

# DemaBot: a Tool to Automatically Generate Decision-Support Chatbots.

Bedilia Estrada-Torres<sup>a</sup>, Adela del-Río-Ortega<sup>b</sup> and Manuel Resinas<sup>b</sup>

<sup>a</sup>Universidad de Sevilla, Seville, Spain

<sup>b</sup>SCORE Lab, I3US Institute. Universidad de Sevilla, Seville, Spain

## Abstract

This article presents DemaBot: a low-code solution to create chatbots designed to automate decision-making. Examples of these chatbots range from COVID-19 screening to first-line incident support, amongst others. Using DemaBot, the developer only needs to model the decision that the chatbot will automate using DMN and, optionally, customize the utterances that the chatbot will use to interact with the user. From this information, DemaBot generates automatically the complete set of components that implement a ready-to-use chatbot. Furthermore, it provides help to guide users during the conversation, and optimizes the conversation flow, being able to recognize several parameters in a single turn and asking only for those that are indispensable for the decision.

## Keywords

Decisions-support chatbot, DMN, Natural language understanding

## 1. Introduction

In recent years, a large number of organizations have implemented virtual assistants that provide information, solve doubts or help in the achievement of a specific task [1]. Often these virtual assistants take the form of chatting bots, also known as *chatbots*, which are tools designed to interact with users through friendly conversations using natural language in a way that simulates interaction with a human [2, 1], and in some scenarios, they can act as teammates of human workers in a collaborative way in complex processes [3].

Although chatbot development platforms have abstracted away many details related to the natural language processing of a chatbot through the automated recognition of user intents and have provided a framework in which the conversation flow can be defined, a chatbot developer is still required to implement a specific conversation flow, to deal with many low-level details about parameters or entities, to provide a set of training phrases for each of the conversation steps, and to provide a generic set of fallback options that guides users who do

---

*Proceedings of the Demonstration & Resources Track, Best BPM Dissertation Award, and Doctoral Consortium at BPM 2021 co-located with the 19th International Conference on Business Process Management, BPM 2021, Rome, Italy, September 6-10, 2021*

✉ iestrada@us.es (B. Estrada-Torres); adeladelrio@us.es (A. del-Río-Ortega); resinas@us.es (M. Resinas)

🌐 <https://personal.us.es/iestrada/> (B. Estrada-Torres)

🆔 0000-0001-7943-276X (B. Estrada-Torres); 0000-0003-3089-4431 (A. del-Río-Ortega); 0000-0003-1575-406X (M. Resinas)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

not know the capabilities the chatbot has, amongst others. Implementing these tasks is usually time-consuming and error-prone even for simple chatbots [4].

In this paper, we focus on *decision-support chatbots*, which can be defined as task-oriented chatbots whose purpose is to help or guide users to make decisions. DemaBot is a web tool that provides a low-code solution to create *decision-support chatbots*<sup>1,2,3</sup>. To this end, the developer only needs to model the decision that the chatbot will automate using a DMN [5] model. DMN (Decision Model and Notation) is a standard that provides constructors for modeling decisions and decoupling them from process models. Based on the DMN model, DemaBot generates automatically a ready-to-use chatbot, thus reducing the implementation time of a chatbot and the possibility of implementation errors. The generated chatbot leverages the natural language understanding (NLU) platform, Dialogflow<sup>4</sup>. DemaBot also optimizes the conversation flow with the user to recognize several parameters in a single conversation turn and asks only for the parameters essential to make the decision, automatically generates training phrases using a dataset generator for NLU models, provides help phrases to guide users if necessary, and allows customization of questions and answers.

## 2. DemaBot Architecture

Before introducing the DemaBot architecture, we introduce DMN decision tables and NLU chatbots. A DMN decision table, shown in Fig. 1a, “is a tabular representation of decision logic, based on a discretization of the possible values of the *inputs* of a decision, and organized into *rules* that map discretized input values onto discrete *output* values”[5]. A NLU chatbot, also known as *agent*, contains a recognition engine that matches user inputs with intentions (*intents*) during a conversation turn. It also contains an execution component capable of executing *actions* for each intent, such as responses to the user. *Intents* are defined through *training phrases*, which are input examples that allow the recognition engine to identify the different phrases that a user can utilize to express an intention. In each *training phrase*, concrete values can be recognized. These values are called *parameters*. The type of the parameters is defined by a specific structure called *entity*, which determines how data from a user input is extracted. NLU platforms provide predefined entities that match many common types of data like numbers, dates, e-mail addresses, etc. but also allow custom *entities* to be defined for enumerated values. A custom entity is composed of a set of *entries*. Each entry is made up of a reference value and a set of synonyms for that reference. These components are reflected in Fig. 1b.

