# CONVERSATIONAL MODELLING FOR CHATBOTS: CURRENT APPROACHES AND FUTURE DIRECTIONS

*Michael McTear*

*Ulster University*
*mf.mctear@ulster.ac.uk*

**Summary:** Chatbots have emerged as a new way to interact with services on the web and with smart devices, providing conversational interfaces that are ostensibly more intuitive and more natural than traditional user interfaces. This paper examines the conversation models of the main chatbot frameworks and assesses the extent to which they model important conversational phenomena such as follow-up questions, changes of topic, out-of-scope utterances, and other conversational phenomena. The paper concludes by considering research directions for intelligent conversational agents of the future.

## 1 Introduction

Chatbots have emerged as a new way to interact with services on the web and with smart devices, providing conversational interfaces that are ostensibly more intuitive and more natural than traditional user interfaces. Almost all the major tech companies have been developing tools and frameworks that enable developers to produce chatbots with varying degrees of functionality. Currently the chatbot developer community has created more than 200,000 chatbots for Facebook Messenger, around 300 for Kik, and more than 10,000 Amazon Alexa skills.

However, while there is general convergence on terminology and technologies for the speech recognition and natural language understanding components of a chatbot, as far as conversational modelling is concerned there is a wide and confusing range of different terminologies and models. This paper examines the conversation models of the main chatbot frameworks and assesses the extent to which they model important conversational phenomena such as follow-up questions, changes of topic, out-of-scope utterances. The next section presents an overview of chatbots and conversational interfaces. This is followed by a presentation of some examples showing how some current chatbot systems handle conversational interaction and what support is provided in current frameworks and developer tools. The paper concludes by considering research directions for intelligent conversational agents of the future.

## 2 Chatbots and Conversational Interfaces

### 2.1 Defining the Conversational Interface

At the most basic level a conversational interface supports interaction on a turn-by-turn basis i.e. the human and the computer take turns in a dialogue that may either be transactional or conversational. A transactional chatbot is used to accomplish some task, such as booking a flight, making a purchase, or asking about football scores, while a conversational chatbot engages in “chit-chat” for primarily social purposes. Where the conversation is text-based, the interaction is similar in form to a chat between two human users on a messaging app, except that in this case one of the participants in the interaction is a chatbot. With voice-based conversations the interaction is similar to a conversation between two human users on the telephone. Figure 1 shows a typical architecture of a conversational interface.

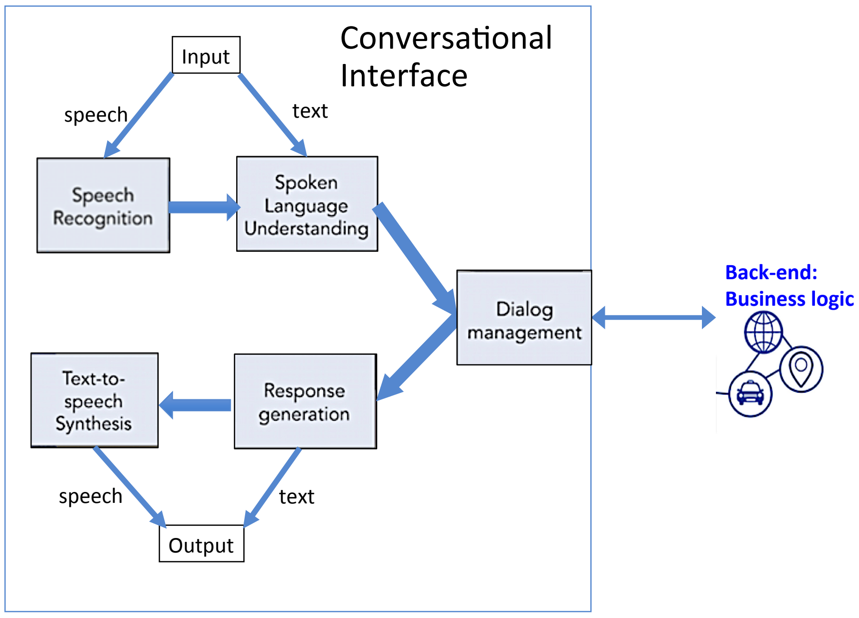

**Figure 1 - Typical architecture of a conversational interface**

The input to the conversational interface can be either speech or text. The input is processed by the Dialog Management component that updates the system's belief state and decides on the next system action. This may involve interacting with the back-end part of the system that includes business logic or access to web services. The system's output is produced either as text or as speech.

