Improving speech interaction with mixed initiative dialogue

Loquendo SDS - Spoken Dialog System

Morena Danieli, Giovanni Manca

1 From system-driven interaction to mixed-initiative dialogue

The transition from system-directed to mixed-initiative human-machine dialogue has been a focus of several research institutes and industries during the past several years [4,5]. The term "mixed-initiative" was introduced in the artificial intelligence literature by Jaime Carbonell, who described a prototype that "is capable of maintaining a mixed initiative dialogue with the student, with questions asked by either side and answered by the other" [3]. A similar definition is proposed by Balentine and Morgan: they write that, the term "mixed-initiative" identifies "a dialogue in which interactions are sometimes initiated by the users and sometimes by the machine" [1], but that smooth turn-taking is still difficult, and not yet fully realized.

The interaction style in mixed-initiative dialogues may be radically different from the question-answer sequences provided by most off-the-shelf speech tools, because users can decide in which order their requests parameters are to be specified, reducing the dialogue time and increasing the success rate. However, the increased flexibility also promotes greater complexity. Actually, several factors affect the transition in question, the most important of which are the troubles related to modelling a user's spontaneous linguistic behaviour, and the development costs of highly flexible dialogue models. Loquendo has identified some feasible solutions to some of these critical factors, implementing them in Loquendo Spoken Dialogue System (Loquendo SDS), a dialogue manager for speech-based applications. Loquendo SDS has two main objectives: on the one hand, reducing the potentially negative impact of unexpected user linguistic behaviour, on the other hand reducing the development costs for mixed-initiative information-seeking dialogues.

In order to fulfil the first objective, Loquendo SDS provides a set of scalable error recovery strategies that make it possible to continue a dialogue with the user even when difficulties in recognizing user speech may occur. In addition, the dialogue manager is based on an accurate separation of (reusable)dialogue knowledge from language and domain-dependent knowledge. This separation helps to lower the development costs of accurate dialogue strategies for spoken language systems.

2 Main features of Loquendo SDS

Loquendo SDS is a system for human-machine dialogue management that can be integrated with every VoiceXML interpreter through the VoiceXML 2.0 <subdialog> element. Its main features include:

- **Multiple parameter acquisition:** through a single phrase spoken naturally, the system can acquire all the necessary information available for matching the user's request:
  - **LoqSDS**: *Please tell me your departure and destination*
  - **User**: *I'd like a direct flight from Washington, DC to Rome, Italy on December, 15th.*
- **Multiple confirmations:** through a single phrase, the user can easily correct speech recognition errors, even if there are more than one:
  - **LoqSDS**: *Would you like to fly from Boston to Rome, Italy on December, 19th?*
  - **User**: *I'm leaving from Washington, flying to Rome on December, 15th*
- **Consistency acquisition control:** the system can verify the contents of the dialog parameters, guaranteeing correct voice recognition whilst allowing the system to provide the user with the correct information.

- **Mixed initiative:** users can speak freely and naturally, requesting relevant information in the order they choose. The system is dynamically configured to respond to the client's requests and it responds in a way that is consistent to the application domain.
    - **LoqSDS**: *Please tell me your departure and destination.*
    - **User**: *Flight from London to Paris, tomorrow around nine-twenty pm. Can you tell me the time of arrival?*
    - **LoqSDs**: *Flight from London to Paris, tomorrow evening. Please tell me the departure airport.*
- **Elevated flexibility:** the system can guide an inexperienced user just as easily as an expert user. Using a dialog that allows the acquisition of necessary information through a single word or through a limited number of interactions, the system is capable of running an articulated conversation in which the number of interactions is increased if necessary.
    - **LoqSDS**: *Please tell me your departure and destination.*
    - **User**: *Paris.*
    - **LoqSDS**: *Leaving from Paris. Where do you want to fly?*
    - **User**: *Milan.*
    - **LoqSDS**: *From Paris, France to Milan, Italy. Which airline do you want to fly with?*
    - **User**: *Air France.*
- **Dialogue development tools:** dialogue knowledge and linguistic knowledge for spoken dialog application can be implemented within the development system of Loquendo SDS. The development system includes a set of tools that can be used to specify domain and linguistic knowledge, to personalize the dialogue manager behaviour by choosing among its set of dialogue and error recovery strategies. For those already acquainted with Loquendo speech technologies, it is worth noting that the dialogue development tools can be integrated in Loquendo Studio. Its use can provide the developer with an intuitive yet powerful environment in which to model sophisticated dialog interfaces.

