge about business processes. The selected candidates are 30% teachers and 70% students. Among all the participants, 80% have some previous experiences with chatbots in general and 93% of them think chatbots might be a support in education.

Figure 14 represents the summary of the results for the evaluation survey after the interaction with the chatbot (Step 2). This survey is conducted with some questions about the users’ experiences after using the chatbot. The main goal of this figure is to get requirements for creating chatbots with conversation adaptivity. In addition, the results in Fig. 14c and f show that users had different opinions about the current chatbot and possible improvements. This variety of viewpoints on controlling the chatbots show that customization of chatbot features is essential in order to accommodate the variety of users. “One conversation for all” does not seem the way to go.

In Step 2, after the questions, involved users were also asked to provide their comments through a free text field and proposals for new features. We thoroughly analyzed all the

answers and we identified the following issues recurrently highlighted by the users:

- several users described the chatbot as too fast at showing messages, especially with respect to the introductory ones, and this may result in the user simply ignoring some of the messages; this issue has been addressed by controlling the speed of the chatbot;
- some participants have found the questions at the end of each item too repetitive and in general annoying for the user, potentially resulting in a loss of interest; therefore, we modified the loquacity settings of the chatbot;
- most of the users requested the addition of specific commands to go back to the previous items and in general to be able to navigate the learning pathway more freely; these buttons have been added to the implemented framework.

After interaction with our chatbot, we asked participants to comment about their experience with chatbot in comparison to the Web interface (Step 3). The majority of participants (96%) agreed that the chatbot is more friendly than the Web interface. On the other hand, they had less strong opinions about whether the chatbot is more effective (76%) and usable (80%). Some of these aspects were analyzed in further detail in the interviews of the following Step 4.

Finally, in Step 4, seven participants were selected for interviews, keeping a balance between students and teachers (four students and three teachers). The interviews were organized according to the result of surveys and especially open comments of learners to investigate in depth pedagogical aspects, acceptance issues, and the “added value” provided by conversational interfaces vs Web interfaces. In these semi-structured interviews a set of predefined questions were provided to the interviewers to guide the interaction (see Appendix).

Overall, the participants found the idea of using chatbot as a tutor effective: *‘Interactive learning is working well and conversations add something’*¹³. Some students specifically liked the step-wise learning style (*‘stepwise learning with chatbot adds additional support’*).

All participants found the content customization (learning pathways) useful, especially for complex contents such as business processes:

- *‘It is good, especially for a difficult topic as BPM.’*
- *‘My style of learning: basic notions first and then details.’*
- *‘I like easy and introductory items first; then the most difficult parts.’*

¹³ Here the italics in quotes represent the direct quote from users in the interviews.