A conversational interface offers several advantages over traditional graphical user interfaces:

- It provides a different (and possibly easier) way to do things e.g. simply ask a query as opposed to navigating menus and drop-down boxes.
- It enables extended interaction e.g. to accomplish a task involving several steps or to engage in social chit-chat.
- It follows the conventions of conversational interaction that are familiar to users since early childhood.
- The recent emergence of chatbots with conversational interfaces is due in part to major advances in artificial intelligence, in particular in machine learning, that have resulted in greater accuracy in the technologies of speech recognition and natural language understanding.

## 2.2 Speech Recognition for Conversational Interfaces

Speech recognition has for a long time been seen as the major source of error in conversational interfaces and much of the design of commercial systems has been motivated by the need to reduce the possibility of speech recognition errors by drastically constraining the user's input to a small number of words or phrases [1,2]. However, with the recent integration of deep learning into speech recognition engines supported by large quantities of training data, speech recognition accuracy has improved dramatically and speech is now being used widely in practical applications [3].

## 2.3 Natural Language Understanding for Conversational Interfaces

Current natural language understanding toolkits for conversational interfaces extract intents and entities from the user's utterances. Intents represent what the user wants to achieve, e.g. find a restaurant, book a taxi, etc. Developers supply sample utterances and the system uses machine learning, typically Support Vector Machines, to create classifiers that can handle variations and similar utterances. Entities are the parameters required to fulfil an intent e.g. a location, time, cuisine, etc. Extracting entities is treated as a sequence classification problem for which techniques such as Conditional Random Fields are used [4]. More recently deep learning techniques have been applied to the identification of intents and the extraction of entities using Recurrent Neural Networks [5].

### 2.4 Conversation Modelling

Despite these advances in speech recognition and natural language understanding, it is still the case that chatbots are less than proficient in terms of their conversational abilities – for example, in dealing with follow-up questions, unexpected inputs, changes of topic, and recovery from error - so that often interactions with them are unnatural and error-prone. Moreover, because engaging in conversation is something that people do naturally from early childhood, it is often assumed that designing conversational interfaces is an easy task, involving the application of common-sense notions of how conversation works. However, conversation is a complex system that has been studied by linguists, conversation analysts, and others for more than five decades, looking at topics such as the structure, the sequential mechanics, and the technology of conversation. It is argued in this paper that designers of conversational interfaces ought to be draw on insights from this work if they want to make their systems more natural and more intuitive.

Developing a conversational interface can cover a potentially wide range of topics, including:

- The choice and use of different media for input and output.
- How to promote engagement and retention.
- How to make the customer experience more personal and more pleasant.
- The use of personas and branding.
- How to measure the quality of the interaction.
- Design guidelines e.g. how to design effective prompts, how to sound natural, how to act in a cooperative manner, being prepared to help at any time, …
- Whether the system simulates human conversation to the extent that it could be mistaken for a human (the so-called Turing test).

While these are important issues for conversational design, in this paper the focus is on a more restricted aspect of conversation modelling i.e. how to model the conversation flow in multi-turn conversations in such a way that the chatbot can receive and interpret a potentially wide range of inputs from the user and provide appropriate responses that keep the conversation coherent and on track and potentially lead to a successful outcome.

### 2.5 Conversation Modelling in Current Conversational Interfaces

There are two main types of conversational interaction in current conversational interfaces:

- One-shot queries
- Slot-filling dialogues

#### 2.5.1 One-shot queries

One-shot queries are **user-initiative**. They take the form of simple input-output pairs in which the user asks a question or issues a command, for example:

User: What’s the weather forecast for tomorrow in Frankfurt?
User: Set an alarm for 7am tomorrow.

**Table 1 – Conversational Features**

| | |
|---|---|
| User-initiated follow-up questions | Query about system's utterance (mixed-initiative on topic)<br>Correction of system's utterance<br>Request for more information<br>Request for help |
| User responses in slot-filling dialogue | User-answering (user does not fill any slots)<br>Over-answering (user fills more slots than requested by system) |
| User changes topic<br>Out-of-scope input by user | User's answer is not understandable or is unexpected |
| System-initiated sub-dialogues | System request for clarification<br>System offers help or additional information to repair a problem |

Typically there are no follow-up questions from the user and no questions are initiated by the system, for example, to clarify something that is unclear in the user's input. Any subsequent questions or command are treated as independent of any previous interactions and there is no maintenance of context across the queries. However, some current systems do maintain context for certain types of follow-up query, as shown in section 3.

