colMOOC™

# D2.1

# Conversational Agent Model Design

Co-funded by the
Erasmus+ Programme
of the European Union

Co-funded by the Erasmus+ Programme of the European Union

# The colMOOC:

# Integrating Conversational Agents and Learning Analytics in MOOCs

# D2.1 - Conversational Agent Model Design

| | |
|---|---|
| **Project number:** | 588438-EPP-1-2017-1-EL-EPPKA2-KA |
| **Grant Agreement No:** | 2017-2841/001-001 |
| **Project acronym:** | colMOOC |
| **Project title:** | Integrating Conversational Agents and Learning Analytics in MOOCs |
| **Programme, Key action, Action type, Topic:** | E+ KA2: Cooperation for innovation and the exchange of good practices, Knowledge Alliances |
| **Start date of the project:** | 01/01/2018 |
| **Duration:** | 36 months |
| **Project web site:** | https://colmooc.eu/ |

| | |
|---|---|
| **Deliverable type:** | Report |
| **Deliverable reference number:** | D2.1 |
| **Deliverable title:** | Conversational Agent Model Design |
| **WP contributing to the deliverable:** | WP2 |
| **Delivery date:** | 06/20/2018 |

| | |
|---|---|
| **WP Leader:** | AUTH |
| **Responsible organization:** | AUTH |
| **Abstract:** | This report will make a thorough state of the art analysis on existing CA approaches in order to identify models, ontologies, vocabularies that can be re-uses, as well as any gaps that could be filled by the proposed model. |
| **Keywords:** | Agent, conversational agent, MOOC, model design |

| | |
|---|---|
| **Dissemination level:** | Public |

## Authors and Reviewers

Stavros Demetriadis, Aristotle University of Thessaloniki (AUTH),Greece, sdemetri@csd.auth.gr

Stergios Tegos, Aristotle University of Thessaloniki (AUTH),Greece, stegos@csd.auth.gr

Georgios Psathas, Aristotle University of Thessaloniki (AUTH), Greece, gpsathas@csd.auth.gr

**REVIEWERS**

Yannis Dimitriadis, Universidad de Valladolid (UVa), Spain, yannis@tel.uva.es

Santi Caballe, Universitat Oberta de Catalunya (UOC), Spain, scaballe@uoc.edu

## Document Change Log

| Version | Date (mm/dd/yyyy) | Author (s) | Sections Changed |
|---|---|---|---|
| 0.1 | 03/01/2018 | See author list | ALL |
| 0.2 | 06/20/2018 | See author list | ALL |
| 1.0 | 06/30/2018 | See author list | Minor Revisions |

## Executive Summary

This deliverable:

- Initially provides a state-of-the-art review of current design and implementation approaches in the area of conversational agents, emphasizing their use in education to support: individual instruction, group learning and MOOC activities.
- Then it presents the key aspects of the conversational agent to be developed in the context of the colMOOC project explaining all important design perspectives, such as:
    - Design to support productive forms of peer dialogue
    - Design as domain-independent teachers' open tool
    - Design as interoperable tool to be integrated in MOOCs platforms.

Other major aspects of the proposed design are also discussed, including:

- Agent intervention strategies
- Software key components (Editor and Player)
- Agent domain configuration by the teacher, and
- Agent integration with various MOOCs platforms

## Table of Contents

## List of Tables

## List of Figures

## List of Acronyms

| Acronym | Description |
|---|---|
| AB | Advisory Board |
| APT | Academically Productive Talk |
| CML | Chatbot Mediated Learning |
| DLP | Deliverable Lead Partner |
| EB | Ethics Board |
| IPG | Intellectual Property Management Group |
| MOOCs | Massive Open Online Courses |
| PC | Project Coordinator |
| QA | Quality Assurance |
| QC | Quality Control |
| QEG | Quality Evaluation Group |
| SB | Project Supervisory Board |
| SM | Scientific Manager |
| TM | Technical Manager |
| ToC | Table of Contents |
| UG | User Group |
| WPL | Work Package Leader |

# 1 Introduction

## 1.1 Purpose of this document

The objective of this deliverable is to:

- A) Present a thorough state of the art analysis on existing Conversational Agent (CA) approaches in order to identify models, ontologies, vocabularies that can be re-used, and
- B) Identify any gaps that could be filled by the proposed model.

## 1.2 Document structure

To accomplish the above objective the present deliverable is split into two major chapters:

### 1.2.1 State of the Art Analysis of Existing Conversational Agent Approaches

In the first chapter a state-of-the-art review is presented of various existing CA approaches highlighting the key functionalities of each system and context of expected use.

This chapter also includes a section entitled "Conversational Agents: The Potential for Education" which discusses the impact of the CA tools in education with emphasis on two major areas of implementation, namely:

- Agents in one-on-one learning settings
- Agents to support groups

The important perspective of using agents in the context of MOOCs (Massive Open Online Courses) is also highlighted and discussed within the above two sections.

### 1.2.2 Proposed model

In the second chapter we present a proposed model for leveraging CA technology in MOOCs as implemented in the context of the colMOOC project.

First, in the subsection "Identified gaps to guide the conversational agent design approach", we analytically discuss the three major pillars of the proposed design, namely:

- The colMOOC agent to support productive forms of peer dialogue (cognitive dimension)
- The colMOOC agent as domain-independent teachers' open tool (socio-cultural dimension), and
- The colMOOC agent as an interoperable tool to be integrated in MOOCs platforms (technological dimension).

Then, we move on to discuss specific aspects of the proposed model such as:

- The agent intervention strategies in students' chat activities
- The agent software key components: the Editor and the Player
- The agent domain configuration by the teacher

## 1.3 Audience

This document is open and publicly available.

# 2 State of the Art Analysis of Existing Conversational Agent Approaches

This chapter introduces the reader to the state-of-the-art in the area of conversational agents, providing also concise reviews of several of the current CA design and implementation approaches. The chapter also reviews the most important perspectives of conversational agents in education to support: individual instruction, group learning and MOOC activities.

## 2.1 An introduction to Conversational Interfaces

In the last decade, Natural Language Processing (NLP) technology has been improved and become more accessible than ever. Nowadays, the combination of NLP algorithms with deep learning and neural networks provide new opportunities that empower developers to build interfaces that are highly engaging and human-like (Pelk, 2016). A conversational user interface provides the privilege of interacting with the computer on human terms giving the illusion of a human-to-human interaction (Klopfenstein, Delpriori, Malatini, & Bogliolo, 2017). Considering that conversation as an interface can be regarded as the most natural way for humans to interact with technology, the new affordances introduced to serve as a paradigm shift from the other, types of user interfaces, such as the command-line interface and the graphical user interface (GUI).

A command line interface (CLI) enables users to type syntax-specific commands in a terminal or console window to interact with an operating system. On the other hand, a GUI leverages graphics to provide an easy-to-use and navigate interface to a program. Along with a mouse, a keyboard, or other input devices, a GUI provides windows, pull-down menus, buttons, scrollbars, iconic images, wizards, and other icons to enable users to interact with the operating system or software application (Oracle, 2018). A graphical user interface provides the ‘space’ (a part of a device or software), in which the interaction between a human being and a machine can take place.

In contrast to the above interface types, a conversational interface does not require the user to have an exact understanding of the system affordances and is based on a more direct, natural and ill-structured type of human-computer interaction (“The Dawn of the Conversational Interface,” 2016). In short, it appears that with the rise of this new generation of interfaces, software developers attempt to move from asking users to learn how to interact with computers to building software that learns how to interact with users (Serban et al., 2017). The conversational interfaces share some similarities with the command line interface, such as their simplicity as they are both text-based mediums, but at the same time they differ significantly because of the fact that rather than requiring the user to remember and recall exact commands, they can interact with a machine simply by using natural language, i.e. the language they use in their everyday life whenever they are talking to a person.

Younger users grew up with computers and are much more computer savvy than older users. As a result, younger users are more comfortable with text messaging and communicating with computers by using commands, which have long been favored by power users over slower GUIs. Consumers, especially affluent consumers, are outsourcing their "chores," such as driving, shopping, cleaning, food delivery, and errands, giving rise to the "gig economy" and the need for more customer service.

In essence, a conversational interface may provide software creators with a new convenient channel for interacting and communicating with their users in a more personalized and engaging manner. This opens up interesting new implementations of bots in various scenarios when integrated with various enterprise applications. Indeed, the increased level of engagement in conversational agent platforms holds tremendous potential for enterprises and organizations in a lot of sectors, such as healthcare, finance, and education.

## 2.2 Existing Conversational Agent Approaches

In this section we:

a. Reflect on the history of the conversational agent approaches starting from the first chatbot that was developed
b. Present the most used conversational agents by examining their technological characteristics, their domain of usage and some screenshots from the user interface they use.

Typically, a conversational agent or chatbot is an automated system or a computer program that usually aims to imitate human conversations through voice commands, text chats or both (Radziwill & Benton, 2016; Tegos & Demetriadis, 2017). Most of the times, such a virtual entity is created to participate in a virtual conversation with humans, which usually takes place in an online environment. A conversational agent usually operates within a well-defined set of rules and parameters that shape its behavior. Such an agent can be designed to provide many functions in real-world settings and can be used in websites or messaging platforms and mobile applications (Brandtzaeg, 2017).

The conversational agent technology was not conceptualized recently. It has a long history. The first chatbot was developed by MIT professor Joseph Weizenbaum in the 1960s and was called ELIZA (Man, 1966). It passed the words that users entered into a computer and then paired them to a list of possible scripted responses. The same goes for voice-based conversational agents or "voice assistants" such as Alexa, Google Assistant, Siri or Cortana. IBM's Shoebox was the first voice assistant developed back in the 1960s (D. Vijayakumar, S.Manikandan, & K.Parthasarathy, 2018). It formed the foundation of the modern voice assistant era.

The primary reason why conversational agents and voice assistants have drawn so much attention recently is that the technology involved in NLP has improved substantially and became a lot easier and cheaper to access. Indeed, the rapid increase in interest in conversational agents in the last years illustrated by the fact that conversational agents appear in many mainstream applications. This rapid adoption is primarily due to the fact that the creation of effective chatbots has become far easier, thanks to techniques like natural language and syntax parsing and machine learning. Additionally, the integration with cloud-based applications is now easier than ever increasing the scalability of conversational agent platforms, which are usually considered as resource-intensive systems. Along with cloud-based applications, chatbots have begun to be used in almost any messaging application allowing users to retrieve information faster and easier.

The rise of conversational agents can be regarded as a result of the changing user behavior in the post-app mobile world and the maturing of key enabling technologies. After nearly a decade of explosive growth, mobile apps have largely stopped growing. And as the Wall Street Journal says in "What Comes After Apps," smartphone users have reached their limits on how many apps they want to install and open on a daily basis.

Users are spending an increasing amount of time in just a few apps. And social and messaging apps emerge as big winners (Fu et al., 2016). Users do not like to exit their messaging app just to glance at small snippets of information (for example, to check the weather, stock prices, or look up a restaurant or map).

Artificial Intelligence has gotten a lot better in the past couple of years in understanding what the user wants in natural language. The API economy has matured to the point where many useful services can be accessed from computers to complete real-world tasks.

In short, chatbots are used in human users' preferred environments (messaging apps), converse with users in natural language, understand what users want, and do the humans' bidding through a large

network of connected services. Chatbots have many different forms of use. They can be used as a questioning-answering computer system, as a multiple choice menu to route users for a more personalized experience, as voice-based bots or chat-based bots that humans can chat with, incorporated into mobile apps and wearable devices with a voice interface, as home automation with voice commands and many other applications

### 2.2.1 IBM Watson Conversation Service

IBM Watson was developed as a part of IBM's DeepQA project. Watson constituted a question-answering computer system capable of answering questions posed in natural language. Nowadays, Watson extends its capabilities adding characteristics such as reading, talking, learning, and recommending and it is not only used as a question - answering system. This conversation service supports multiple channels for users to interact with such as text, speech, and image. Watson uses Artificial Intelligence that understands users' intentions and further trains the assistant.

Users can build conversational interfaces by using Watson assistant into any website and web application, mobile application, messaging channels, and customer service tools in 13 different languages. It supports a wide collection of SDKs like Node, Java, Python, Swift, iOS, Android Unity, .NET, Ruby among others, to solve complex problems. Numerous industries use Watson as their main conversation service. Those industries include advertising, customer engagement, education, financial services, health, IoT, media, talent, and work. You can refer to the image below to see a Bank bot implementation utilizing Watson.