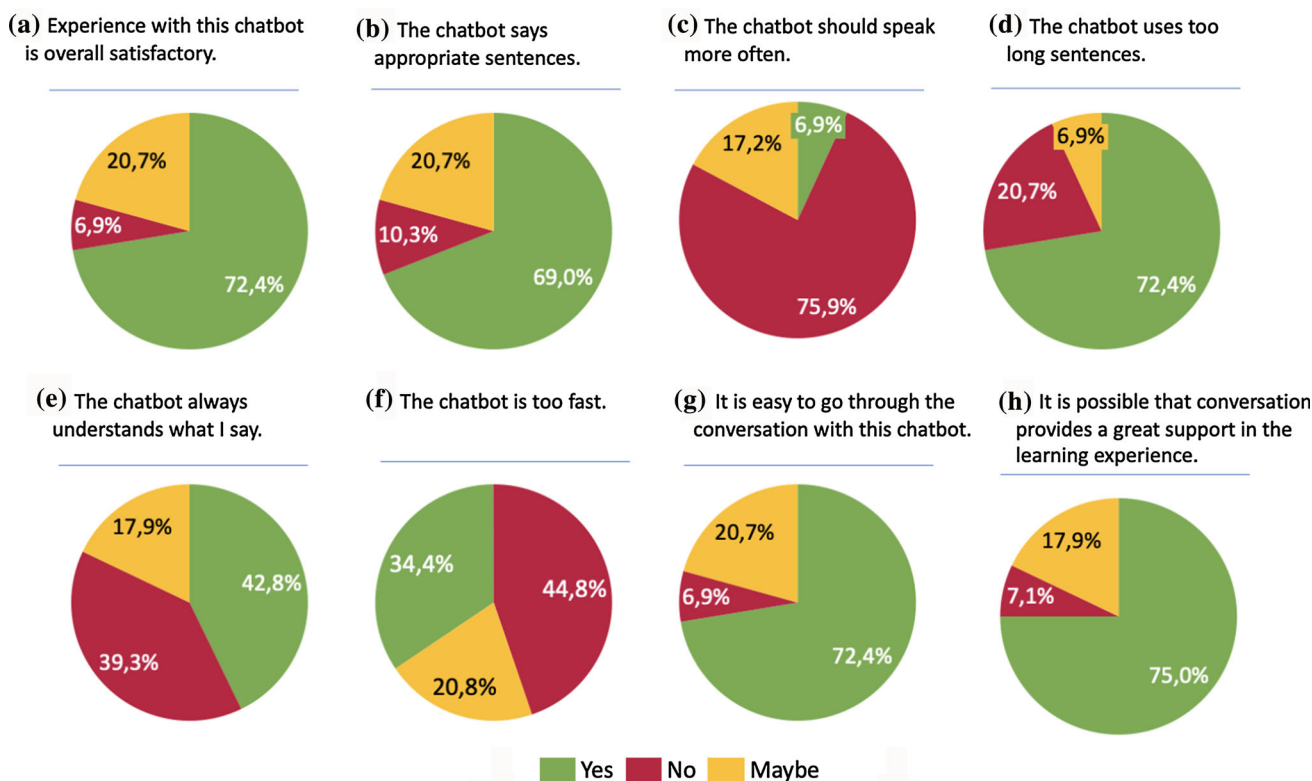


Fig. 14 Results of the survey about after interaction with chatbot (Step 2)

Participants had different opinions in advantages of conversation over interactive interface. For some of them the main advantage is functional ‘It helps focusing on content, since the chatbot takes care of details’. Apparently the chatbot seems to be helpful in traversing the pathway. For others the main advantage is psychological:

- ‘There is a feeling of someone is with you. Even greetings are OK’.
- ‘Someone is with you; you are not alone (while learning)’.
- ‘It seems that someone is with you (despite being artificial)’.

Everyone seems to like to be able to control the way the chatbot speaks. Students would like general control (e.g., the number of turns of the chatbot) or advanced control (e.g., the subject of the turns of the chatbot). Almost all teachers declared that would like to rewrite the wording of what the chatbot says. With this feature, teachers can write their own wording in a few hours effort, depending on the modifications scale, and add that style to the chatbot wording and personalize the learning experience for their students.

- ‘It would be great to improve the usability of the chatbot (that is too loquacious, currently).’

- ‘I would like to associate customization to my profile’.
- ‘I will spend for sure a few hours on this. It will greatly enhance the feeling that I’m there (presence) with them while they study’

Overall this preliminary validation shows that the chatbot is considered more friendly for learning with respect to the more traditional Web point-and-click interface and it is appreciated by learners with an interest in learning business processes. Moreover, this adaptive chatbot was appreciated by participants to be more usable and effective than Web interface. The adaptativity features (both for content and conversation) have a great potential for playing an essential role in eLearning and our approach (aCHAT-WF) seems to move in the right direction. These customization features seem to be a necessary ingredient for educational chatbots.

10 Concluding remarks and future work

This paper illustrated an original approach to generate a conversational chatbot separating the different aspects of the conversation: the chatbot interface, the conversation engine managing the conversation, the interpretation engine managing how to proceed with the conversation, and the action engine to select the next content item in the conversation.

In particular, this separation is achieved by adopting state machines as a conceptual representation mechanism, and simple structures, namely configurable tables, as an effective implementation tool.

The paper focused on business process education. A methodology and tool for extracting learning items from process models, represented in BPMN, have been illustrated and an example use case has been discussed. The level of knowledge for delivering a BPM can be different according to the various actors in an organization. The BPM knowledge experts define learning pathways or use metadata for creating an automated learning experience to include strategies of education for BPM.

We conducted a user assessment involving 30 different users, specifically recruited among early practitioners in BPM and students in Computer Engineering. Users were asked to evaluate the proposed solution configured to teach an example business process. Additionally, users were asked to compare the usability of the proposed solution with respect to a traditional one for learning using a Web application. Results of surveys firstly show how users generally evaluated well our solution, preferring it over the Web application. Users also provided comments through a free text field that will be taken into account for the development of future versions of the system. Although in our approach parsing a BPM and creating digital content (to plug it to some visual media and then to a configurable chatbot for delivering conversation) have been appreciated, we believe that, as far as learning is concerned, adaptive pathways by chatbots provide a different way to deliver content, not new content. Chatbots offer a new opportunity to create a different learning experience that being pleasant and smooth, and an artificial assistant may help the learner both functionally and psychologically (not feeling alone). We cannot claim that the learning a BPM with chatbot is self-sufficient, but we would like to point out by existence of the content related to BPM; according to what experimental data suggest, chatbots can provide an additional effective way of going through the content in a customized (or adaptive) way.