### 2.5.2 Slot-filling dialogues

Slot-filling dialogues are **system-initiative**. Here the interaction is controlled by the system. For example, the following dialogue shows how the system collects information to answer the user's query about flights:

System: Where do you want to fly to?
User: Frankfurt.
System: What date do you want to travel?
User: Next Friday.

In most current systems the user is not able to take the initiative and ask questions, as in:

System: What date do you want to travel?
User: Is there a flight on Friday that would get me there before 10 am?

This type of interaction is referred to as **mixed-initiative**, as both system and user can take control of the interaction.

### 2.5.3 Towards open-ended dialogue

In contrast to one-shot and slot-filling dialogues a more natural conversation would involve mixed-initiative interaction and be potentially multi-turn, with context maintained across the conversation. Table 1 presents a preliminary list of conversational features that would be required in a more natural conversational interface (for similar lists of features, see [6,7]).

In order to investigate whether and to what extent these conversational features are used in current chatbot systems, conversational interactions were recorded using the following systems:

Google Home
Google Assistant (Android phone)
Amazon Alexa (Echo)

The following frameworks were then consulted to investigate whether they supported the conversational features – even if they did not appear to be implemented on some devices (or in particular actions or skills on the devices):

DialogFlow (Google)[1]
Alexa: Design Guidelines for Conversation[2]
Microsoft Bot Framework[3]
IBM Watson Conversation[4]

## 3 One-shot queries in Current Chatbots and Frameworks

### 3.1 Extending one-shot queries with user-initiated follow-up questions

To be able to answer follow-up questions the system needs to maintain the context of the original question-answer pair. Two types of follow-up are used commonly in current systems:

1. Questions that maintain the topic but change one or more parameters in the original query e.g.

   What's the weather forecast for tomorrow in Frankfurt?
   What about Belfast?
   No I meant London.
   What about Wednesday?

For these follow-up questions Google Home and Google Assistant were able to maintain context and answer the questions correctly. Amazon Alexa failed to maintain the context and treated the follow-up question as a new question:

User: What about Belfast?
Alexa: Belfast is the capital and largest city of Northern Ireland …

2. Questions that use anaphoric reference (e.g. pronouns) in the follow-up query e.g.

   Who is the Prime Minister of the United Kingdom?
   Is she married?
   How old is he?
   How old is she?

All three systems were able to handle follow-up questions involving anaphoric reference. In addition to pronouns, Alexa was able to handle anaphora involving the adverb *there*:

User: What's the weather in London?
Alexa: In London it's 8 degrees with mostly cloudy skies.
User: What's the population there?
Alexa: The population of London is about 8 million 7 hundred and 90 thousand …

### 3.2 Extending one-shot queries with system-initiated follow-up questions

System-initiated follow-up queries are required when there is something missing, ambiguous, or under-specified in the user's input, e.g.

[1] https://dialogflow.com/
[2] https://developer.amazon.com/designing-for-voice/
[3] https://dev.botframework.com/
[4] https://www.ibm.com/watson/services/conversation/

User: What's the weather in Spain?

Here the query is not specific enough as the weather forecast could be different in different parts of Spain. For this question Google Home and Google Assistant assume Madrid as the default location for the query, whereas Alexa asked for a particular location and provided additional help:

Alexa: Which location was that? I'm best with a specific city in Spain.

## 3.3 How follow-up questions are handled in chatbot frameworks

### 3.3.1 DialogFlow

DialogFlow allows follow-up intents to be added to any parent intent and provides several built-in follow-up intents that address the most common use cases: *fallback* (for queries that the chatbot cannot answer), *yes/no*, *later*, *cancel*, and *custom*. Context is maintained across intents and follow-up intents by setting an output context for a parent intent and the same context value for follow-up intents. This ensures that only those follow-up intents with input contexts that have the output context of a parent intent will be matched. Contexts for the built-in follow-up intents are generated automatically.

### 3.3.2 Amazon Alexa Developer

Alexa has a large number of built-in intents for common dialogue situations e.g. *cancel*, *help*, *repeat*, etc, as well as a built-in intent library e.g. *weather*, *local search*, *books*, *calendar*, etc. These intents can be customised. For example: a new skill-specific utterance can be added to the built-in 'help' intent e.g.

