

Interpretations and Discourse Obligations in a Dialog System

Márcio Mourão, Pedro Madeira, and Nuno J. Mamede

L²F – Spoken Language Systems Lab

INESC-ID Lisboa / IST, R. Alves Redol 9, 1000-029 Lisboa, Portugal

{Marcio.Mourao,Pedro.Madeira,Nuno.Mamede}@l2f.inesc-id.pt

<http://www.l2f.inesc.pt>

Abstract. Not all user intentions can be extracted from the speech acts associated with an user utterance. Extra reasoning may be needed to interpret what the user said. We describe how interpretations can be generated using (i) the knowledge associated with each domain object; and (ii) the system's expectations. We also present a way of connecting interpretations to discourse obligations, from where system decisions are taken.

1 Introduction

When we turned our efforts to implement the initial version of a natural language dialog system, we selected an architecture that could satisfy the requirements of adaptability to different domains. Our dialog system follows the TRIPS architecture [1], which includes the following modules: Interpretation Manager (IM), Task Manager (TM), Discourse Context (DC), Behavioral Agent (BA), and Generation Manager (GM). We use frames to represent both the domain and the information collected during the interaction with the users [5].

The IM controls the system's interpretation. When the user produces an utterance, the IM receives a set of speech acts [2], and generates interpretations and discourse obligations [3]. The TM houses domain specific task knowledge, which it abstracts away from the rest of the dialog system [5]. The DC manages all knowledge about the discourse, including the discourse stack, turn-taking information, and discourse obligations. The BA enables the system to be mixed-initiative: regardless of what the user says, the BA has its own priorities and intentions. The GM coordinates the generation activities.

2 Interpretation Manager

The input of the IM module consists of a set of speech acts. A speech act contains among other items, an identifier, the sentence said by the user, and a set of components that represent the user's intentions, like <TYPE> (the utterance performative), <WHO> (the utterance maker), and <WHAT> (with objects representing what was said). The <ITEM> tag encapsulates these objects.

2.1 Constructing Interpretations

Speech acts give rise to intentions, which we represent as interpretations. One interpretation has: (i) the same components of the speech act; (ii) recognized domain objects; (iii) one or more Problem Solving Acts (PSAs). Interpretations represent possible combinations of speech acts and the meaning associated to each object they contain. A PSA [5] is an instantiation of a frame, and is used by the IM to: (i) select the best formed interpretations; and (ii) generate Discourse Obligations (DOs) to proceed with the dialogue.

Using the TM and the recognition rules described in [5], the IM, after receiving a "Give me a ticket to Lisbon!" speech act, for example, starts by collecting knowledge related to the incoming objects, like "ticket" and "Lisbon". If "ticket" is known by the system as a request and nothing else, and "Lisbon" can be a departure or an arrival city and no more, the result should have two interpretations that differ in the semantics of these objects (see Fig. 1). But we also know, from the semantics of the speech act, that "Lisbon" is an arrival city. Therefore, one of the interpretations fails and is removed (see Fig. 1). Objects contain at least a <NAME>, a <DOMAIN>, and a specific type, shown as <SLT>. To express that "Lisbon" cannot be a departure city, we defined, in XML, a set of conditions (between interpretation properties and object types) that cannot be simultaneously verified within an interpretation. For example, an object considered to be a departure city cannot be included in the <towhere> interpretation property.

We try to match the interpretations with a previous system question. If the dialog is just starting, the system knows the user is not giving an answer and interpretation construction is finished. Suppose now that "Porto" is the content of a second utterance. After identifying "Porto" as a departure or an arrival city, and constructing the two corresponding interpretations, the IM tries to match them with a previous question. If the user's utterance is the answer to a system's question, "What is the desired departure city?", the system targets the first interpretation to be the intended one. If no interpretations can be formed, either as a result of an unrecognizable speech act, or by eliminating the only formed one, because of contradictions (the process described earlier), the system accepts the obligation of solving the problem. In these cases, the system may request the user's help, using a question like "Sorry, what did you say?". The same happens when we have an interpretation with an object that does not fit in any slot of the frame, or when there is no valid match with the context of the dialog. What the system should do is closely related to Discourse Obligations.

2.2 Constructing Discourse Obligations

In the context of a dialog, discourse obligations (DOs) can be created to represent the tasks of the system. The DOs can be used together with intentions [4], and most importantly, can account for the connection between a question and its answer. We create DOs by analyzing the existing interpretations.