The DemaBot architecture consists of 3 modules illustrated in Fig. 2:

- **Chatbot Designer** supports the definition of the inputs of DemaBot, which are a *DMN file* containing a DMN model, and configuration parameters including the customization of the chatbot conversation and the *credentials file* associated with a Dialogflow project where the generated chatbot components will be stored. *Chatbot designer* provides 3

---

<sup>1</sup>Tool is available at: <http://resisa01.us.es>

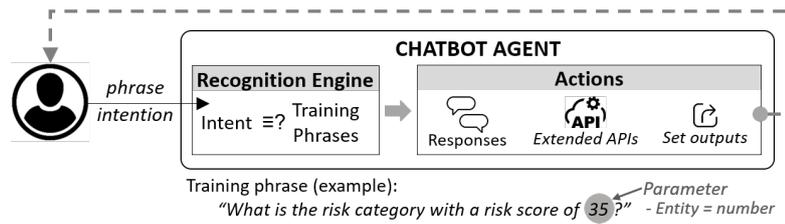
<sup>2</sup>Tutorial available at: <https://bit.ly/2SRZ0TY>

<sup>3</sup>Screencasts is available at: <https://youtu.be/kP3X5gxkCSU>

<sup>4</sup>DialogFlow: <https://dialogflow.cloud.google.com/>

Decision		Inputs		Output
Risk Category	Existing Customer	Risk Score	Credit Score	Risk Category
U				
1	True	<=120	-	"LOW"
2	True	>120	<600	"HIGH"
3	True	>120	[600, 625]	"MEDIUM"
4	True	>120	>625	"LOW"
5	False	<=100	<580	"HIGH"
6	False	<=100	[580, 600]	"MEDIUM"
7	False	<=100	>600	"LOW"
8	False	>100	-	"HIGH"

(a) DMN decision table



(b) Interaction between NLU chatbot components

**Figure 1:** Examples of background concepts

functionalities: (i) the DMN editor integrated with Camunda<sup>5</sup> from which a DMN model (a decision table or a hierarchical structure of decision tables) can be generated, or an existing *DMN file* can be imported; (ii) the possibility to customize the chatbot conversation by configuring the questions and answers concerning an input in a conversation turn, and (iii) the generation of the chatbot, which requires loading the *credentials file* and providing the *project ID*.

- **Engine.** At design time, this module processes the inputs received from the *Designer* and interacts with Dialogflow to generate and store all the chatbot components. *Engine* also connects with Chatito<sup>6</sup> to build the phrases that will train the generated chatbot. At runtime, *Engine* interacts with the *Interface* module and analyzes the inputs received from the user to optimize the conversation. *Engine* has three submodules described below.

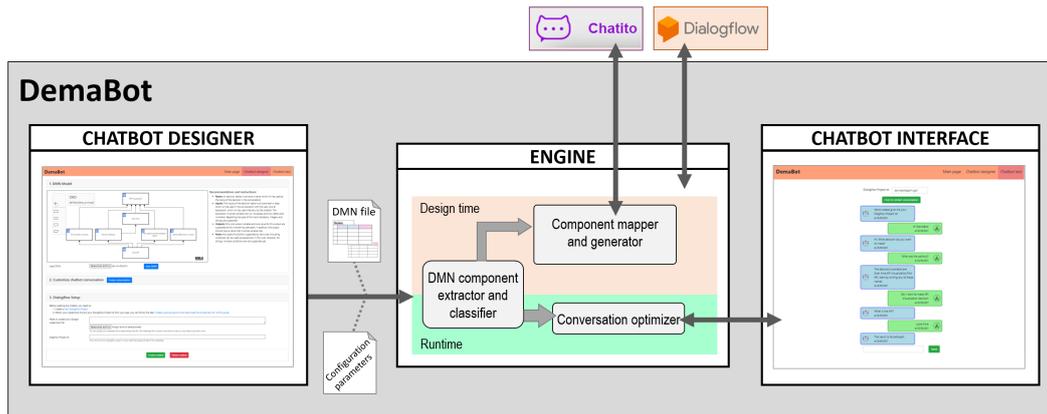
(i) *DMN component extractor and classifier* submodule identifies the DMN components (decisions, rules, inputs and outputs) and classifies the expected values and types for each component from the *DMN file*. At design time, this data, the *credentials file* and the *project ID* are sent to the *Component mapper and generator*. At run time, the data is sent to the *Conversation optimizer*.

(ii) *Component mapper and generator* submodule receives the classified DMN elements and identifies the chatbot components to automatically generate for each of them, so it has a *generator* for each component. *Entities generator* creates an entity for each type of attribute not supported by the platform; *Intents generator* creates an intent for each DMN input, as well as generating help intents to give additional information to the user, for example the list of decisions available to be made, or the types of values expected for an input; *Actions generator* specifies the actions to be performed, including the responses to be sent to the user; *Multiple decision manager* establishes relationships between decision tables in hierarchical structures when the output of one table is the input of another; and *Training phrases generator* uses the Chatito engine to build training phrases that will help the user interact with the chatbot. These phrases are automatically generated using the parameters extracted from the *DMN file* and optionally, from user-defined phrase configuration parameters.