Alexa: Would you like me to send a car to pick you up at home or work?
User: How do I set my address? (customisation of 'help' intent)

Similarly the system can be set to prompt for missing or ambiguous information, as in a question for weather information where the location is missing. This strategy is similar to slot-filling as the user has not specified a required slot in their query.

### 3.3.3 Microsoft Bot Framework / Cortana

In an informal test with Cortana on an Android phone the follow-up question "What about Belfast" to a previous query about the weather in Frankfurt was treated as a new query, as seen previously with Amazon Alexa. The documentation for Microsoft Bot Framework states that "a *key to good bot design is to track the context of a conversation, so that your bot remembers things like the last question the user asked.*" There did not appear to be a specific reference to follow-up questions in the documentation. However, the Bot Framework provides a range of methods for maintaining state (for example, user data, conversation data), and there may be some way to implement follow-up questions using the state data.

### 3.3.4 IBM Watson Conversation

IBM Watson Conversation makes use of slots to handle follow-up questions as well as under-specified user queries. If a user asks a follow-up question with new variable values, as in the examples above, the original slot values are over-written. This method also supports a user-correction of a value. For under-specified queries that require a clarification request by the system (e.g. User: what's the weather?), the system enters a slot-filling dialogue and asks for the required variables e.g. location and day.

#### 3.3.5 Other approaches to contextual modelling

Slots are used in an approach called Slot Carry Over (SCO) [8]. Here a combination of rules and machine learning models with lexical and structural features form the current and previous turns are used to decide which slots from previous turns are still relevant.

## 4 Slot-filling dialogues in Current Chatbots and Frameworks

Slot-filling dialogues are supported in current chatbots for particular actions or skills. For example, in Amazon Alexa, invoking a skill for booking flights, such as the SkyScanner skill, initiates a slot-filling dialogue to collect the values for the required slots.

The mechanisms for slot-filling are fairly similar across the different chatbot frameworks and build on the methods pioneered in VoiceXML[5].

- **DialogFlow** makes use of a Parameter Table in which the required parameters and their associated slots are specified.
- In **Amazon Alexa** a Dialog Model is created in which required slots are specified along with optional slot and intent confirmation prompts. The Dialog Interface manages the slot-filling using the `dialogState` property to determine if additional steps are required in the dialog or if all steps have been completed.
- **Microsoft Bot Builder** provides two different approaches to slot-filling. In the Bot Builder SDK for .NET, the developer creates a `FormFlow` – a C# class or JSON schema – that is used to collect information form the user. The dialog is automatically generated based on the form. As an alternative the Bot Builder SDK for Node.js makes use of a Waterfall dialogue model in which a sequence of steps is specified for completing an action or prompting the user for information. The waterfall is implemented as an array of functions where the results of the first function are passed as input to the next function.
- **IBM Watson Conversation** makes use of a parameter table similar to that of DialogFlow.

These methods are able to support over-answering and under-answering by monitoring whether the required slots have been filled.

## 5 Modelling open-ended conversations

As can be seen, current chatbots and frameworks support the conversational features listed in Table 1 to some extent and in various different ways. In some cases special handlers and other similar mechanisms are used to deal with “unexpected input”. This is a viable solution for simple conversation but is unlikely to be applicable for longer, more open-ended, multi-turn conversations, where the user’s input is more difficult to anticipate.

The structure of conversation has been a topic of enquiry for linguists, sociologists, and others for more than five decades. Two main traditions can be identified;

- Describing conversation structure in terms of dialogue grammars.
- Describing the interactional work that participants perform in order to make conversations work.

[5] https://www.w3.org/TR/voicexml20/

## 5.1 Describing conversation structure in terms of dialogue grammars

Dialogue Grammars have been developed by linguists and computational linguists working with conversational data and interested in specifying the structure of conversation. Examples of conversational structures are: adjacency pairs [9], exchanges [10], discourse segments [11], and conversational games [12]. Dialogue grammars have proved useful for annotating transcripts of conversations, but they have been extensively criticized on a number of grounds [13]. For example, they treat conversation as a product rather than providing explanations for various interactional phenomena that occur in conversation and how participants interpret these phenomena. Moreover, the description of conversation sequences in a similar way to the syntactic structure of sentences implies that sequences that do not fit the grammar are ungrammatical. However, such ungrammatical sequences in conversation are hard to find – rather participants in conversation work to find the relevance of responses in the context of the ongoing conversation.