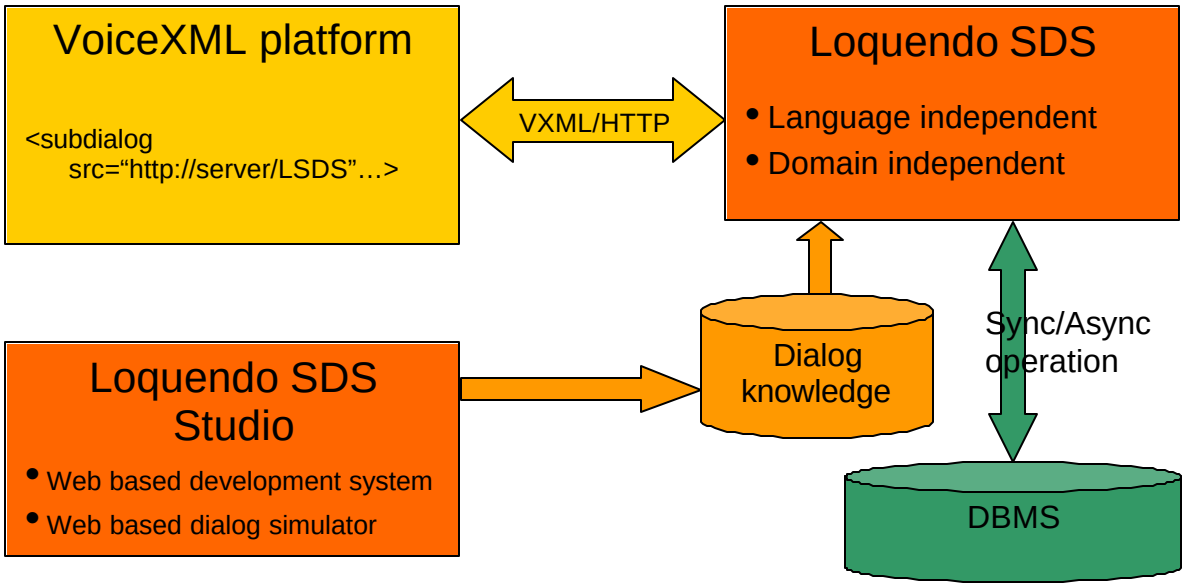

Figure 1: Loquendo SDS integration schema

Loquendo SDS can be accessed as a web service over HTTP. It relies on an Internet Information Server and exports the interfaces illustrated in Figure 1. The picture shows how Loquendo SDS can interact with a VoiceXML platform. The first interface is showed by the bi-directional arrow that

connects the two boxes at the top. It is based on VoiceXML through the <subdialog> tag. It allows an easy integration with every VoiceXML interpreter. The second interface is based on a proprietary XML format, created by the Loquendo SDS Studio, represented by the box on the left at the bottom. This is a web based development tool that can assist the dialogue designer in creating the dialogue domain and the linguistic knowledge. It defines a set of commands and responses and can be used to integrate Loquendo SDS into any environment. The dialogue knowledge that describes the information content of the application can thus be used by the domain and language-independent spoken dialogue system, granting it the necessary knowledge for managing the domain application. The dialogue system can also interact with a DBMS in both synchronous and asynchronous interactions. Thus, the dialogue system can receive information from the DBMS without necessarily stopping the dialogue with the user.