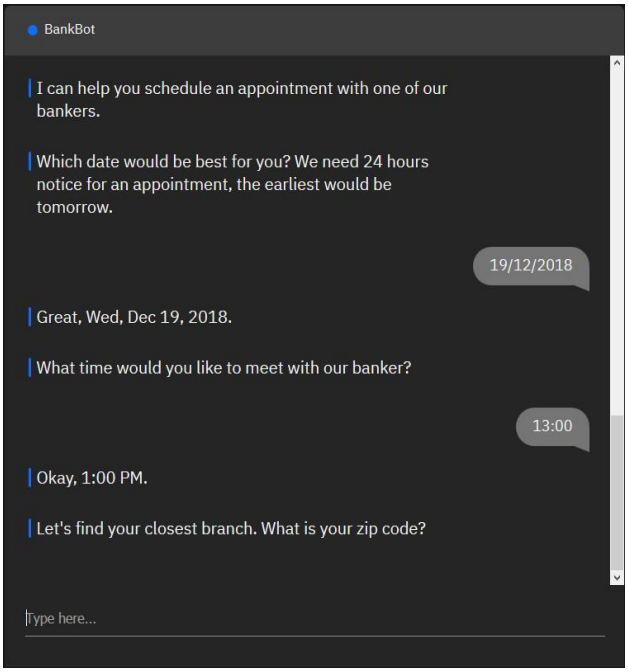


**Figure 1**: A BankBot implementation utilizing Watson.

*Retrieved from IBM, 2018, https://www.ibm.com/cloud/watson-assistant/features/*

Chatbot designers can implement their assistant through three main components:

- Intents, which are the goals that users of the chatbot might have and they will probably try to solve through chatbot

- Entities, which are terms that provide context for an intent
- Dialog, which is the flow that incorporates intents and entities

### 2.2.2 Agentbot

Agentbot is a product of Aivo Family, which provide customer service solutions powered by Artificial Intelligence. Agentbot uses neural networks to understand users' intent and identify relevant meaning from their questions or comments, even if users use informal expressions, jargon, phrases with grammatical errors or other language deviations.

Developers can incorporate agentbot by using several channels, included web and web mobile, mobile apps, Facebook messenger, CRM platforms, SMS and messaging API, and third party applications via integrations. Agentbot supports three languages, English, Spanish, and Portuguese and can be used in telecommunications and cable operators, e-commerce and online services, banking and financial services, and government industries.

Agentbot incorporates feedback mechanism in each response so as to learn from users' ratings to identify improvement opportunities. Also, Agentbot provides a management tool to create, edit, and delete questions without requiring technical knowledge from the user.

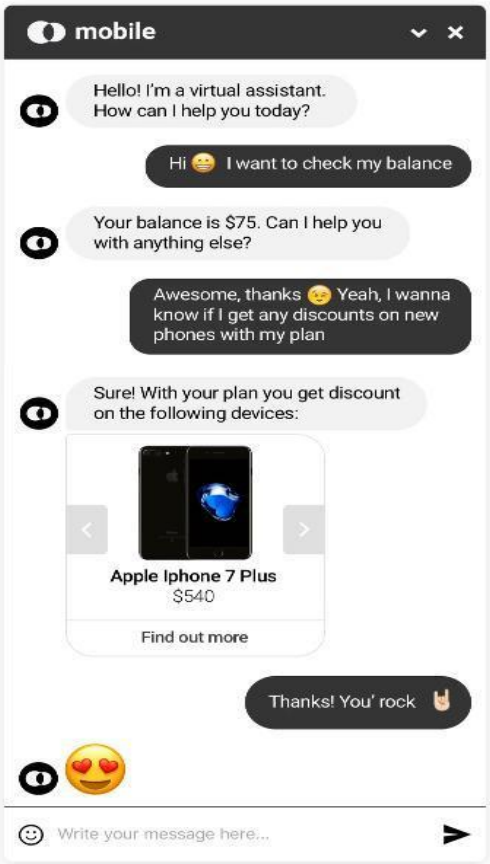


**Figure 2**: A chatbot implementation utilizing Agentbot.

*Retrieved from agentbot, 2018, https://www.aivo.co/en/agentbot-automatic-support-chat/*

### 2.2.3 WIT

With wit.ai a developer can create text or voice-based bots that humans can chat with, mobile apps and wearable devices with a voice interface, and home automation with voice commands. It is owned by Facebook, is free to use and open-source, and they claim that wit.ai is used by over 180.000 developers.

Developers can incorporate wit.ai into their projects by using libraries with Ruby and Python or SDK for Node.js, or they can integrate wit.ai with the Facebook messenger or their website. Also, wit.ai is available for developers to use with iOS, Android, Windows phone, and Raspberry. Wit.ai supports 50 languages. To understand users' input, wit.ai uses natural language processing (NLP).


**Figure 3**: A high level depiction of the utilization of wit.ai at a home automation application

*Retrieved from Wit.ai, 2018, https://labs.wit.ai/demo/index.html*

### 2.2.4 Rasa

Rasa is a machine learning framework for developers to create chatbots and assistants. It is an open source project with support of over 1000 community members. Rasa NLU has been used successfully in insurance, banking, healthcare, travel and transport, and telecom industries.

Rasa handles contextual conversations with machine learning to identify users' intention and extract entities. In order to understand users' intention, rasa supports dozens of natural languages.

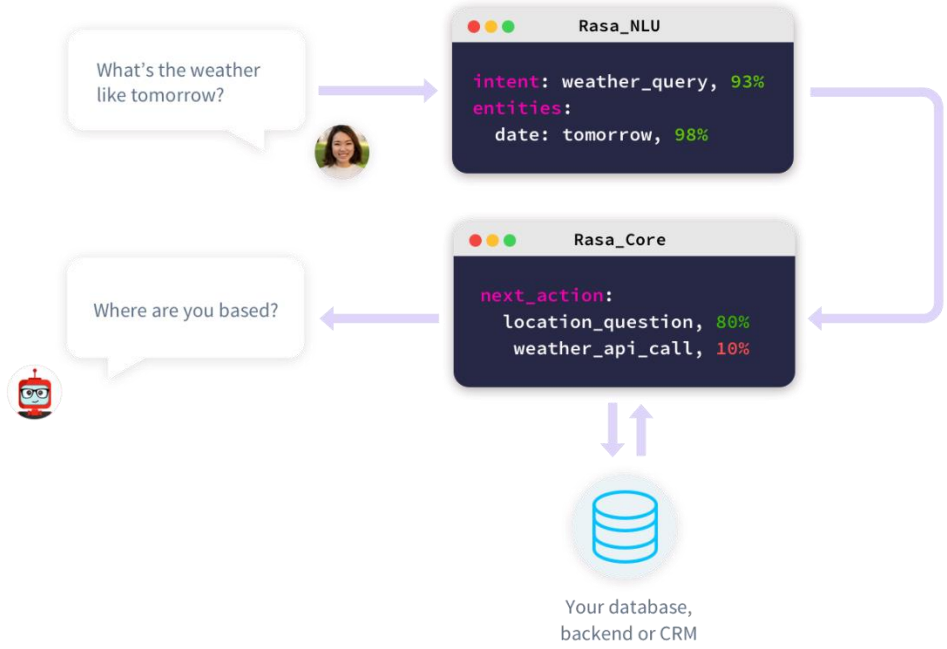


**Figure 4**: An example of Rasa stack usage,

*Retrieved from Rasa, 2018, https://legacy-docs.rasa.com/docs/core/0.8.6/motivation/*

The Rasa stack consists of the rasa NLU and rasa core libraries. Rasa NLU performs the natural language understanding by converting text to structured data. For example, NLU can take the sentence "I am looking for a Mexican restaurant in the center of town" and return structured data like:

- intent: search_restaurant
- entities:
    - cuisine: Mexican
    - location: center

Rasa core performs dialog management by auditing the conversation and deciding how to act.

Developers can connect with rasa NLU through APIs to bring chatbot conversations into their applications. Connectors can be used to embed rasa into messaging channels like Facebook, slack, telegram, or a custom made one.

### 2.2.5 Dialogflow

Dialogflow (formerly api.ai) is owned by Google. By using this tool, developers can build voice and text-based conversational interfaces, powered by artificial intelligence. Those interfaces can be further embedded into websites, mobile applications, Facebook Messenger, and other popular devices and platforms. Dialogflow is widely used in more than 400M Google Assistant devices.

Just like the other tools we referred previously, dialogflow uses machine learning to understand users' dialogues, to analyze their intents, and to respond in the most useful way. The basic flow of conversation involves:

- A user enters input to the system
- The system parses that input

- The system responds to the user

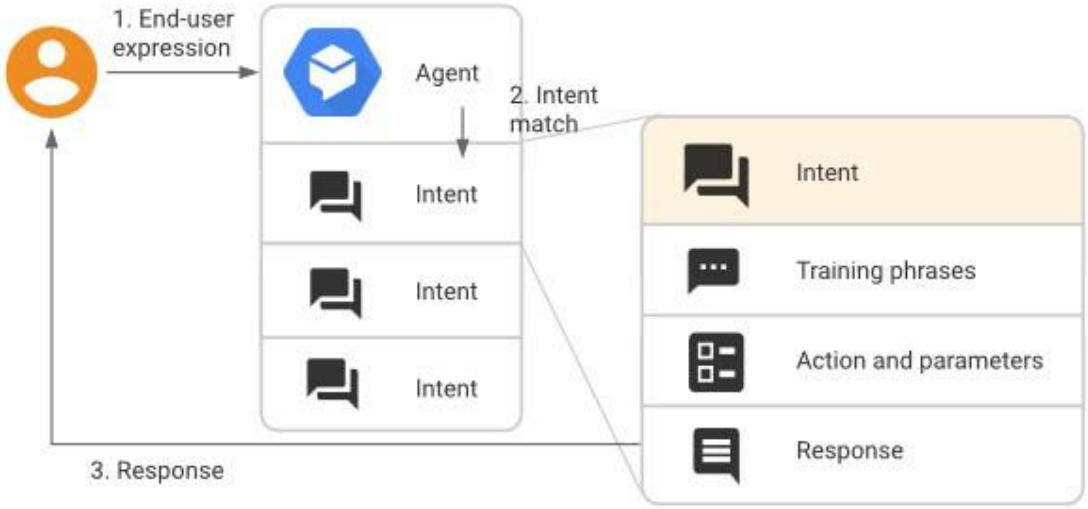


**Figure 5**: The sequence of events in Dialogflow after an utterance from a user

*Retrieved from Google, 2018, https://cloud.google.com/dialogflow/docs/basics*

For each intent, developers can define phrases that trigger the intent (training phrases), what to extract from the utterance and the response of the agent back to the user.

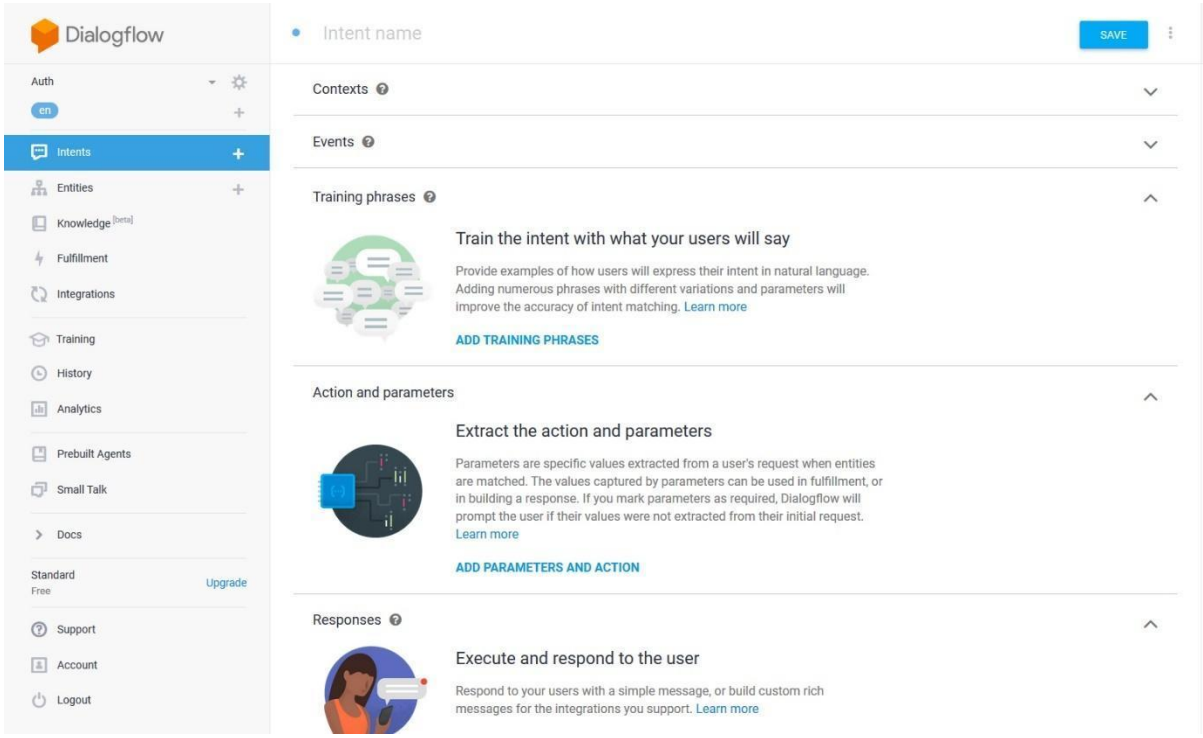


**Figure 6**: Each intent in Dialogflow includes training phrases, actions, and responses

Dialogflow can be embedded in developers' applications or websites by a large number of SDKs and libraries which helps to change agent's behavior by allowing developers to create, read, update and delete intents, entities, and contexts. There are available libraries for Node.js, Python, Java, Go, Ruby, C#, and PHP and SDKs such as Android, Botkit, C++, Cordova, iOS, Java, Javascript, .NET, Node.js, Python, Ruby, Unity, and Xamarin.

### 2.2.6 Microsoft Language Understanding Intelligent Service (LUIS)

With Microsoft language understanding intelligent machine learning-based service, developers can add natural language understanding into their applications. Any conversational application could become a client for LUIS, such as social media apps, chatbots, and speech-enabled desktop applications.