In a more general sense, we have a broad line of research, based upon the following directions and some limitations remain to be overcome:

- although using chatbot technology was well accepted by most of the participants, a few learners pointed that they do not like chatbots in general; nowadays, due to the popularity of the chatbots in different ways (e.g. Alexa or Siri), the willingness and the ability to trust the chatbot technologies varies between individuals;
- guided learning pathways are not always pleasant to the learners, and we got this feedback from a few participants

that they would like to explore everything rather than follow a pathway¹⁴;

- chatbots do not need be “truly humans”, but conversations should be rich and fluent, in order to engage users, in particular for complex tasks (such as learning); in adaptive conversations, a conversation designer can change the conversation model with a set of tables; later, a teacher can also add her wording to the chatbot; this feature is empowering different actors and stakeholders;
- configuration features over programming are crucial since they lower the costs and empower actors to directly control their chatbot; extensive use of configuration to control the chatbot, speak faster vs slower, or change the chatbot loquacity is needed to make the chatbot application sustainable in the organizations;
- adaptivity (for end users) and customization (for business actors along the production chain) are crucial aspects for the development of a pervasive chatbot market;
- for serious-purpose chatbots, usability, persuasiveness and effectiveness are more relevant than “human-like” features; due to the nature of teaching, this chatbot is more proactive and controls the learning experience, rather than being a question-answering machine similar to other chatbots.

Future work is needed to make the chatbot more adaptive to the dynamic conditions of the learning and the user’s profile. We have to improve the strength of the architecture, also considering how to add several of the functionalities suggested by users. In addition, while the chatbot is used only for teaching in this paper, it can be potentially used at process runtime for analysis purposes. As an example, if a process monitoring service is available, as soon as an exceptional condition occurs, the chatbot can be used to explain at which stage the problem is and possible countermeasures to be applied. In this sense, the proposed chatbot is very well suited to be integrated with Digital Twins - DTs. A DT is a digital representation of a physical entity, and a common employment of this concept is to wrap human and not-human actors of the production process (e.g., industrial equipment). As such, DTs events can be mapped to tasks in a business process, and then be easily integrated with the chatbot, thus providing information, warnings and error notifications to human operators and fostering a Human-In-the-Loop approach to smart manufacturing architectures [6]. Additionally, so-called Experimental DTs also provide prediction functionalities, thus allowing to perform what-if analyses whose parameters could be selected through the chatbot.

¹⁴ One participants said that ‘In general I think they are, but I personally tend to explore everything that is given to me, so for me it would end up as a waste of time because I would have to check every pathway’

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 and the EU H2020-WIDESPREAD project DESTINI. 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.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix: Evaluation surveys

Step 1: Questions to assess user knowledge before the interaction with chatbot (Yes or No)

Question Number	Question
1	Do you have any knowledge about business process models?
2	Have you ever used a chatbot?
3	Do you think that a chatbot could help you supporting your learning process?

Step 2: Questions to assess interaction after using the chatbot (Likert scale 1 to 5)

Question Number	Question
1	My experience with this chatbot was overall satisfactory
2	This chatbot always says appropriate sentences
3	This chatbot should speak more often
4	This chatbot uses sentences of the proper length
5	This chatbot uses too long sentences
6	This chatbot uses sentences of the proper style
7	This chatbot always understands what I say
8	Chatbot messages are clear to me
9	It is easy to go through the conversation with this chatbot?
10	The chatbot is too fast
11	The chatbot is too slow
12	It is possible that conversation provides a great support in the learning experience?

Question Number	Question
13	Add your general comments here (free text)
14	Which features would you like to be added to the chatbot? (free text)

Step 3: Questions to assess comparison interaction of chatbot's features vs. Web interface (Likert scale 1 to 5)

Question Number	Question
1	Chatbot is more friendly than the web interface
2	Chatbot is more effective than the web interface
3	Chatbot is more usable than the web interface

Step 4: Interview guideline for discussion with participants

Question Number	Question
1	What do you think of the "idea" of chatbots in education (BPM)?
2	What do you think about the customization of learning (pathways)?
3	Where do you see the possible advantages of conversations over Web interactive interfaces?
4	Would you prefer a combination of interactive/ conversational interfaces?
5	Would like direct control over loquacity/ style/ wording?
6	What do you like and dislike about this chatbot?