3 Loquendo SDS approach

3. 1 Loquendo SDS primitives

Loquendo SDS is based on a set of primitives that the developer uses for describing the dialogue domain. **Communication Acts,** different types of **Parameters, Tasks** and **Query Acts** are the Loquendo SDS primitives. The most important primitive is *Communication Act* (CA). Each CA defines the actions that the system can activate in order to interact with the user, given the state of one or more dialogue variables. The CAs enable the acquisition and confirmation of meaningful pieces of information being exchanged by user and system (we call them *Parameters,* there are some different types of parameters) for each phase of the dialogue (we call that phase *Tasks,* in a dialogue there can be several Tasks). During the interaction with the user, Loquendo SDS can access external resources, such as databases, through the execution of special actions (we name these actions *Query Acts*) in a synchronous or asynchronous way. A special class of Parameters (Data parameters) are used for representing the data acquired by accessing external data.

3. 2 The separation of domain and language knowledge

In developing spoken dialogue applications, it is useful to keep separate knowledge about the domain (such as the information items domain-specific, communicative actions that can be performed, etc.), from knowledge about the language (such as the speech acts uttered by the system, the ASR objects used in the recognition phase, etc.). In the Loquendo SDS approach the domain knowledge of an application is the set of parameters, tasks, and query acts defined for the domain. For example, in a train timetable application, the parameters are the departure, the arrival, the type of train, the cost of the travel, the date and time of the travel, and so on. It is also part of the domain knowledge the set of communicative acts that can be used to talk about the parameters, i.e. requesting departure and arrival cities, confirming them, negotiating possible inconsistencies. In the same domain, the tasks are accessing the timetable database for retrieving the timetables, or accessing the database for retrieving the cost of a selected train connection, and so on. All these pieces of information contribute to the tailoring of the behaviour of the general-purpose dialogue manager to a specific domain.

Since the same domain can be dealt with by speaking in several different languages, if we keep separate the domain and language knowledge (i.e. prompts, repair messages, links to the ASR objects, and so on), the separation of the two sources of knowledge can dramatically reduce the porting costs of an application from one language to a different one. Loquendo SDS greatly emphasizes this separation, and its development tools drives the developers in thinking first of the domain, and then to the linguistic resources needed to implement it [2].

3.3 Domain Parameters

The goal of the interaction between caller and system is to give the system the information items it needs for solving the caller problem. In some sense, all the dialogue is organized around the information exchanged between caller and system. We call these information items with the name "parameters". Formally speaking, the parameters are <attribute, value> pairs. In the following subsections we will introduce the different types of parameters and their properties as supported by

the Loquendo SDS dialogue manager. Loquendo SDS saves parameter values, as soon as they are acquired, in an internal data structure. Although the application developer *does not need* to model the implicit goal of the dialogue manager, it may be useful for her/him to know that Loquendo SDS passes through the following steps:

a) one or more interaction turns with the user for acquiring the parameters values

b) one or more interaction turns for having their values confirmed, either implicitly, or explicitly, or by confidence levels associated with them.

3.4 Domain Tasks

A Loquendo SDS task is a subset of the domain parameters, which may be acquired independently from the remnants of the parameters. Loquendo SDS always assumes a *default task*, including all the parameters declared for a given domain. However, it could be useful to identify subsets of parameters, i.e. single tasks, to be executed autonomously. Each Loquendo SDS task is characterized by its name, and the subset of domain parameters involved in the task.

3.5 Communication acts and linguistic knowledge

Communication Acts are the communicative actions triggered by Loquendo SDS for processing the dialogue with the user. The communicative functions usually uttered in goal-driven information dialogue are making a request, asking for verification, keeping the communication channel, opening and closing the dialogue, solving ambiguities.