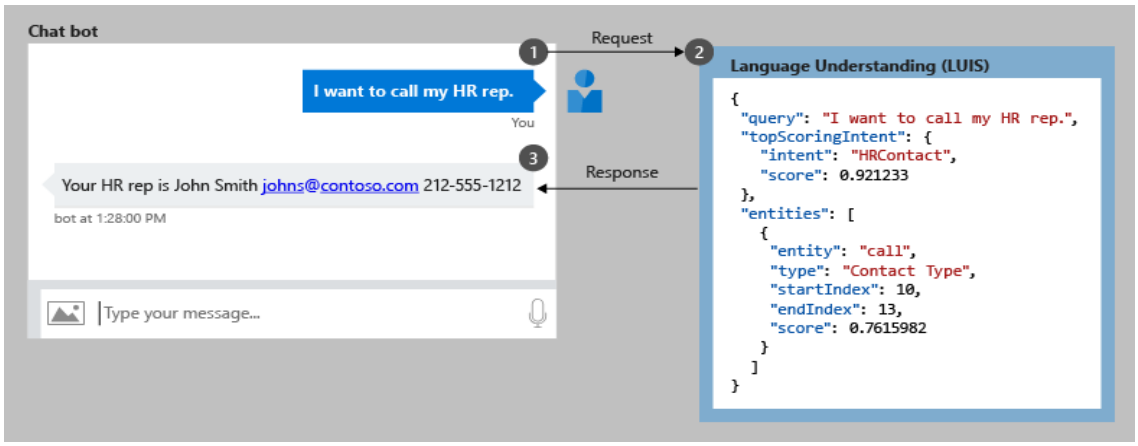


**Figure 7**: A request from a user and the response from a chatbot based on LUIS

*Retrieved from Microsoft, 2018, https://docs.microsoft.com/en-us/azure/cognitive-services/luis/what-is-luis*

LUIS works as follow:

- A user uses a conversational application to ask a question
- The conversational application sends a request to LUIS
- LUIS tries to analyze the content of the message by finding user' intent and entities.
- LUIS sends back to the conversational application a response as a JSON file
- The conversational application uses the JSON file to make decisions about the reaction to the user's request. There are numerous of rest APIs to extract the intent from the response in C#, Go, Java, Node.js, and Python.

Luis can be used with:

- Information chatbots, that answers questions

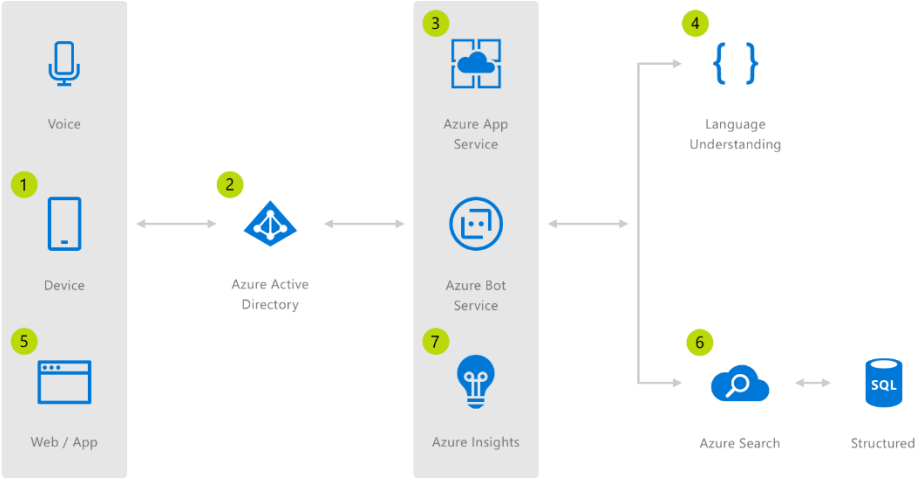


**Figure 8**: The architecture of an information chatbot using LUIS

*Retrieved from Luis.ai, 2019, https://www.luis.ai/home*

- Commerce chatbots. Together with the Azure Bot, developers can create conversational interfaces for various scenarios. With Azure Bot Microsoft service, developers can build intelligent bots to interact naturally with the clients on websites, applications, Cortana, Microsoft Teams, Skype, Slack, Facebook Messenger, and more.

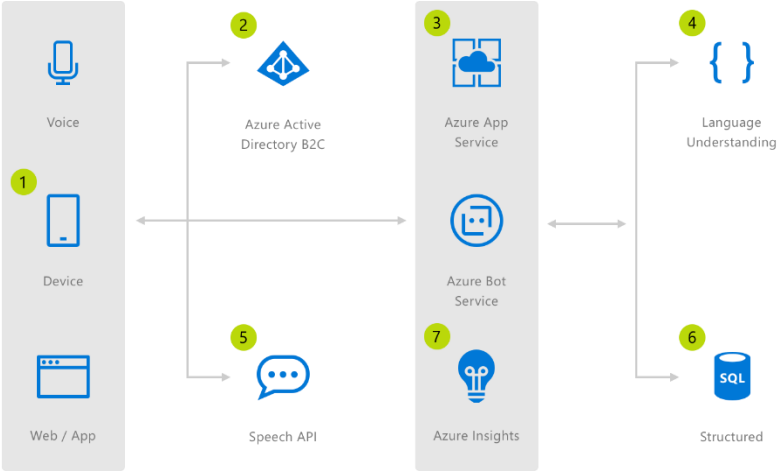


**Figure 9**: The architecture of a commerce chatbot using LUIS

*Retrieved from Luis.ai, 2019, https://www.luis.ai/home*

- IoT Devices. Developers can create conversational interfaces to interact with all of the internet-accessible devices such as television or fridge in order to translate commands into actions.

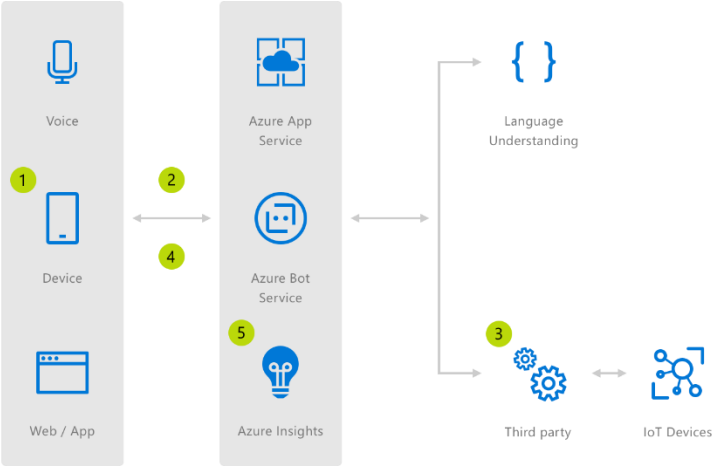


**Figure 10**: The architecture of a conversational interface for IoT devices using LUIS

*Retrieved from Luis.ai, 2019, https://www.luis.ai/home*

LUIS uses a language detection API which evaluates text input and returns a language identifier. Text analysis recognizes up to 120 languages.

### 2.2.7 Chatfuel

Chatfuel is a bot platform that powers more than 46% of all the Facebook Messenger bots and supports 97 languages. The basic components of every chatbot are the "cards" and the "blocks". Cards are the most basic building component of the bots. Cards provide all the information to users which can be of a form of text, image, gallery, and plugin.

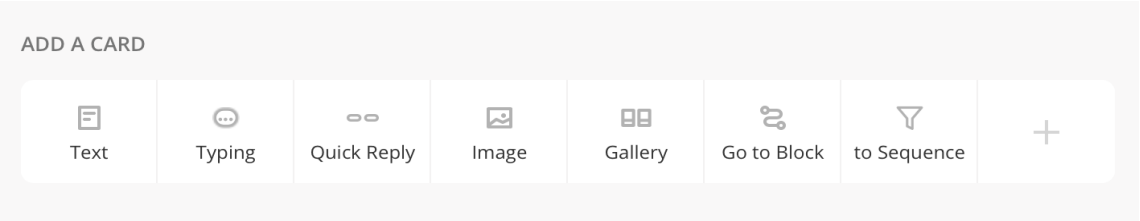


**Figure 11**: Chatfuel platform uses cards as a basic component for every chatbot

A block is also a basic building component of every bot. It consists of one or more message cards that are sent together to a bot user. Developers can link blocks with each other using buttons in text cards or in gallery cards.

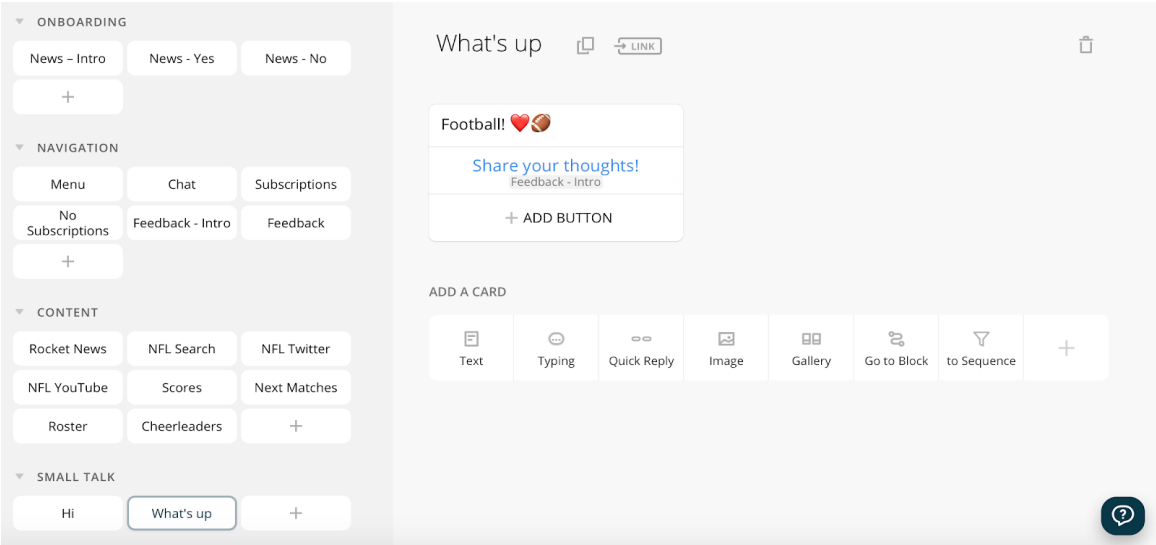


**Figure 12**: Chatfuel platform uses cards as a basic component for every chatbot

With plugins, bot developers can expand bot's usability by adding extra features such as route users for a more personalized experience, collect input from users, display to users target content from the website, play video or audio messages directly into the bot and many more.

Chatfuel also uses an artificial intelligence engine that detects user phrases similar to ones predefined by the bot developer to show a relevant message to the user.

### 2.2.8 Pandorabots

Pandorabots is another leading company which deals with artificial intelligence that provides a web platform for developers to build and deploy chatbots. Pandorabots supports the Artificial Intelligence Markup Language (AIML) to create natural language software agents. One of the advantages of AIML usage is that AIML can be written in almost any natural language. They claim that more than 300000 chatbots have been created by using their platform.

Pandorabots provides API access to its chatbot hosting platform and offers Java, Ruby, Go, PHP, Python, and Node.js SDKs. Through integrations, developers can publish chatbots on third-party channels such as Skype, slack, messenger, twillio, twitter and many more.

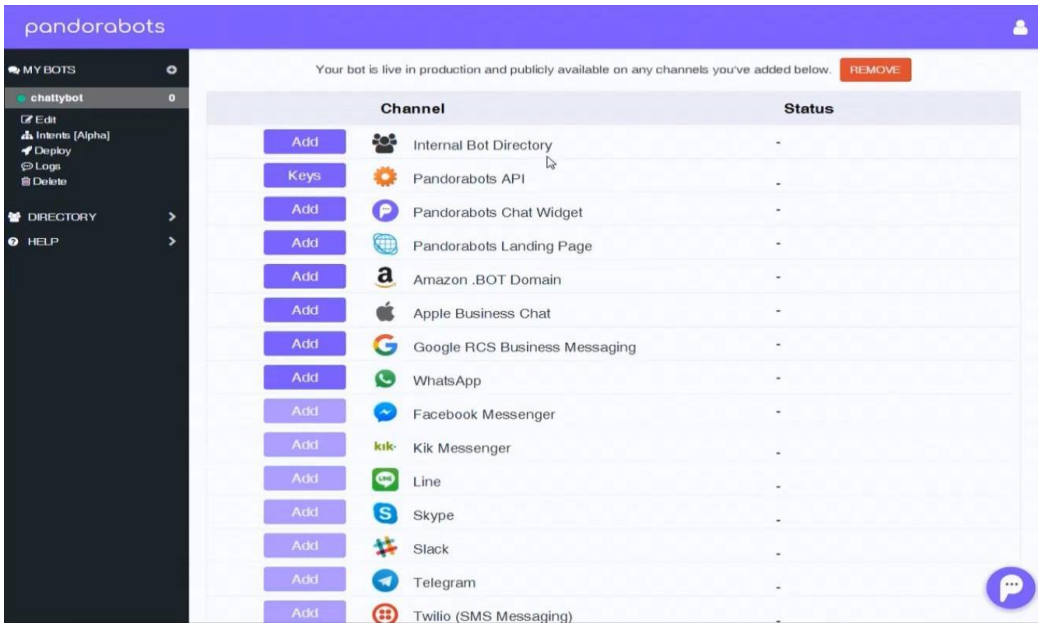


**Figure 13**: A list of the channels that can be connected with Pandorabots

### 2.2.9 Chatterbot

Unlike the previous bot platforms, chatterbot is a Python library that helps bot developers to generate responses to user's input. Chatterbot can power applications developed with Django because it has direct support for integrations with Django framework. Chatterbot can be trained to speak any language.