(iii) *Conversation optimizer* submodule interacts with the *extractor and classifier* submod-

<sup>5</sup><https://camunda.com/>

<sup>6</sup>Chatito is available at: <https://rodrigopivi.github.io/Chatito/>



**Figure 2:** DemaBot architecture

ule at runtime to communicate with the Dialogflow platform to recognize the intents, parameters and entities from the user utterances. It analyzes the information provided by DialogFlow to identify the required inputs to make a decision. It also applies an optimization algorithm to ask for input values only if the input is required depending on the values captured so far, and recognizes several parameters (inputs) received in a single user message, thus minimizing the number of interactions to produce the decision.

- **Chatbot interface** provides an environment for the communication between the user and the generated chatbot. It communicates with the *Conversation optimizer* to minimize conversation turns with the user. The chatbot interface can also be used as a chatbot test web environment to check the behavior of the chatbot while designing the model.

### 3. Maturity

DemaBot is the result of several iterations derived from its use in different scenarios and feedback of a set of pilot users. DemaBot's first version focused on the semi-automatic components generation of a fully functional chatbot. It was tested and validated with a set of 17 IT professionals with a scenario based on bank credit decision making. The participants identified the importance of the tool, and also helped to identify areas for improvement such as: the fully automatic generation of all components, the inclusion of hierarchical decision tables definition, the improvement of the definition of help intents to guide the user in the interaction with the chatbot, the optimization of aspects of the conversation so that unnecessary questions are not asked to make a decision. We considered these suggestions and are included in the current version. This version has been successfully applied to two scenarios. The first scenario helps to choose the best design pattern for building Trello boards based on characteristics of the domain [6]. The second scenario seeks to build a dashboard showing the results of process performance indicators according to the characteristics of these indicators based on a decision tree proposed in [7]. In a Master's Thesis [8] at the University of Seville, a chatbot based on a part of this decision tree was implemented manually, consuming more than 50 hours of work,

while with DemaBot it was done in less than an hour.

## 4. Conclusions and Future Work

This paper presented DemaBot, a web tool for the automatic generation of fully functional *decision-support chatbots* from DMN models. The automatic generation reduces the possibility of introducing errors during the chatbot construction, as well as drastically reducing the time required for its generation. In addition, DemaBot provides additional features such as the automatic generation of user help responses, the automatic generation of training phrases for the chatbot and the optional configuration of phrases to interact with the user, as well as optimizing user interactions to try to minimize conversational turns and identify multiple inputs in a phrase. As future work we want to address multi-user interaction in conversational turns for decision making as well as its integration with other collaborative tools.

## Acknowledgments

This work has been partially supported by projects RTI2018-101204-B-C22 (MCI/AEI/FEDER, UE) and P18-FR-2895 (Junta de Andalucía/FEDER, UE).

We would like to thank Alejandro Gómez Caballero, member of the development team of the ISA group, for his support in the development of DemaBot.

## References

- [1] S. Valtolina, B. R. Barricelli, S. D. Gaetano, P. Diliberto, Chatbots and conversational interfaces: Three domains of use, in: Fifth International Workshop on Cultures of Participation in the Digital Age, volume 2101 of *CEUR Workshop Proceedings*, 2018, pp. 62–70.
- [2] S. Hussain, O. A. Sianaki, N. Ababneh, A Survey on Conversational Agents/Chatbots Classification and Design Techniques, in: Workshops of the 33rd International Conference on Advanced Information Networking and Applications, volume 927, 2019, pp. 946–956.
- [3] I. Seeber, E. A. C. Bittner, R. O. Briggs, et al., Machines as teammates: A research agenda on ai in team collaboration, *Information & Management* 57 (2020) 103174.
- [4] G. Daniel, J. Cabot, L. Deruelle, M. Derras, Multi-platform Chatbot Modeling and Deployment with the Jarvis Framework, in: 31st Int. Conf. on Advanced Information Systems Engineering (CAiSE 2019), volume 11483 of *LNCS*, Springer, 2019, pp. 177–193.
- [5] I. Object Management Group, Decision Model and Notation (DMN) specification. V1.3, <https://www.omg.org/spec/DMN>, 2019. Accessed: 03-03-2021.
- [6] J. Peña, A. Bravo, A. del-Río-Ortega, M. Resinas, A. Ruiz-Cortés, Design patterns for board-based collaborative work management tools, in: 33rd Int. Conf. Advanced Information Systems Engineering (CAiSE 2021), volume 12751 of *LNCS*, Springer, 2021, pp. 177–192.
- [7] Ü. Aksu, A. del-Río-Ortega, M. Resinas, H. A. Reijers, An approach for the automated generation of engaging dashboards, in: OTM, volume 11877 of *LNCS*, 2019, pp. 363–384.
- [8] M. D. Gil, Chatbots para cuadros de mando, Master’s thesis, E.T.S.I. Informática, 2020. Webhook Dialogflow: <https://github.com/MrManoloDG/friday-chatbot>.