The notion that a conversation can be mapped out in advance according to a grammar is essentially similar to the practice of designing a conversation flow to handle all the predicted paths through a conversation. Many chatbot platforms provide graphical tools to design conversation flow e.g. Converse.ai[6], ChatflowKitt.ai[7], PullString[8], and others. As with dialogue grammars, conversation flow diagrams may be suitable for simple conversations with few choices but they can quickly become very complex and unmanageable [14].

## 5.2 Describing the interactional work that participants perform to make conversations work

An alternative approach, developed by researchers in a tradition known as Conversation Analysis, is to view conversation as an interactional achievement where structure emerges as a result of interactional work by the participants. In this view conversation is locally organized as participants and emerges on a turn-by-turn basis. For example: adjacency pairs, which at their most basic level consist of a “first pair part” followed by a “second pair part” (e.g. a question followed by an answer) are not treated as objects of a grammar but as opportunities for interactional work. In this view, after one participant has produced a first pair part, there is an expectation that the second participant will produce a response that displays an understanding of that first pair part. As a conversation proceeds, each turn provides an opportunity for the participants to monitor the conversation and display their understanding. See also [15]. In this way conversational structure emerges dynimically and when problems arise, as in a misunderstanding, interactional work is performed to make a repair.

Conversation Analysis has informed a new approach to the implementation of multi-turn dialogue known as the Natural Conversation Framework (NCF). NCF is a design framework for conversational user experience being developed at IBM-Almaden by Moore and colleagues [7]. NCF provides a library of generic conversational UX patterns inspired by natural human conversation patterns as documented in the Conversation Analysis literature. The basic sequence is the adjacency pair but this is expandable in terms of pre-, insert-, and post-expansions to produce dynamically evolving patterns. The framework has been implemented on the Watson Conversation Service but the authors claim that it is applicable on other platforms.

[6] http://www.converse.ai/
[7] https://login.kitt.ai/#/?from=chatflow
[8] https://www.pullstring.com/

## 6 Some issues

### 6.1 Can conversation structure be hand-crafted

There are three main approaches to the implementation of computational models in conversational interfaces: hand-crafted, statistical, and end-to-end systems using deep neural networks.

The techniques that have been described in this paper and that are available in the main chatbot frameworks to deal with the complexities multi-turn conversations require a considerable degree of hand-crafting. Commercial systems are mostly hand-crafted and to some extent designers feel that this gives them more control over the systems that they create. However, the downside of hand-crafting is that it makes systems costly to develop, they are more likely to be error-prone, and the solution adopted may not be easily generalizable to new domains [14].

Statistical approaches use machine learning from data. The most widely used statistical approach is reinforcement learning using partially observable Markov decision processes (POMDPs), where the system learns an optimal dialogue strategy from interactions with users or simulated users. Systems developed using reinforcement learning are more robust to errors as they model uncertainty explicitly and they can generalize more easily to new domains, given suitable training data. However, there are tractability issues given the size of the state-action space as a belief distribution is maintained over all states with the system pursuing all possible dialogue paths in parallel. Learning a dialogue policy is also intractable, requiring efficient approximation techniques such as the summarisation of state and action spaces [16].

End-to-end dialogue systems using deep neural networks are the most recent development in the field. These systems learn to produce responses from large corpora of dialogues using generative models in which the responses are generated word-by-word [17]. Sequence to sequence mapping between an input and a response means that the system does not require the traditional pipelined architecture consisting of several components, such as speech recognition, natural language understanding, and so on. Deep learning requires large amounts of data and as yet there are not sufficient corpora of conversational data that cover the required range of conversational interactions.

### 6.2 Issues for developers

There are several problems that developers of conversational interfaces face:

- As we have seen, there is no clear model of conversation to guide the design and development of conversational interfaces.
- It is difficult to decide what functionalities to support in a conversational interface and even to know what can be supported in the various frameworks.
- Terminology is confusing - different frameworks use different terminology where, for example, terms such as ‘dialog’ can refer to a complete conversation and in other cases to a two part exchange or adjacency pair.
- There is a plethora of tools and frameworks with extensive documentation but many developers lack a detailed technical background in NLP to enable them to make full use of the tools.
- There is a need for standards in the area of conversational interfaces. Some attempts to address this issue are currently underway [18], but the danger is that a multitude of developers are creating conversational systems using only their intuitions for how conversation works rather than drawing on a rich literature and practical experience that has been developed over more than five decades of research [19].