Developers need to create a new bot instance following by training procedures. During the training process, bot developers load example dialogs into bot's database. Also, each time a user enters a statement, the text they enter and the response to user's input are also stored into the library. During a dialogue, the bot selects the closest matching response by searching into the library and then chooses a response from the selection of known responses to that statement.

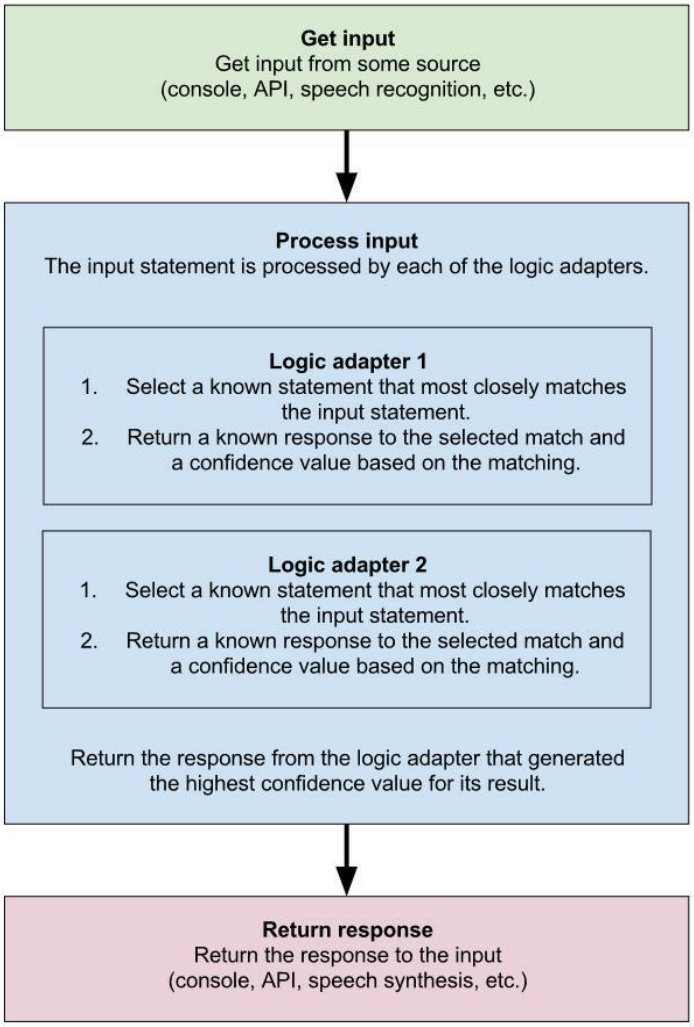


**Figure 14**: A depiction of how chatterbot generate responses based on input

*Retrieved from Chatterbot, 2018, https://chatterbot.readthedocs.io/en/stable/*

### 2.2.10 Octane.ai

Octane AI chatbots help businesses, brands and individuals with tasks such as customer support, showing off content, showcasing merchandise, answering frequently asked questions, letting customers subscribe to notifications and more. It is actually a Facebook messenger bot that can:

- answer to questions

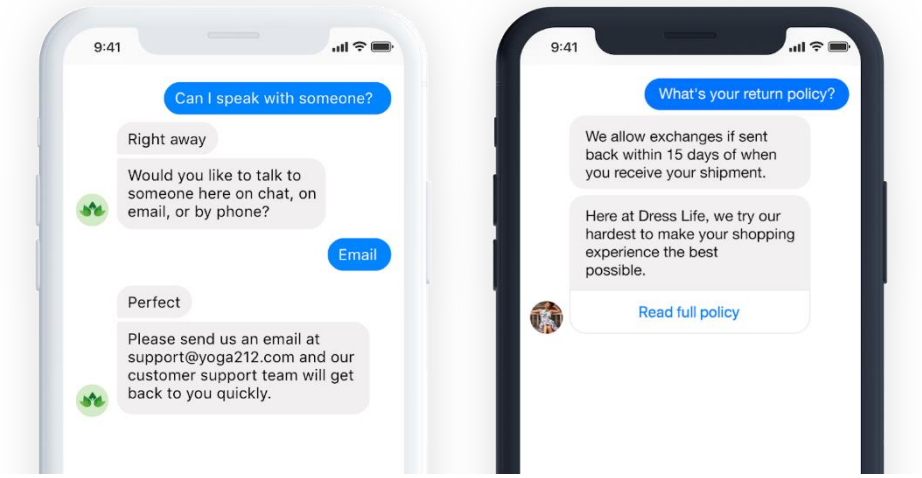


**Figure 15**: Octave.ai bot answers a question

*Retrieved from Octane.ai, 2018, https://join.octaneai.com/platform/answer-questions*

- allow users to subscribe to Messenger for exclusive discounts, notifications etc through a popup window

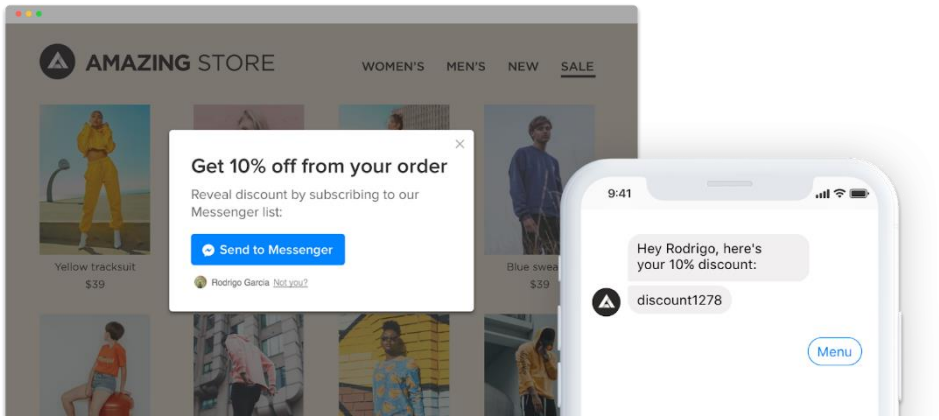


**Figure 16**: Octave.ai bot shows a dialog box for user subscription

*Retrieved from Octane.ai, 2018, https://join.octaneai.com/platform/facebook-messenger-pop-ups*

- send personalized Facebook messages to customers that visited your Shopify store

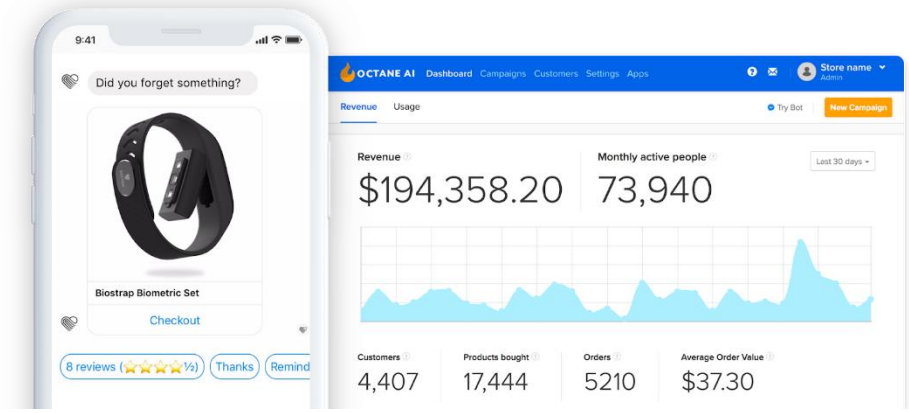


**Figure 17**: Octave.ai bot sends personalized messages to customers

*Retrieved from Octane.ai, 2018, https://join.octaneai.com/platform/abandoned-cart-recovery*

- send custom follow-up messages


**Figure 18**: Octave.ai bot sends follow up messages

*Retrieved from Octane.ai, 2018, https://join.octaneai.com/platform/followup-messages*

### 2.2.11 Reply.ai

Reply.ai is another conversational agent that answers questions made by users. Reply.ai uses machine learning and natural language processing to understand user's input. Reply.ai is a complete solution as it comes with a built-in CRM and customizable, real-time analytics dashboard.

Reply.ai supports Facebook Messenger, Kik, Telegram, LINE, SMS or your own chat screen. They use native custom UI elements in each channel. A web widget is also available.

Developers can use reply.ai in various industries. In the insurance industry, they can add a conversational layer to the entire customer journey to answer general insurance related questions, integrate an FAQ bot into call centers, and gather data from the customer without a broker. Also, in the food and beverage industry, voice-activated bots can be used in conjunction with Amazon

Alexa to provide replies to the user's input. Reply.ai also used in the consumer electronics industry for onboarding new users.

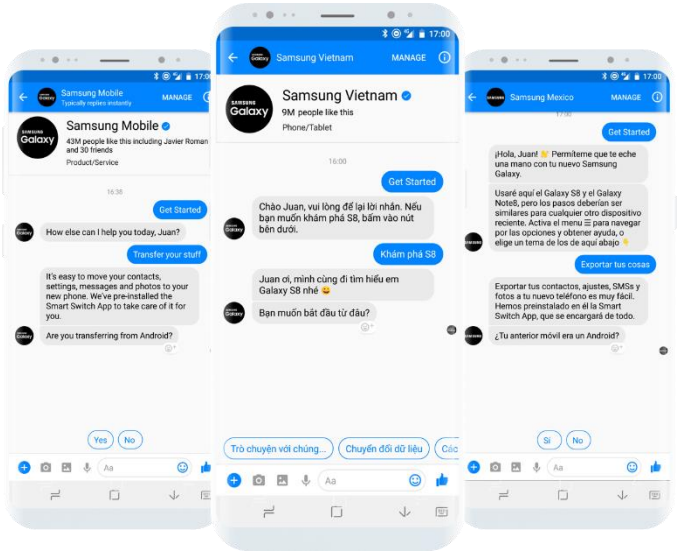


**Figure 19**: An implementation of a chatbot utilizing Reply.ai

*Retrieved from Reply.ai, 2018, https://www.reply.ai/industry/ electronics/*

In the travel and hospitality industry, developers can add a bot to answer to user's questions regarding the booking process.

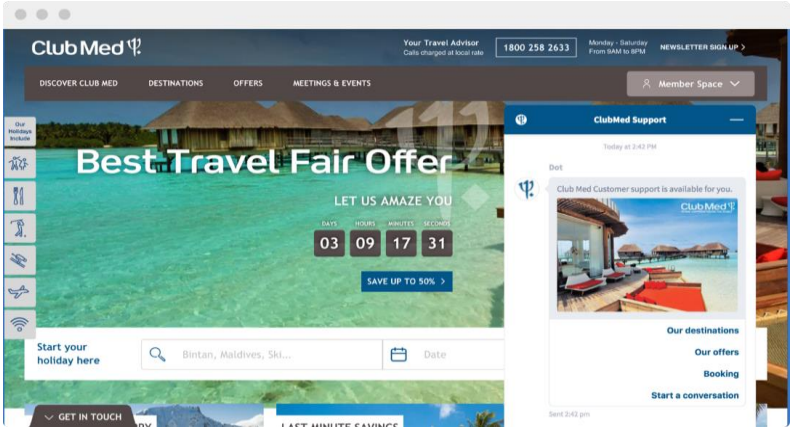


**Figure 20**: The use of a chatbot in the travel industry

*Retrieved from Reply.ai, 2018, https://www.reply.ai/industry/travel/*

### 2.2.12 ChatScript

ChatScript is a rule-based chatbot engine where rules are created by the developers in a process that is called scripting. ChatScript combines natural language engine, which constitutes the basis for natural language company for a variety of tech startups, and a dialog management system. The engine is open source on Github.

```
topic: ~food []

#! I like spinach
s: ( I like spinach ) Are you a fan of the Popeye cartoons?

        a: ( ~yes )  I used to watch him as a child. Did you lust after Olive Oyl?
            b: ( ~no ) Me neither. She was too skinny.
            b: ( yes ) You probably like skinny models.

        a: ( ~no ) What cartoons do you watch?
                b: ( none ) You lead a deprived life.
                b: ( Mickey Mouse ) The Disney icon.

#! I often eat chicken
u: ( ![ not never rarely ] I * ~ingest * ~meat ) You eat meat.

#! I really love chicken
u: ( !~negativeWords I * ~like * ~meat ) You like meat.
```

**Figure 21**: A depiction of a script created in ChatScript engine

*Retrieved from Github, 2018, https://github.com/ChatScript/ChatScript*

### 2.2.13 Conclusions

Concluding the above analysis we emphasize the following:

- There are a lot of conversational agent approaches available for developers to incorporate them into their projects.
- Many industries use agents such as insurance, banking, healthcare, travel, transport, telecommunications, advertising, customer engagement, education, health, media and many more.
- Most of the usage concern questioning and answering system, voice commands and general assistance to users.
- Some of the conversational agent approaches try to identify users' intent by using Artificial intelligence technology to adapt to the user's objective.
- There are a lot of different application approaches, regarding social media, conversational applications such as Skype and Slack as libraries that can be used by bot developers.
- There is limited reference to conversational agents application to education and most of the chatbot implementations include conversations between a user and a chatbot.
- All platforms provide tools for the developer to built a chatbot/agent but do not address the perspective of providing some working environment for the end-users themselves to build their chatbot/agent.