In the Loquendo SDS approach communication acts and the linguistic forms used for implementing them are a different source of knowledge, as we emphasized above. This is true in human-human dialogue, too. Actually, in dialogue interactions between humans, the competence used for asking a clarification is independent from the specific nature of the information item, and from the language spoken in the interaction. Analogously, in Loquendo SDS the knowledge for deciding which communicative action will be triggered in the next turn is independent from the specific linguistic knowledge. On the contrary, what will be actually said, will be application dependent, and it will be addressed in the linguistic section of the application knowledge.

Let us clarify the concepts above with an example. In the train timetable domain, if the value of the parameter Departure has a low recognition score, Loquendo SDS decides that that particular item will need to be confirmed in an explicit confirmation turn. This decision is independent from the nature of the involved information: it applies to every information item whose particular value is uncertain. The ability to take this decision is part of the dialogue competence of Loquendo SDS. For implementing this competence, Loquendo SDS has four types of CA: Request, Verify, Request_and_Verify (in the same dialogue turn), and Select (for solving ambiguities).

If we go on with our example, for asking the user to confirm the uncertain value of Departure, Loquendo SDS needs to know the specific prompt the dialogue designer wants to be issued in that particular dialogue situation, and the recognition object apt for the task. The definition of the linguistic prompts associated with Loquendo SDS and the association of them with labels of ASR objects is done with the aid of the Loquendo SDS development tool.

3.6 Error recovering

Each time Loquendo SDS detects a miscommunication of the kind or a failure of consistency, it is able to enter a sub dialogue for recovering the errors. Entering a recovery sub-dialogue means that the dialogue manager does not proceed in acquiring or verifying parameter values other than the one(s) it is focusing in the current communication turn. Loquendo SDS can detect automatically the occurrence of miscommunication, and possibly its source. If miscommunication occurs, the dialogue manager behaves differently, depending on the kind of communication failure. In particular:

c) If the communication failure is due to ASR errors or to inadequate linguistic behaviour of the user, i.e. s/he does not reply to a system request, the user is prompted again to provide the system with the focused parameter.

d) If the communication failure is due to inconsistent values of some parameter, the dialogue manager can either **negotiate** the correct values in a negotiating sub dialog, or **discard** the acquired values and re-prompt the user with a new request.

4. Conclusions

In this paper we have presented the main features and the general philosophy of Loquendo SDS, a dialogue manager able to deal with some common problems of spoken dialogue applications. We believe that Loquendo SDS may be effective in facilitating the moving on from system-directed to mixed-initiative dialogues, since it offers to the dialogue designers a set of general, yet flexible, dialogue strategies for managing possible miscommunication, and for keeping coherence in the dialogue. In addition, the Loquendo SDS approach is based on the separation of domain knowledge from linguistic knowledge. We believe that this features contribute to the reduction of the implementation costs of mixed initiative, multilingual applications.
In this paper most of the interesting features of Loquendo SDS are briefly covered: the interested dialogue designer can refer to the documentation available in [Loquendo_SDS_24p.pdf].

5. References

[1] B.Balentine, D.P.Morgan, *How to Build a Speech Recognition Application*, Enterprise Integration Group, 1999.

[2] G. Boella, R.Damiano, M.Danieli, and L.Lesmo, "Using XML for Representing Domain Dependent Knowledge in Dialogue", in V.Motouek, et al. (Eds), *Texts, Speech, and Dialogue*, Lecture Notes in Computer Science, Berlin, Springer-Verlag, 2001.

[3] J. R. Carbonell, *Mixed-Initiative Man-Computer Instructional Dialogues*, PhD Thesis, Boston, MIT, 1970 (https://dspace.mit.edu/handle/1721.1/13801)

[4] D.Dahl, *Practical Spoken Dialog Systems (Text, Speech and Language Technology)*, Springer, 2005.

[5] M.F. McTear, "Spoken dialogue technology: enabling the conversational user interface", *ACM Computing Survey*, 2002.