References

- 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)
- Barakat, K. A., Dabbous, A.: Understanding the factors that affect the sustained use of chatbots within organizations. *IADIS Int. J. on WWW/Internet* **17**(2),(2019)
- Biococchi, 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)
- 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. 144–48. CEUR-WS, (2019)
- 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)
- 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 *2019 IEEE*

- International Conference on Web Services (ICWS)*, pp. 229–236. IEEE, (2019)
7. Chandar, P., Khazaeni, Y., Davis, M., Muller, M., Crasso, M., Liao, Q. V., Shami, N. S., Geyer, W.: Leveraging conversational systems to assist new hires during onboarding. In Regina Bernhaupt, Girish Dalvi, Anirudha Joshi, Devanuj K. Balkrishan, Jacki O'Neill, and Marco Winckler, editors, *Human-Computer Interaction - INTERACT 2017*, pp. 381–391, Cham, (2017). Springer International Publishing
 8. Di Blas, N., Paolini, P., Sabiescu, A.G.: Collective digital storytelling at school: a whole-class interaction. *Int. J. Arts Technol.* **5**(2–4), 271–292 (2012)
 9. Doan, A. H.: Human-in-the-loop data analysis: A personal perspective. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pp. 1–6, (2018)
 10. Frank, A.G., Mendes, G.H.S., Ayala, N.F., Ghezzi, A.: Servitization and Industry 4.0 convergence in the digital transformation of product firms: A business model innovation perspective. *Technol. Forecast. Social Chang.* **141**, 341–351 (2019)
 11. Heilman, M., Smith, N. A.: Question generation via overgenerating transformations and ranking. Technical report, Carnegie Mellon University, Pittsburgh, PA, Language Technologies Institute, (2009)
 12. Hermann, M., Pentek, T., Otto, B.: Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 3928–3937. IEEE, (2016)
 13. Holotescu, C.: MOOCbuddy: a chatbot for personalized learning with MOOCs. In *RoCHI-International Conference on Human-Computer Interaction, A. Iftene and J. Vanderdonck (eds), Bucarest*, pp. 91–94, (2016)
 14. Ilievski, V., Musat, C., Hossmann, A., Baeriswyl, M.: Goal-oriented chatbot dialog management bootstrapping with transfer learning. In *IJCAI'18: Proceedings of the 27th International Joint Conference on Artificial Intelligence*, (2018)
 15. Kalia, A. K., Telang, P. R., Xiao, J., Vukovic, M.: Quark: a methodology to transform people-driven processes to chatbot services. In *International Conference on Service-Oriented Computing*, pp. 53–61. Springer, (2017)
 16. López, A., Sánchez-Ferreres, J., Carmona, J., Padró, L.: From process models to chatbots. In *International Conference on Advanced Information Systems Engineering*, pp. 383–398. Springer, (2019)
 17. Mantravadi, S., Jansson, A. D., Møller, C.: User-friendly mes interfaces: Recommendations for an ai-based chatbot assistance in industry 4.0 shop floors. In *Asian Conference on Intelligent Information and Database Systems*, pp. 189–201. Springer, (2020)
 18. McTear, M.: Conversation modelling for chatbots: current approaches and future directions. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung* **2018**, 175–185 (2018)
 19. Moore, R. J.: A natural conversation framework for conversational ux design. In *Studies in Conversational UX Design*, pp. 181–204. Springer, (2018)
 20. Moormann, J., Bandara, W.: Learning, teaching and disseminating knowledge in business process management. *Knowl. Manag. E-Learn.: Int. J.* **4**(4), 390–394 (2012)
 21. Paek, T., Pieraccini, R.: Automating spoken dialogue management design using machine learning: An industry perspective. *Speech Commun.* **50**(8–9), 716–729 (2008)
 22. 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)
 23. Polyvyany, A., Smirnov, S., Weske, M.: Business process model abstraction. In *Handbook on Business Process Management 1*, pp. 147–165. Springer, (2015)
 24. Porreca, S., Leotta, F., Mecella, M., Vassos, S., Catarci, T.: Accessing government open data through chatbots. In *International Conference on Web Engineering*, pp. 156–165. Springer, (2017)
 25. Rabelo, R. J., Romero, D., Zambiasi, S. P.: Softbots supporting the operator 4.0 at smart factory environments. In Ilkyeong Moon, Gyu M. Lee, Jinwoo Park, Dimitris Kiritsis, and Gregor von Cieminski, editors, *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0*, pp. 456–464, Cham, (2018). Springer International Publishing
 26. Rahman, A. M., Al Mamun, A., Islam, A.: Programming challenges of chatbot: Current and future prospective. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 75–78. IEEE, (2017)
 27. Romero, D., Stahre, J., Wuest, T., Noran, O., Bernus, P., Fast-Berglund, Åsa, Gorecky, D: 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)
 28. Rooein, D., Bianchini, D., Leotta, F., Mecella, M., Paolini, P., Pernici, B.: Chatting about processes in digital factories: A model-based approach. In Selmin Nurcan, Iris Reinhartz-Berger, Prina Soffer, and Jelena Zdravkovic, editors, *Enterprise, Business-Process and Information Systems Modeling - 21st International Conference, BPMDS 2020, 25th International Conference, EMM-SAD 2020, Held at CAiSE 2020, Grenoble, France, June 8-9, 2020, Proceedings*, volume 387 of *Lecture Notes in Business Information Processing*, pp. 70–84. Springer, (2020)
 29. Tegos, S., Demetriadis, S., Psathas, G., Tsiatsos, T.: A configurable agent to advance peers productive dialogue in moocs. In *International Workshop on Chatbot Research and Design*, pp. 245–259. Springer, (2019)
 30. Visnjic, I., Ringov, D., Arts, S.: Which service? how industry conditions shape firms service-type choices. *J. Prod. Innov. Manag.* **36**(3), 381–407 (2019)
 31. Zimmer, M., Yacoub, A., Ferreira, P., Lohse, N.: Towards Human-Chatbot Interaction: A virtual assistant for the ramp-up process. In *UK-Robotics and Autonomous Systems Conference (UK-RAS2020)*, (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Donya Rooein is a PhD student in Information Technology Engineering at Politecnico di Milano. Her research interest is Conversational AI, machine learning, and information systems. These topics applied to several applications such as developing sustainable chatbots for industry 4.0 and education field. Her PhD program is supported by EIT Digital (European Institute of Innovation and Technology) and IBM Italy.