## 2.3 Conversational Agents: the Potential for Education

In this section we:

a. Analyze the existence of the pedagogical agents, the application of agents on education, and a subgroup of these agents, the conversational agents,
b. Present one-to-one conversational agent solutions where tutorial dialogues occur between a student and an agent,
c. Introduce agents that support groups in order to trigger productive forms of peer dialogue and scaffold students' learning in a CSCL context, by providing implementation examples.

In the previous section, we presented an extensive body of evidence suggesting that conversational agents are becoming a trend in multiple fields such as healthcare, marketing, management, tourism, advertising, customer engagement, financial services, IoT, media, talent, and work. Up until now the implementation of conversational agents in education has been scarce.

In the field of technology-enhanced learning, conversational agents have their roots on pedagogical agents, which have a long history in the field. A pedagogical agent can be regarded as an autonomous computer-generated virtual character, aiming to fulfill specific pedagogical purposes in a learning environment (Gulz, Haake, Silvervarg, Sjödén, & Veletsianos, 2011). Numerous technological advancements in the visual embodiment and computational linguistics areas have led to the development of the social and personal pedagogical agents used nowadays (Gulz et al., 2011). Agents have evolved into modern virtual characters that are able to exhibit social skills and interact with learners on a broad range of topics apart from tutoring (Veletsianos & Russell, 2014). Indeed, many pedagogical agents have been developed over the years in order to serve a wide variety of instructional roles, such as expert, motivator or mentor (Baylor & Kim, 2005). Pedagogical agents have been repeatedly used as part of intelligent tutoring systems in order to provide customized instruction or feedback to the learners in various learning scenarios (Baker, 2016; Kim, Baylor, & Shen, 2007). A pedagogical agent can often interact with a learner using multiple communication channels, such as text (Chaudhuri, Kumar, Howley, & Rosé, 2009) or speech (Wik, & Hjalmarsson, 2009).

However, with the latest innovations in computational technologies and the promising possibilities for integrating discourse in educational systems, a new subcategory of pedagogical agents has emerged, called “conversational agents”. This type of agents mainly aim to engage learners in a conversation through natural language (Gulz et al., 2011). Over the past, conversational agents have been used to accomplish a variety of educational goals such as tutoring (Heffernan & Croteau, 2004; VanLehn et al., 2007), question-answering (Feng, Shaw, Kim, & Hovy, 2006), language learning practice (Griol, Baena, Molina, & de Miguel, 2014; Wik & Hjalmarsson, 2009), and the promotion of health-related behavioral changes (Kennedy et al., 2012) and of metacognitive skills (Kerly, Ellis, & Bull, 2008).

Conversational agents are usually regarded as a subgroup of pedagogical agents involving learners in natural language interactions (Kerly, Ellis, & Bull, 2009). Research has shown that using conversational agents to engage learners in one-to-one (student-agent) tutorial dialogues can improve students’ comprehension and foster students’ engagement and motivation (Veletsianos & Russell, 2014). Conversational agents appear to have a positive impact on students’ satisfaction as well as learning outcomes (Kerly, Hall, & Bull, 2007; Huang, Lee, Kwon, & Kim, 2017).

### 2.3.1 Agents in one-on-one learning settings

In their study, VanLehn et al. (2007) have revealed that a tutorial dialogue initiated and controlled by a conversational agent can provide several benefits over a monologue, such as the detection and remediation of failed communication, the correction of inaccurate student knowledge and increased interactivity. Many conversational agents were created in a similar fashion, aiming to engage learners in student-agent tutorial dialogues in one-to-one settings (e.g., Rus, D’Mello, Hu, & Graesser, 2013). Those type of agents leverage Artificial Intelligence methods in order to simulate the behavior of a human expert and be able to converse with the learners on a series of predefined topics.

This can be really useful for providing individual learning support. Especially in large-scale learning scenarios, such as in a university or massive open online courses (MOOCs), agents appear to may compensate the insufficient individual support of instructors, which constitutes one of the key factors negatively affecting retention rates (Hone & El Said, 2016). A study by the University of Georgia Tech focused on the utilization of a conversational agent that attempted to handle forum

posts by learners enrolled in a computer science course (Goel, Creeden, Kumble, Salunke, Shetty, & Wiltgen, 2015). The agent was developed with IBM Watson bot development platform and resulted in increased students engagement.

Another study by the Universidad Carlos III de Madrid has explored the usage of conversational agents operating as a voice assistant supporting MOOCs (C.C. Aquirre et al., 2018). In the context of a Java-related MOOC, the study focused on a prototype conversational agent, which could assist students with their course tests and providing feedback based on their weaknesses. Students were able to answer to agent's messages using voice, which received positive students' comments.

Conversational agents can be promising tools since they can be used as learning partners and offer continuing feedback to both learners and instructors in educational environments. There are a lot of successful stories (e.g., ReTuDiS, AutoTutor, ITSPOKE) regarding the beneficial use of such agents in individual learning settings. The findings revealed that the utilization of conversational agents can lead to significant learning as well as memory gains, enhance self-assessment skills and increase motivation (Kerly et al., 2008; VanLehn et al., 2007).

In their recent review analyzing the potential of chatbots in education Winkler and Söllner (2018) identify five major issues in the field (numbered next as questions, 'Qx') and provide respective answers ('Ax') based on available literature, as follows:

- Q1: *How do individual differences of students affect chatbot-mediated learning (CML) processes and CML outcomes?*
  - A1: People interact in different ways with chatbots. Research indicates that personality traits, educational background, and self-regulated skills of students may have an impact on CML process quality. It might be advisable that chatbots should be tailored to the individual differences of students. However, it is important to consider that currently, no study analyzed whether personalized chatbots have a significant impact on learning outcomes.
- Q2: *In which educational settings are chatbots applied?*
  - A2: There are many chatbot applications in the health and well-being domains, where chatbots are used to support students in their medical education or even patients with their therapy.
  - Language learning is another area where chatbots are being developed to help students communicate freely and without mistake anxiety in the language, they learn.
  - Additionally, chatbots seem to have an impact on the development of students' metacognitive skills, by providing formative feedback to students during their learning.
  - Also, chatbots may deeper engage students in learning by asking challenging questions which is the main driver of intrinsic motivation (Oudeyer, Gottlieb, & Lopes, 2016). The review, however, emphasizes that additional studies regarding a systematic method for evaluating chatbots in education are necessary.
- Q3: *What approaches are being used to build and design a chatbot in learning settings and how does that influence CML processes and CML outcomes?*
  - A3: The review reports that there are at least three different dimensions which can be used for classifying the chatbot build and design approaches, namely:
    - 1) *Retrieval-based or generative*: whether the chatbot natural language understanding is based on predefined rules (retrieval-based) or another type of analysis (usually based on machine learning algorithms) that enable the chatbot to produce an original response.
    - 2) *Use speech or/and text as input mode*: whether the chatbot processes only text input or has also speech recognition capabilities.

  - 3) *Integrate or not context-data*: whether the chatbots possess (or not) capabilities of modeling some type of contextual data (such as, for example, the emotional state of the students) to increase the appropriateness of their response.
- Q4: *What are determinants of CML process quality and how do they relate to chatbot design and CML learning outcomes?*
- A4: The review indicates that typically students tend to use shorter messages and poorer vocabulary to communicate with chatbots and the makes it harder for chatbots to understand student' intents. Therefore, developers should consider this difference in way of communication to improve CML processes and learning outcomes.
- Furthermore, the way how students implement chatbots during their learning process is crucial. For example, the CML process quality is low when students do not recognize the higher value of chatbots during learning immediately. This may happen when lecturers do not explain how to best use chatbots, in which case chatbots might stay unsuccessful (Söllner et al., 2017).
- The review emphasizes that available studies mostly explore the creation of a codependent and intelligent relationship between teacher and chatbot, utilizing both for their strengths and deliver the best student experience (Hughes, 2018). It is important, therefore, for the future to further explore how and where human or chatbot assistance is preferable.
- Q5: *How can different kinds of learning outcomes be influenced by chatbot design, individual differences of students and CML process quality?*
- A5: From the available research outcomes it seems possible that chatbots may be used to support different kinds of learning outcomes such as affective, cognitive and metacognitive. It is also advisable that chatbots include context information, for example, detecting knowledge gaps of students thus increasing domain-relevant learning outcomes.
- Some researchers also suggest (e.g. Duffy & Azevedo, 2015) that chatbots should be used as metacognitive tools, empowering students to actively control and regulate their learning process.
- Finally it is suggested that chatbots should be easily accessible and respond fast, while their visualization seems to be controversial; some studies support the view that visualizing and building up a personality of the chatbot has an impact (Berry et al., 2005), while others reject this conclusion (Ben Mimoun et al., 2015).

### 2.3.2 Agents to support groups

Although the main research interest of the past focused on the creation of agents acting in one-to-one learning settings, researchers also turned their attention to the design of conversational agents that support groups of learners instead of individuals (e.g., Kumar & Rosé, 2011; Walker, Rummel, & Koedinger, 2011). Indeed, several research groups explored the utilization of such agents as a means to trigger productive forms of peer dialogue and scaffold students' learning in a CSCL context.

A series of studies explored the impact of conversational agents on the quality of peer dialogue and, consequently, on both group and individual learning outcomes. An agent engaging learners in directed lines of reasoning, i.e., a prescribed reasoning path, was found to enhance learning performance and increase the conceptual depth of students' conversations (Chaudhuri et al., 2009). Another study explored the usage of a conversational agent displaying reflective prompts, during a reciprocal peer tutoring scenario. This time it was found that the adaptive support provided by the agent can increase the conceptual content of students' utterances (Walker et al., 2011).

Nevertheless, some criticism was expressed as well in some studies. It was found that, on some occasions, students may not interact fruitfully with conversational agents, not paying much attention to their messages, providing oversimplified responses or even ignoring them. Another point of criticism emphasized that there are often no possibilities of customization on the part of

the implementing teachers. Lastly, it was shown that, unfortunately, most agents tend to specialize to a specific instructional domain, being unable to operate in a variety of learning contexts without having increased development and maintenance-related costs (Dyke, Adamson, Howley, & Rosé, 2013).

Inspired by the work of the teachers' community, which explored methods to promote and support effective classroom discussions, another research direction that emerged explored the use of conversational agents as an agile and efficient tool for scaffolding productive peer discussions. This work focuses on the design of agent-based dynamic intervention deriving from the Academically Productive Talk (APT) discourse framework (Resnick, Michaels, & O'Connor, 2010).

As discussed in deliverable D1.2, the APT framework originates from a substantial body of work on modeling useful classroom discussion practices and norms. The APT framework emphasizes the key role of social interaction in inducing beneficial mental processes. In contrast to other popular classroom discourse frameworks, APT prioritizes reasoning over correctness and attempts to orchestrate student-centered discussions (Michaels, O'Connor, Hall, & Resnick, 2010). In this manner, it proposes a series of moves (see Table I), aiming to stimulate constructive forms of peer dialogue.

**Table 1**: Examples of APT-agent interventions

| APT Move | Example |
|---|---|
| 1. Add-on | [Student], would you like to add something to what your partner said about [Concept]? |
| 2. Agree-Disagree | [Student], what do you think of what [Partner] said about [Concept]? Do you agree or disagree? |
| 3. Verify-and-Clarify | [Student], do you agree with the following statement: [Concept1] [verb expression] [Concept2]? Why? |
| 4. Build-on-Prior-Knowledge | [Student], do you think [Concept1] is somehow related to [Concept2]? How? |

A research direction has emerged in the past few years investigating the usage of conversational agents that perform academically productive talk moves like teachers often do in collaborative activities taking place in classroom (e.g., Tegos, Demetriadis, Papadopoulos, & Weinberger, 2016; Adamson, Dyke, Jang, & Rose, 2014). Such APT-based agents typically display interventions that aim at eliciting student reasoning instead of providing content-specific explanations or instructional assistance. The few studies conducted in this area have demonstrated conversational agents displaying unsolicited APT moves can conceptually enrich students' discussions and positively impact collaborative learning outcomes (Tegos & Demetriadis, 2017; Tegos et al., 2016). The results of Tegos and Demetriadis were also confirmed by a recent study (Winkler et al., 2019), which highlighted that a group of learners interacting with a Smart Personal Assistant (SPA) tutor can provide better understanding towards solving a task as compared to a group of learners receiving assistance by a human tutor. In line with the aforementioned results, Dyke, Adamson, Howley, & Rosé (2012) and Adamson, Ashe, Jang, Yaron, & Rose (2013) studies revealed that APT-based agents can intensify the knowledge exchange among learning partners and increase students' explicit reasoning as well as participation in subsequent classroom discussions.


**Figure 22**: An example of an APT-based conversational agent system

With the recent rise in focus on MOOCs and the positive impact of agile conversational agents in collaborative learning settings, researchers also started recently to explore the usage of agent-based supportive mechanisms in the context of MOOCs (Rosé & Ferschke, 2016). As already described in D1.1 and D1.2, such agents can increase students' engagement, minimize dropout rates and amplify the tremendous support resources that the students can offer to each other by themselves (Fersche, Yang, Tomar, & Rose, 2015). Thus conversational agents that support small-group activities appear to have a direct application in MOOCs (Tomar, Sankaranarayanan, & Rosé, 2016).