### 6.3 Issues for users

Users of conversational interfaces face a number of issues. In particular, it is often not clear how to interact with a conversational interface. There is a great deal of inconsistency across different system:

- The same functionality may be handled differently on different systems e.g. whether they allow follow-up inputs, and which types of follow-up.
- Whether they support mixed-initiative e.g. user clarification requests.
- Inconsistency within the same chatbot: different versions of the same skill (e.g. weather forecast) may behave differently on the same chatbot.

## 7 Concluding remarks

In conclusion, it has been shown in this paper that conversation is a complex system, and we cannot rely on common-sense notions to design conversational interfaces. Modelling the sequential mechanics of conversation is essential to make interactions with chatbots intuitive and natural.

Finally, designers of conversational interfaces do not need to replicate how humans engage in conversation i.e. aiming at psycholinguistic reality. Nor do they need to try to fool users into thinking that they are conversing with another human being, as in the so-called Turing test and competitions like the Loebner Prize. Rather the aim should be what Oren Jacobs of PullString calls "human fidelity conversation" i.e. meeting the user's expectations of what is involved in engaging in a natural conversation.

## 8 References

[1] COHEN, M., J. GIANGOLA, and J. BALOGH: Voice User Interface Design. Addison-Wesley Professional, New York, 2004.

[2] PEARL, C.: Designing Voice User Interfaces. O'Reilly, Boston, 2016.

[3] LI, J., L. DENG, R. HAEB-UMBACH, Y. GONG: Robust Automatic Speech Recognition: A Bridge to Practical Applications. Academic Press, Oxford, UK, 2015.

[4] TUR, G., R. de MORI (eds): Spoken language understanding: systems for extracting semantic information from speech. Wiley, Chichester, 2011.

[5] MESNIL, G., Y. DAUPHIN, K. YAO, Y. BENGIO, L. DENG, D. HAKKANI-UR, X. HE, L. HECK, G. TUR, D. YU, and G. ZWEIG: Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Transactions on Audio, Speech, and Language Processing 23(3):530–539, 2015.

[6] LARSSON, S.: User-initiated Sub-dialogues in State-of-the-art Dialogue Systems. Proceeding of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 15-17 August 2017, Saarbrücken, Germany, 2017.

[7] MOORE, R. et al.: Conversational UX Design. https://researcher.watson.ibm.com/researcher/view_group.php?id=8426

[8] R. SARIKAYA et al.: An Overview of End-to-end Language Understanding and Dialog Management for Personal Digital Assistants. Proceedings of the Spoken Language Technology Workshop (SLT), 13-16 December 2016, San Diego, CA, USA, 2016

[9] SCHEGLOFF, E. A and H. SACKS: Opening up closings. Semiotica 8(4):289–327, 1973.

[10] SINCLAIR, J. M and M. COULTHARD: Towards an analysis of discourse. Oxford University Press, Oxford, 1975.

[11] GROSZ, B. J. and C. L. SIDNER: Attention, intentions, and the structure of discourse. Computational Linguistics 12(3):175–204, 1986.

[12] KOWTKO, J., S. ISARD and G. M. DOHERTY: Conversational games within dialog.

Research paper HCRC/RP-31, Human Communication Research Centre, University of Edinburgh, 1993.
[13] LEVINSON, S. C.: Pragmatics. Cambridge University Press, Cambridge, 1983.
[14] PAEK, T. and R. PIERACCINI: Automating spoken dialogue management design using machine learning: an industry perspective. Speech Communication 50:716–729, 2008.
[15] CLARK, H. and S. E. BRENNAN: Grounding in communication. In: RESNICK LB, J. M. LEVINE and S. D. TEASLEY (eds) Perspectives on socially shared cognition. American Psychological Association, Washington: 127–149, 1991.
[16] YOUNG, S., M. GASIC, B. THOMPSON, and J. WILLIAMS: POMDP-based Statistical Spoken Dialogue Systems: A Review. Proceedings of the IEEE, Volume 10, Issue 5: 1160-1179, 2013.
[17] SERBAN, J. V., A. SORDONI, Y. BENGIO, A. COURVILLE and J. PINEAU: Building End-to-end Dialogue Systems Using Generative Hierarchical Neural Network Models. https://arxiv.org/abs/1507.04808, 2016.
[18] W3C Voice Interaction Community Group. https://www.w3.org/community/voiceinteraction/
[19] MCTEAR, M., Z. CALLEJAS and D. GRIOL: The Conversational Interface: Talking to Smart Devices. Springer, New York, 2016.