Devis Bianchini is full professor at Department of Information Engineering, University of Brescia. Main research interests are focused on models and tools for Web Information System (WIS) design, service-oriented architectures, semantics-driven resource discovery, methods and techniques for the management and exploration of Big and Open Data. These topics have been applied several application domains such as, among them, Industry 4.0 and Smart Cities. He published over

130 research papers and coordinated (totally or partially) research projects on the above topics.



Francesco Leotta is assistant professor at Department of Computer, Control and Management Engineering of Sapienza Università di Roma, Italy. His research concerns algorithmic, methodological, experimental and practical aspects in different areas of Computer Science, including ubiquitous computing, human-computer interaction and digital humanities. Such topics are challenged in the application domains of smart spaces, smart manufacturing and cultural heritage.



Massimo Mecella is a full professor with Sapienza Università di Roma. His research focuses on business process management, cyber-physical systems and Internet-of-Things, advanced interfaces and human-computer interaction, software architectures and service-oriented computing, with applications in multiple fields including digital government, smart spaces, Industry 4.0, healthcare, disaster/crisis response and management, cyber-security, digital humanities. He published over

180 research papers and chaired different conferences in the above areas.



Paolo Paolini is founder of HOC-LAB, a multidisciplinary laboratory of Politecnico di Milano. He graduated at the University of Milan; Master and Ph.D. in Computer Science from the University of California at Los Angeles (UCLA). He has been active in different research fields; recently use of technology for Cultural Heritage and Education, behaviour data analysis, technology enhanced intervention, data driven chatbots supported by AI. He has authored more than 200 scientific papers

and has coordinated more than 50 research projects.



Barbara Pernici is full professor in Computer Engineering at the Politecnico di Milano. Her research interests include adaptive information systems, data quality, IS energy efficiency, and social media analysis. She has published more than 70 papers in international journals and about 350 papers at international level. She has lead the information systems group of Politecnico di Milano in many European and national projects. She is currently responsible of the Politecnico unit of the

Crowd4SDG European H2020 project. She was an elected chair of TC8 Information Systems of the International Federation for Information Processing (IFIP), of IFIP WG on Information Systems Design, vice-chair of the IFIP WG on Services-Oriented Systems, and chair of the Steering Committee of the international Conference on Advanced Information Systems Engineering (CAiSE). She has chaired or cochaired main conferences, as general chair or program chair, including CAiSE, ER, BPM, ICSOC, Coopis, tracks in ICSE and ICIS.