Figure 23 depicts an implementation of an agent that supports groups of students in a chat room. Rosé & Ferschke created a chat based collaborative environment where students pair in dyads and enter a dedicated for the group room. In that room, students can collaborate and an agent intervene to assist them. The agent tracks events that occur in the chat room and decide when and how to intervene. On Figure 23 we can see some agent's interventions based on the events of users entering the chat room where agent reminds users to press the "We're Ready" button or type "ready" to begin.

However, a MOOC environment may present a series of practical challenges such as the coordination of participants, the connectivity issues and drop outs, the diverse learner populations and even the different time zones (Tomar et al., 2016; Tomar et al., 2017). In order to facilitate the formation of ad-hoc study groups for the chat activity, Ferschke et al. (2015) made use of a simple setup referred to as a Lobby. Students entered the Lobby with a simple, clearly labeled button integrated with the edX platform. Upon entering the Lobby, students were asked to enter a username that would be displayed in the chat. Once registered in the Lobby, the student waited to be matched with another participant. If the student was successfully matched with another learner who arrived at the Lobby within a couple of minutes to interact with, he and his partner were then presented with a link to click on to enter a chat room created for them in real time. Otherwise they were requested to come back later. Some students needed to make up to 15 attempts in order to be successfully matched for a chat. Thus, many students were frustrated. A follow up analysis (Ferschke et al., 2015) reports a negative impact on commitment to the course for students who experienced this frustration. An important lesson learned from this study was that whereas

providing the opportunity for synchronous chat was positive for students for whom it was possible to be matched for a chat easily, this positive effect was balanced with a negative effect in the case where the lack of critical mass despite the total enrollment of 20,000 students from their MOOC was not sufficient to enable a quick match. Additionally, research on conversational agents shows that the efficacy of the agents is heavily influenced by multiple factors such as the cognitive skills of the learner, the teacher's authority, the students' background, the agent verbosity level, the difficulty of the instructional domain, the students' interaction pace as well as the type of the facilitation strategy employed by the agent.

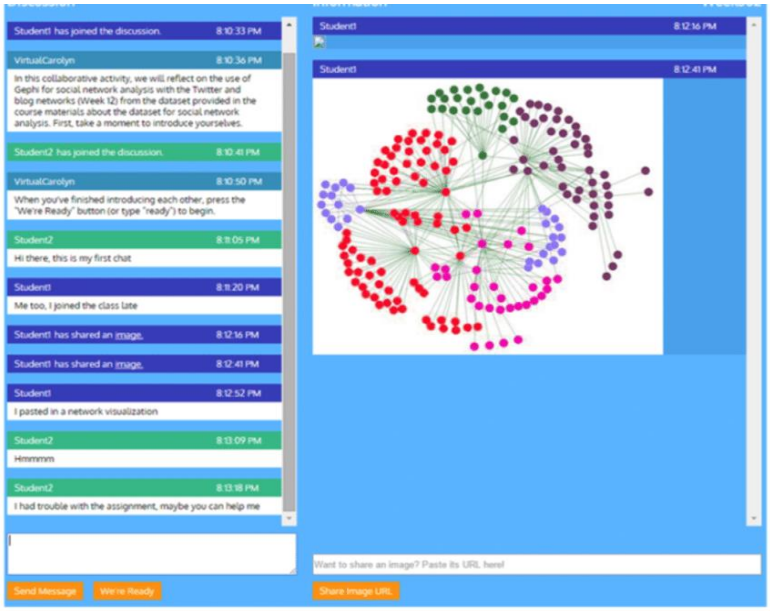


**Figure 23**: An example of an agent-based synchronous chat activity in a MOOC environment (Rosé & Ferschke, 2016)

The combination of all those factors illustrate that more experimentation is required in order to fine tune the design of conversational agents providing collaborative learning support of considerable pedagogical value in MOOC settings. There is a clear need to define a group of best practices and strategies enabling the implementation of efficient conversational agents that effectively support and guide students participating in MOOCs and are agile enough to be able to operate in different online discussion contexts without requiring a lot of setup effort.

### 2.3.3 Conclusions

Concluding the above analysis, we emphasize the following:

- Conversational agents (or chatbots) have been used in learning settings as a special type of pedagogical agent, to support any type of learning outcomes.
- Agents have been used mostly in one-on-one learning settings (agent interacting with one student), while research so far has identified both benefits and risks/shortcomings of the approach. Importantly, current studies highlight the need to understand how to benefit most from engaging both a teacher and an agent in the same educational environment.
- Recently, research is published (limited though) focusing on the issue of integrating an agent in collaborative learning settings, thus supporting the peers in processes of collaborative knowledge building.

- Massive Open Online Courses (MOOCs) also have attracted research interest as a promising and challenging learning setting for deploying conversational agents, aiming to provide essential students support in the absence of human teacher's presence.
- Finally, it is important to emphasize that conversational agent building in the learning setting is typically considered as an expert's group task while the perspective of developing agents to be configured by teachers (thus, engaging teachers as agent co-creators) is not explored at all.

# 3 Proposed model

## 3.1 Identified 'gaps' to guide the conversational agent design approach

In this section we identify 'gaps' as opportunities for innovative design, development and research based on the previously presented (and also partially in other deliverables as well) CA design approaches and relevant research outcomes. The objective of this section is to highlight how these identified gaps have become major aspects in the design and development of the colMOOC project agent service.

As discussed previously, many agent approaches exist and can be used in many different fields. In education, conversational agents have been used to reveal social skills and interact with learners on a broad range of topics, to perform a wide variety of instructional roles, to provide customized instruction or feedback to the learners in various learning scenarios, or even interact with learners using various communication channels.

Conversational agents can be used in one-to-one learning settings but also can support groups. Limited research has been made regarding the use of conversational agents in dyads conversation settings where students interact through a chat interface and a conversational agent intervenes in the discussion. In such settings, the key objective is to have productive conversational peer interactions, such as argumentation, explicit explanation and mutual regulation, triggered and expanded by the agent intervention and contribution to the group activity. In order for the agent to make proper interventions it must be aware of the domain students are talking about and the APT moves that could potentially enrich students' discussion and positively affect the results of collaborative learning. In a learning context, the key design feature of an agent would its configurability by the teacher who is the expert to shape the domain properly. This functionality, to provide teachers with an interface to create the knowledge domain of the agent, is not currently supported by existing conversational agent approaches.

In the scope of the above background, a prototype configurable conversational agent system was designed (from now on we refer to this prototype as the 'colMOOC agent'). In designing the colMOOC agent we considered three design perspectives as 'gaps' that provide opportunity for innovation. These design perspectives address respectively the cognitive, socio-cultural and technical dimensions of technology-enhanced learning. In detail:

a) colMOOC agent to *support productive forms of peer dialogue* (cognitive dimension).
b) colMOOC agent as *teachers' open tool for configuring a domain-independent agent* (socio-cultural dimension).
c) colMOOC agent as an *interoperable tool to be integrated in MOOCs platforms* (technological dimension).

These three critical design perspectives of the colMOOC agent are presented in detail in the following subsections. The ongoing design and implementation process of the overall colMOOC system is viewed as an opportunity to explore relevant critical research questions.

### 3.1.1 colMOOC agent to support productive forms of peer dialogue (cognitive dimension)

Although the conditions that facilitate collaborative knowledge construction cannot be easily identified, numerous research efforts focused on the factors making group discussions productive for learning and community building (Sionti, Ai, Rosé, & Resnick, 2012). As discussed in D1.2, *transactivity* and *explicitness* can be regarded as two valuable indicators of a productive dialogue. Drawing on the socio-cognitive theoretical framework, researchers universally value:

- (a) *Explicitness*: the explicit articulation of reasoning in dialogue, as well as

- (b) *Transactivity*: the existence of references and connections among items of articulated reasoning (Stahl & Rosé, 2011; Sionti et al., 2012).

Building on those findings, which emphasize the educational benefits of learning activities where partners use one another as information resources, we propose that the development of conversational agents supporting learners during collaborative activities should take into consideration the *value of agent-generated prompts* in encouraging (and also challenging) learners to explicate their reasoning and sustain transactive dialogue as a productive form of peer interaction.

Considering that peer interactions constitute the primary learning mechanism in collaborative learning settings, the agent design does not focus on thoroughly modeling each learner's understanding by using complex knowledge structures for each different domain. Instead, attention is given on identifying efficient techniques of modeling and triggering constructive peer interactions through fine-tuned APT agent interventions (Tegos, Demetriadis, Papadopoulos, & Weinberger, 2016).

Taking into account that the 'explicitness' of students' reasoning is typically regarded as a constructive conversational behavior and a prerequisite for dialogue transactivity, some research groups investigated the impact of automated agent interventions on students' explicitness. A study showed that the operation of a conversational agent delivering APT-based interventions can increase the levels of explicit reasoning reported in students' discussions (Tegos, Demetriadis & Karakostas, 2015). An examination of students' agent-induced contributions revealed that students often respond to agent questions using statements exhibiting explicit reasoning about the key domain concepts being discussed (Tegos & Demetriadis, 2017). Dyke, Adamson et al.'s (2013) have reported similar findings, suggesting that a APT agent intervention mechanism, prompting students to share and clarify their own thinking, can drastically increase the amount of students' reasoning.

In their research study, Tegos, Demetriadis and Karakostas (2015) also proceeded to explore whether the dialogue 'explicitness' factor has a significant mediating effect on students' learning outcomes. Typically, mediation is regarded as a hypothesized causal chain in which one variable affects a second variable that, in turn, affects a third variable. The intervening variable, M, is the mediator, which 'mediates' the relationship between a predictor, X, and an outcome, Y. After analyzing all students discussions in search of peers' explicit contributions, it was found that students 'explicitness' could serve as a mediator (M), carrying the influence of the conversational agent interventions on students learning outcomes.

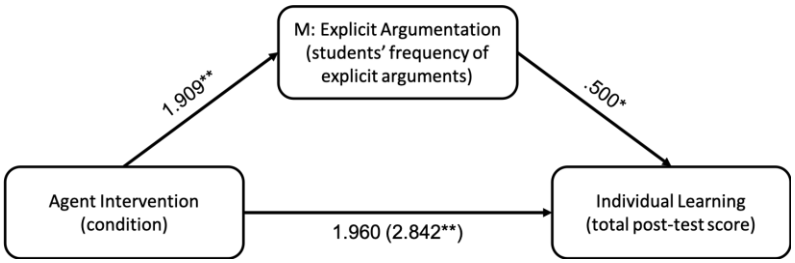


**Figure 24**: The indirect effect of the agent interventions on individual learning through explicit argumentation

The model illustrated in Figure 24 indicates that the beneficial impact of the conversational agent interventions on learning was mediated by the stimulation of explicit arguments throughout

students' discussions. This effect shows that the efficacy of a conversational agent displaying unsolicited APT interventions in collaborative learning settings could be greatly enhanced if the agent was able to systematically engage peers in explicit argumentation. This finding is in line with other substantial research evidence highlighting the pedagogical value of engaging collaborators in argumentative discourse (Andriessen et al., 2013).

### 3.1.2 colMOOC agent as teachers' open tool for configuring a domain-independent agent (socio-cultural dimension).

Another key design perspective is that the colMOOC agent is designed as a tool enhancing the impact of the facilitation strategies as configured by the teachers themselves. This design approach further supports the role of the teacher/instructor who in charge of the orchestration and scaffolding of a learning activity.

All reviewed design approaches in developing agents for learning are strongly based on the perspective of an agent developed and configured by experts and integrated in an educational setting partly replacing the teachers in their role. In our approach, however, the conversational agent is not designed to serve as a teacher replacement but as a tool configured by the teacher, aiming to facilitate students' discussions through a series of dynamic interventions, which have been proven to be effective in classroom settings. The colMOOC conversational agent service enables the teacher to configure the dialogue-based behavior of the agent in chat activities and scaffold peer discussion through APT-based dynamic agent interventions.

Therefore, in the colMOOC project, a 'teacher-as-agent-designer' viewpoint is adopted and the agent can be considered as a 'teacher-group mediator', meaning that the conversational agent software aims to automatically multiply and extend the impact of teachers' constructive interventions that otherwise would be impossible to perform. We believe that this design approach 'respects' the socio-cultural role of teachers/instructors by empowering them to act as the sole designers of the domain-relevant interventions performed by the agent. Thus, when peers interact with the agent the learning experience is based on teachers' understanding of the domain and relevant instructional design decisions and not some third-party experts.

### 3.1.3 colMOOC agent as an interoperable tool to be integrated in MOOCs platforms (technological dimension).

Massive Open Online Courses (MOOCs) are becoming increasingly popular as an important alternative supplier of knowledge, training and accreditation, aiming principally at large-scale interactive participation and open access via the web (Littlejohn, 2013). Several MOOCs (so called 'xMOOCs') are mainly developed as informational landscapes offering video-based tutoring enhanced with typical learning interactions, such as closed type questions, quizzes and certain student forums. They adopt mainly a 'knowledge transmission' perspective emphasizing teacher-led individual learning, rather than learning through peer interaction (Conole, 2013). However, current trends in MOOC design include also consistent effort to advance more social and collaborative forms of learning by providing tools for peers to constructively interact during the course. In this way MOOC designers expect -among others- to also increase students' motivation, engagement and satisfaction from the learning experience, and, in return, to lessen MOOC students' high dropout rates (e.g. Barak, Watted and Haick, 2016; Ortega-Arranz et al., 2017; Yang et al., 2017).

The colMOOC project acknowledges that MOOCs provide an opportunity for exploring the development of smart technologies as teachers' open tools that might further motivate and productively engage MOOCs learners in the learning activity. Therefore, the colMOOC conversational agent tool is developed as a web-based service able to be connected to MOOC

platforms and activated as a service for implementing chat-based activities. During a chat activity the agent service will be able to model peer dialogue through linguistic analysis and decide when to implement questioning interventions based on the teacher-configured domain model and on contextual parameters relevant also to the APT model.

In order for the agent tool to be reusable in several domains, one key aspect of the colMOOC agent design is that the domain model can be imported in the agent tool by the teacher; thus, the agent becomes a flexible domain-independent tool. Using the provided editing interface, the MOOC teacher can set up online chat-based activities, configure the domain model of the conversational agent for each activity, and review visualizations of the agent-peers learning interactions provided by relevant learning analytics component. This also opens the way for building a community of practice, with instructors sharing and annotating successful designs of 'colMOOC' type conversational agents, including teacher-configured domain models that generate agent questioning interventions relevant to interesting tasks and problems for students to discuss.

### 3.1.4 Conclusions

The 'gaps' identified as opportunities for innovative design in the context of the colMOOC project are:

- Build an agent software tool able to enact teacher designed interventions during peer discussions to promote more challenging and productive forms of peer interactions. Thus, the colMOOC agent is designed with a strong focus on APT-based teacher interventions which are expected to promote more productive and transactive forms of peer dialogue (cognitive dimension).
- Build the agent software tool as teachers' open tool for configuring a domain-independent agent (socio-cultural dimension), as opposed to the typical approach of building the agent system pre-configured by domain experts.
- Build the colMOOC agent as an interoperable tool/service to be integrated in MOOCs platforms (technological dimension), as opposed to the non-integrable software design approach.

In the following we describe how the above identified opportunities guided the design decisions of the colMOOC agent tool, regarding three respective dimensions:

- *Agent intervention strategies*: how the agent enacts intervention strategies based on the APT model of teacher interventions.
- *Agent domain configuration*: how the teacher can model any specific domain and provide a conceptual basis for the agent intervention strategies.
- *Agent integration in MOOCs platforms*: how the agent software architecture supports the integration of the agent service in MOOCs platforms.

## 3.2 Agent intervention strategies

In order for the agent to perform any specific intervention during peer chat discussion it is necessary to provide a model of what an intervention strategy is and how it can be computationally implemented. In this section we:

- a) Model the concept of 'agent intervention strategy' and relevant 'transaction patterns' in the context of the colMOOC project, and
- b) Provide, as a showcase, an example of an intervention strategy (namely, the 'AddOn' strategy). A complete list of the implemented intervention strategies and transaction

patterns of the colMOOC agent is available in deliverable D2.2. However, for the current deliverable D2.1 to be self-contained we include here a specific example.

Definitions:

1) *Intervention strategy*: by this term we refer to the abstract representation of the process implemented in the agent software system that eventually results to the agent taking part in the peer discussion. The result of an intervention strategy is -most of the times- that the agent avatar appears in the chat service and poses a question to peers or makes some other statement (could be informative or providing some guidance). An intervention strategy comprises several levels of implementation, ranging from:
    a) the *higher* level abstractions that provide the overall perspective and pedagogical rationale of the strategy (for example, the 'AddOn' agent intervention strategy is relevant to the teacher practice to intervene in peer discussion and ask/encourage one peer to further comment ('add something more') in relation to what the other peer has just stated),
    b) the *medium* level that includes the definition of several aspects of the strategy in detail,
    c) the *lower* level of code-based implementation in the specific computational setting where it is implemented.
2) *Intervention*: we use the simple term "intervention" to refer to the concrete onscreen manifestation of any intervention strategy of the agent. For example, an intervention stemming from the 'AddOn' strategy might be the appearance of the agent avatar on screen prompting a student as follows: "Maria, would you like to add something to what Steve mentioned about constructivism being a learning theory?"
3) *Transaction pattern*: by this term (or simply 'pattern' if there is no risk of confusion) we refer to the exact conditions happening during peer dialogue that trigger the agent to enact some intervention strategy and eventually perform an intervention. 'Exact' refers to the requirement that the pattern should be defined in such a way that enables its computational representation in the form of a clearly defined algorithm. For example, the pattern for the intervention strategy 'AddOn' can be described as follows: "10 seconds after a domain concept was introduced by a student, their partner has either remained silent or sent a short reply, consisting of 3 or less words". We chose to name this kind of patterns as 'transaction' (and not -for example- 'interaction' patterns) mostly to remind to all stakeholders that it is the 'transactional' quality of the dialogue that matters for triggering an intervention strategy. The transactional quality of the dialogue is relevant to the degree that peers reason on each other's contributions/comments thus collaboratively developing a common understanding or problem solving strategy. Transaction patterns, therefore, represent somehow some transactionally poor peer dialogue situation where the agent is programmed to identify as opportunity for enacting an intervention strategy. In the example above the system recorded information that "...partner has either remained silent or sent a short reply, consisting of 3 or less words" refers to a transactionally poor situation that becomes part of a pattern and triggers the respective intervention strategy.

The table below presents how a specific agent intervention strategy called "AddOn" is structured and modeled. First, the intervention is identified by:

1) Its specific "Intervention name",
2) Its current implementation "version" and,
3) A short information regarding its "development state" and inclusion in colMOOC player component v.0.1.

The following rows of the table, present:

4) "Intervention Description": a description of the conditions triggering the intervention and of what is accomplished by the delivery of the agent intervention,
5) "Purpose" (or goal): the learning (cognitive) objective of the intervention,
6) "Transaction pattern" ("triggers when"): a precise description of the conditions that may lead to the delivery of the actual intervention,
7) "Requires Modeling Of": a list of contextual information that needs to be modeled in order to enable the detection of the transaction pattern,
8) "Requires Conceptual Link(s)": the conceptual links that are relevant to the implementation of the specific intervention strategy,
9) "Transaction pattern example": showcasing an occasion that can be considered an intervention trigger, and
10) "Intervention message example": displaying the text of the actual intervention that could be delivered by the agent in a colMOOC chat environment.

**Table 2: The 'AddOn' intervention strategy**

| | |
|---|---|
| Intervention Name | AddOn |
| Intervention Strategy Version | 1.0 |
| Development State | Implemented in colMOOC player v.0.1. |
| Intervention Description | After a student introduces a particular key concept into the dialogue and their partner have remained silent or provided a very short response within 10 seconds, the agent intervenes asking their partner if he/she has anything to add on that specific concept. |
| Purpose | • Prompt a student for further participation and encourage them to explicate their thoughts on a key domain concept introduced by their partner.<br>• Support accountability to the Learning Community. |
| Triggers When (Transaction Pattern) | Ten (10) seconds after a domain concept was introduced by a student, their partner has either remained silent or sent a short reply, consisting of three (3) or less words. |
| Requires Modelling Of | • Domain concept detected in last student utterance<br>• Domain concepts already discussed (chat history)<br>• 10-second monitoring of partner's actions (detection of utterance size) |
| Requires Conceptual Link(s) | With 1 or 2 concepts<br>*[Concept A]* or *[Concept A + Relationship + Concept B]* |
| Transaction Pattern Example | *"[10:32:28] Student A: I think [Concept] is really important since...*<br>*[10:32:32] Student B: ok"* |

| | |
|---|---|
| Intervention Message Example | *"[10:32:38] Agent: [Student A], would you like to add something to what your partner [Student B] said about [Concept A/B]?"* |

*Comments*: This particular type of intervention derives from the APT discourse framework and aims to support accountability to the learning community. In general, accountability to the learning community emphasizes the need for students to listen to one another, attending carefully so that they can use and build their ideas upon their partners' ideas. In essence, an AddOn interventions attempt to keep group members on the same page and help them link their contributions to the on-going conversation relating to a domain concept (e.g., Concept A). For instance, if Student-A has talked about Concept A, the agent may intervene asking Student-B a question such as "Student-B, do you have anything to add to what Student-A has said about `Concept A`?"
Please note that a full description of the APT-model that constitutes the basis of the agent intervention strategies is provided in the 2.3 section of D1.2.

## 3.3 The colMOOC Editor and Player components

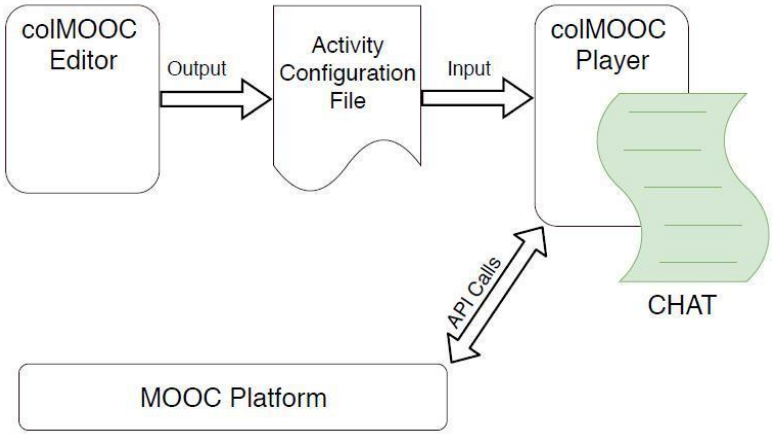


**Figure 25**: Major components of colMOOC architecture: Editor and Player

The colMOOC basic architecture consists of two major components: the colMOOC editor and the colMOOC player. With the colMOOC editor component, teachers can create chat-based activities where a topic is given to students which they asked to discuss on, and furthermore provide their collaborative answer. Teachers can enter all the activity's relevant information, such as the discussion topic, guidance instructions, and shape the agent's domain model by inserting conceptual links.
The output of the editor becomes input for the player to enact the chat-based activity. The colMOOC player is in direct connection with the MOOC platform through an Application Programming Interface (API) to receive relevant information such as the course ID and the students' IDs. The player component is now responsible for auditing learners' chat and makes intervention where necessary. Those interventions are being orchestrated by the domain model that has been shaped by the instructor and from the intervention model. The interventions are depicted in the chat dialogue area. The type of interventions is based on the APT model to encourage participants to have a more productive dialogue. Please note that explicit information regarding the

colMOOC editor and colMOOC player components and their functionalities is reported in deliverable D2.2 (CA Module) and deliverable D4.2 (1st version of the colMOOC Platform).
To create a colMOOC chat activity in a MOOC platform, the specific MOOC platform must be "colMOOC compatible". Being a platform colMOOC compatible means that the specific platform provides teachers the opportunity to create a new "colMOOC chat activity" just like they create activities of other types. By choosing to create a new colMOOC chat activity, teachers create a direct link to the editor part of the activity at the colMOOC server, to further set up the agent. MOOC platform transfers all available data to the colMOOC server through API, such as the course ID and the student IDs that participate in the course.
After setting up the agent, the activity configuration file is being created at the server, and the same link that was used by the teacher to set up the activity is being used by the students to load the player and access the activity at the colMOOC server. By entering the player, students form dyads in order to start the activity.

## 3.4 Agent domain configuration

In this section we present how the teacher user can model any specific domain and provide a conceptual basis for the agent intervention strategies.
The colMOOC editor employs an administration interface enabling educators to create a conversational agent by entering a series of conceptual links, forming the agent domain model in a task. This approach originates from the work of Tegos & Demetriadis (2017), who utilized a concept mapping interface allowing teachers to easily configure web-based conversational agents. The colMOOC editor follows a similar approach and enables the dynamic configuration agent-based chat activities. Each activity, which is associated with a task, asks students to work in groups and provide a joint response to an open-ended learning question.
A teacher can use the domain configuration panel of the colMOOC editor in order to model the agent domain knowledge by entering a series of conceptual links, which are simply structured statements involving key domain concepts. Each statement may consist of a subject (concept A), an object (concept B), and a verb or verbal phrase (connection). All those elements can be rendered by the colMOOC platform and visualized in a concept map as illustrated in the prototype mockup (Figure 26 below). This map can essentially be regarded as the domain knowledge representation of the agent for the specific activity.

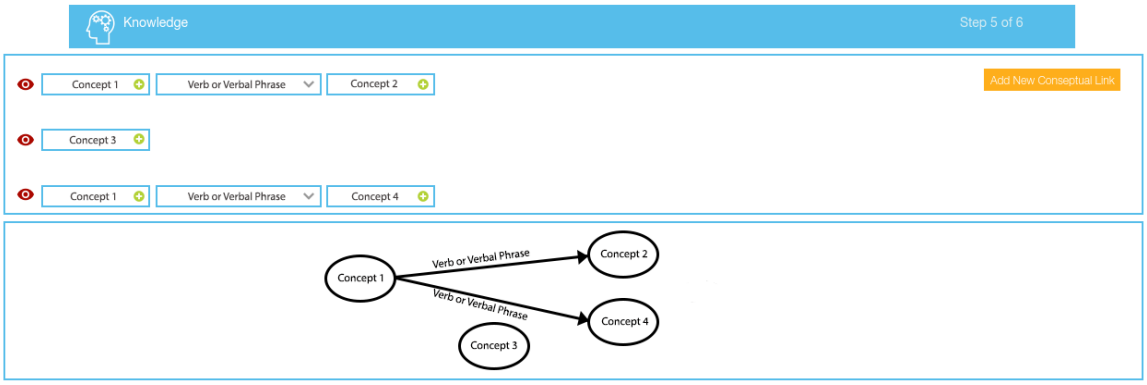


**Figure 26**: Agent domain knowledge configuration (prototype UI)

## References

A. Littlejohn, “Understanding massive open online course,” CEMCA EdTech Notes, 2013, 1–12. Retrieved from http://cemca.org.in/ckfinder/ userfiles/files/EdTech.

A. Ortega-Arranz, L. Sanz-Martínez, S. Álvarez-Álvarez, J.A. Muñoz-Cristóbal, M.L. Bote-Lorenzo, A. Martínez-Monés, and Y. Dimitriadis, “From Low-Scale to Collaborative, Gamified and Massive-Scale Courses: Redesigning a MOOC”, Proceedings of the Fifth European MOOCs Stakeholders Summit, EMOOCs 2017, Madrid, Spain, May 2017.

Adamson, D., Ashe, C., Jang, H., Yaron, D., & Rosé, C. P. (2013). Intensification of group knowledge exchange with academically productive talk agents. In N. Rummel, M. Kapur, M. Nathan, & S. Puntambekar (Eds.), To See the World and a Grain of Sand: Learning across Levels of Space, Time, and Scale: CSCL 2013 Conference Proceedings (vol. 1, pp. 10-17). ISLS.

Adamson, D., Dyke, G., Jang, H., & Rosé, C. P. (2014). Towards an agile approach to adapting dynamic collaboration support to student needs. International Journal of Artificial Intelligence in Education, 24(1), 92-124.

Andriessen, J., Baker, M., & Suthers, D. (Eds.) (2013). Arguing to learn: Confronting cognitions in computer-supported collaborative learning environments (Vol. 1). Springer Science & Business Media.

Baker, R. S. 2016. Stupid tutoring systems, intelligent humans. International Journal of Artificial Intelligence in Education, 26(2): 600–614.

Baylor, A. L., & Kim, Y. (2005). Simulating instructional roles through pedagogical agents. International Journal of Artificial Intelligence in Education, 15, 95–115.

Ben Mimoun, Mohammed Slim, & Poncin, I. 2015. A valued agent: How ECAs affect website customers' satisfaction and behaviors. Journal of Retailing and Consumer Services, 26: 70–82.

Berry, D. C., Butler, L. T., & Rosis, F. de 2005. Evaluating a realistic agent in an advice-giving task. International Journal of Human-Computer Studies, 63(3): 304–327

Brandtzaeg, P. B. (2017). Why People Use Chatbots, 377–392. https://doi.org/10.1007/978-3-319-70284-1

C. C. Aguirre, C. D. Kloos, C. Alario-Hoyos and P. J. Muñoz-Merino, "Supporting a MOOC through a Conversational Agent. Design of a First Prototype," *2018 International Symposium on Computers in Education (SIIE)*, Jerez, 2018, pp. 1-6. doi: 10.1109/SIIE.2018.8586694

Chaudhuri, S., Kumar, R., Howley, I., & Rosé, C. P. (2009). Engaging collaborative learners with helping agents. In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Eds.) Proceedings of the 14th International Conference on Artificial Intelligence in Education (pp. 365-372). Amsterdam: IOS Press.

D. Yang, T. Sinha, D. Adamson, and C. P. Rosé, “Turn on, tune in, drop out: anticipating student dropouts in massive open online courses,” NIPS Data-driven Education Workshop. vol. 11, p. 14, 2013.

Dyke, G., Adamson, D., Howley, I., & Rosé, C. P. (2013). Enhancing scientific reasoning and discussion with conversational agents. *Learning Technologies, IEEE Transactions on*, *6*(3), 240-247.

Dyke, G., Adamson, D., Howley, I., Rosé, C. P. (2012). Towards academically productive talk supported by conversational agents. In S. A. Cerri, W. J. Clancey, G. Papadourakis, & K. Panourgia (Eds.), Proceedings of the 11th International Conference on Intelligent Tutoring Systems (pp. 531-540).

Ferschke, O., Yang, D., Tomar, G., & Rosé, C. P. (2015). Positive impact of collaborative chat participation in an edX MOOC. In International Conference on Artificial Intelligence in Education (pp. 115-124). Springer, Cham.

Fu, Y., Xiong, H., Member, S., Lu, X., Yang, J., & Chen, C. (2016). Service Usage Classification with Encrypted Internet Traffic in Mobile Messaging Apps. *IEEE Transactions on Mobile Computing*, *15*(11), 2851–2864. https://doi.org/10.1109/TMC.2016.2516020

G. Conole, "MOOCs as disruptive technologies: strategies for enhancing the learner experience and quality of MOOCs," RED - Revista de Educación a Distancia, 39, 1–17, 2013.

Goel, A., Creeden, B., Kumble, M., Salunke, S., Shetty, A. & Wiltgen, B. (2015). Using Watson for Enhancing Human-Computer Co-Creativity

Griol, D., Baena, I., Molina, J. M., & de Miguel, A. S. (2014). A Multimodal Conversational Agent for Personalized Language Learning. In C. Ramos, P. Novais, C. Ehrwein Nihan, & J. M. Corchado Rodríguez (Eds.), Ambient Intelligence-Software and Applications (pp. 13-21). Springer International Publishing.

Gulz, A., Haake, M., Silvervarg, A., Sjödén, B., & Veletsianos, G. (2011). Building a Social Conversational Pedagogical Agent: Design Challenges and Methodological Approaches. In D. Perez-Marin & I. Pascual-Nieto (Eds.), Conversational Agents and Natural Language Interaction: Techniques and Effective Practices (pp. 128-155). IGI Global.

Heffernan, N. T., & Croteau, E. A. (2004). Web-based evaluations showing differential learning for tutorial strategies employed by the Ms. Lindquist tutor. In Proceedings of the 7th International Conference on Intelligent Tutoring Systems (pp. 491-500). Berlin: Springer.

Hone, K. S., & El Said, G. R. (2016). Exploring the factors affecting MOOC retention: A survey study. Computers and Education, 98, 157–168. https://doi.org/10.1016/j.compedu.2016.03.016

Huang, J.-X., Lee, K.-S., Know, O.-W., & Kim. Y.-K. (2017). A chatbot for a dialogue-based second language learning system. In k. Borthwick, L. Bradley & S. Thouësny (Eds), CALL in a climate of change: adapting to turbulent global conditions - short papers from EUROCALL 2017 (pp. 151-156). Research-publishing.net. https://doi.org/10.14705/rpnet.2017.eurocall2017.705

Hughes, C. 2018: The Year of the Chatbot – Chatbots Magazine; https://chatbotsmagazine.com/2018-the-year-of-the-chatbot-fc5a5f780a31, 08 Jan 2018.

Kennedy, C. M., Powell, J., Payne, T. H., Ainsworth, J., Boyd, A., & Buchan, I. (2012). Active Assistance Technology for Health-Related Behavior Change: An Interdisciplinary Review. Journal of Medical Internet Research, 14(3).

Kerly, A., Hall, P., and Bull, S. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. Know.-Based Syst., 20(2):177–185.

Kerly, A., Ellis, R., & Bull, S. (2008). CALMsystem: a conversational agent for learner modelling. Knowledge-Based Systems, 21(3), 238-246.

Kerry A., Ellis R., Bull S. (2009) Conversational Agents in E-Learning. In: Allen T., Ellis R., Petridis M. (eds) Applications and Innovations in Intelligent Systems XVI. SGAI 2008. Springer, London

Klopfenstein, L. C., Delpriori, S., Malatini, S., & Bogliolo, A. (2017). The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms. Proceedings of the 2017 {Conference} on {Designing} {Interactive} {Systems}, 555–565. https://doi.org/10.1145/3064663.3064672

Kumar, R., & Rosé, C. P. (2011). Architecture for Building Conversational Agents that Support Collaborative Learning. IEEE Transactions on Learning Technologies, 4(1), 21-34.

M. Barak, A. Watted, and H. Haick, "Motivation to learn in massive open online courses: Examining aspects of language and social engagement," Computers & Education, 94, 49–60, 2016.

Man, C. B. (1966). ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine.

Michaels, S., O'Connor, M. C., Hall, M. W., & Resnick L. B. (2010). Accountable Talk Sourcebook: For Classroom That Works. University of Pittsburgh Institute for Learning.

Oracle; Differences Between Command Line Interface and Graphical User Interface - https://docs.oracle.com/cd/E19683-01/806-7612/startup-78447/index.html, 07 Dec 2018.

Oudeyer, P.-Y., Gottlieb, J., & Lopes, M. 2016. Chapter 11 - Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. In B. S. a. S. Knecht (Ed.), Progress in Brain Research : MotivationTheory, Neurobiology and Applications, Volume 229: 257–284: Elsevier.

Pelk, H. Machine learning, neural networks and algorithms – ITNEXT; https://itnext.io/machinelearning-neural-networks-and-algorithms-4cd27a105b89, 07 Dec 2018.

Radziwill, N., & Benton, M. (2016). Evaluating Quality of Chatbots and Intelligent Conversational Agents.

Resnick, L. B., Michaels, S., & O'Connor, C. (2010). How (well structured) talk builds the mind. In R. Sternberg & D. Preiss (Eds.), From genes to context: New discoveries about learning from educational research and their applications (pp. 163-194). New York: Springer.

Rosé, C. P., & Ferschke, O. (2016). Technology support for discussion based learning: From computer supported collaborative learning to the future of massive open online courses. International Journal of Artificial Intelligence in Education, 26(2), 660-678.

Rus, V., D'Mello, S., Hu, X., & Graesser, A. (2013). Recent advances in conversational intelligent tutoring systems. AI magazine, 34(3), 42-54.

S. Tegos, S., and S. Demetriadis, "Conversational Agents Improve Peer Learning through Building on Prior Knowledge," Educational Technology & Society, 20 (1), 99–111, 2017.

Serban, I. V, Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., … Bengio, Y. (2017). A Deep Reinforcement Learning Chatbot arXiv : 1709 . 02349v2 [ cs . CL ] 5 Nov 2017, 1–40.

Sionti, M., Ai, H., Rosé, C. P., & Resnick, L. (2012). A framework for analyzing development of argumentation through classroom discussions. In N. Pinkwart, & B. McLaren (Eds.), Educational technologies for teaching argumentation skills (pp. 28–55). Bentham Science Publishers.

Söllner, M., Bitzer, P., Janson, A., & Leimeister, J. M. 2017. Process is king: Evaluating the performance of technology-mediated learning in vocational software training. Journal of Information Technology, 18(2): 159

Stahl, G., & Rosé, C. P. (2011). Group cognition in online teams. In E. Salas, & S. M. Fiore (Eds.), Theories of team cognition: Cross–disciplinary perspectives. New York, NY: Routledge/Taylor & Francis.

Tegos, S., & Demetriadis, S. N. (2017). Conversational Agents Improve Peer Learning through Building on Prior Knowledge. Educational Technology & Society, 20(1), 99-111.

Tegos, S., Demetriadis, S., & Karakostas, A. (2015). Promoting academically productive talk with conversational agent interventions in collaborative learning settings. Computers & Education, 87, 309-325. DOI: 10.1016/j.compedu.2015.07.014

Tegos, S., Demetriadis, S., Papadopoulos, P. M., & Weinberger, A. (2016). Conversational agents for academically productive talk: A comparison of directed and undirected agent interventions. International Journal of Computer-Supported Collaborative Learning, 11(4), 417-440.

The Dawn of the Conversational Interface. (2016). https://doi.org/10.1007/978-3-319-32967-3

Tomar, G. S., Sankaranarayanan, S., & Rosé, C. P. (2016, March). Intelligent conversational agents as facilitators and coordinators for group work in distributed learning environments (MOOCs). In 2016 AAAI Spring Symposium Series (Vol. 2).

Tomar, G. S., Sankaranarayanan, S., Wang, X., & Rosé, C. P. (2017). Coordinating collaborative chat in massive open online courses. arXiv preprint arXiv:1704.05543.

VanLehn, K., Graesser, A., Jackson, G. T., Jordan, P., Olney, A., & Rosé, C. P. (2007). Natural Language Tutoring: A comparison of human tutors, computer tutors, and text. Cognitive Science, 31(1), 3-52.

Veletsianos, G., & Russell, G. S. (2014). Pedagogical agents. In M. Spector, D. Merrill, J. Elen, & M. J. Bishop, (Eds.), Handbook of research on educational communications and technology (pp. 759-769). New York, NY: Springer.

Walker, E., Rummel, N., & Koedinger, K. R. (2011). Designing automated adaptive support to improve student helping behaviors in a peer tutoring activity. International Journal of Computer-Supported Collaborative Learning, 6(2), 279-306.

Wik, P., & Hjalmarsson, A. (2009). Embodied conversational agents in computer assisted language learning. Speech communication, 51(10), 1024-1037.

Wik, P., & Hjalmarsson, A. (2009). Embodied conversational agents in computer assisted language learning. Speech communication, 51(10), 1024-1037.

Winkler, R. & Söllner, M. (2018): Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. In: Academy of Management Annual Meeting (AOM). Chicago, USA

Winkler, Rainer; Söllner, Matthias; Neuweiler, Maya Lisa; Rossini, Flavia Conti & Leimeister, Jan Marco: Alexa, Can You Help Us Solve This Problem? How Conversations With Smart Personal Assistant Tutors Increase Task Group Outcomes. 2019. - CHI'19 Conference on Human Factors in Computing Systems. - Glasgow.

Y. Kim, A. L. Baylor, and E. Shen, “Pedagogical agents as learning companions: The impact of agent emotion and gender,” J. Comput. Assisted Learn., vol. 23, pp. 220–234, 2007