Examples of DOs are: (i) *Greet* and *Introduce Self*, which are domain independent; (ii) *Domain Disambiguation*, *Slot Disambiguation*, *Request Slot*, *Validate Discourse*, and *Show Results*, all domain dependent; and (iii) *Clarify*, *Confirm*, *Correct*, which may be domain dependent (generated from a Problem Solving Acts - PSA) [1] [5] or

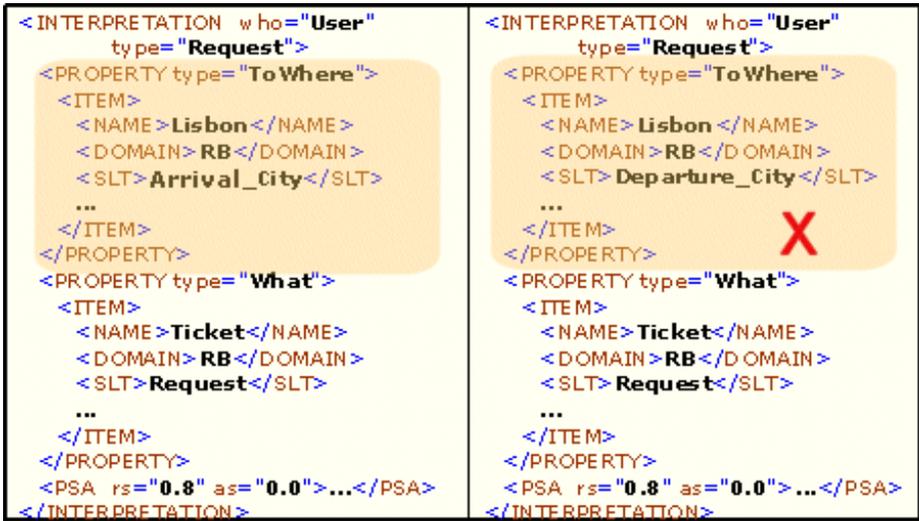


Fig. 1. Two interpretations: “Lisbon” is recognized as an Arrival City and as a Departure City

domain independent (generated directly from an interpretation). When no interpretations are formed from the speech act, or if there exists one object that is not recognized, the system creates a *Clarify*, a *Confirm*, or a *Correct DO*, depending exactly on how the interpretations are formed.

DOs like *Greet* and *Introduce Self* can be used to make the system behave politely. If, at this stage, other DOs were not yet created from the interpretation(s), the IM module will request TM to generate the PSAs associated with each interpretation.

More than one PSA can be associated with the same interpretation (one PSA is always related to one domain). An interpretation is associated with two or more PSAs when at least one object belongs to more than one domain (assuming the existence of two or more domains).

2.3 Building Domain Dependent Discourse Obligations

A Domain Disambiguation obligation is created when more than one PSA is associated with the interpretation(s). Let us suppose there are two domains: a movie ticket domain (<CIN>) and a bus ticket domain (<RB>). Two interpretations are created, because the object <Lisbon> belongs to both domains. A Domain Disambiguation is created to introduce this subject in the system’s agenda.

A Slot Disambiguation is created when one domain contains more than one interpretation: there is one object that fits at least two slots of the frame. This situation arises, for example, when the user says “Porto” and this object can be placed either in the departure or in the arrival city slot of a frame (see Fig. 2).

A Validate Discourse obligation is created when there is only one interpretation, only one domain involved, and the answer score is lower than 1.0. This conjunction means

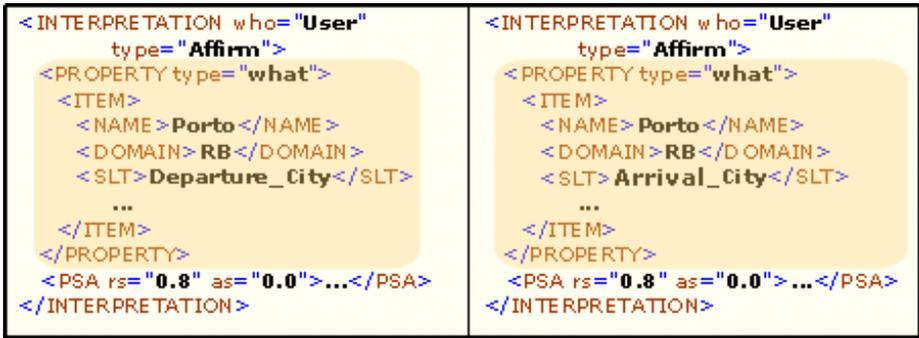


Fig. 2. Two interpretations for the object “Porto” that give rise to a “Slot Disambiguation” discourse obligation. The shaded areas show the two possible types of the “Porto” object.

that the frame is invalid (the best value for the recognition and answer scores is 1.0, while the worst is 0.0).

A Request Slot is generated whenever there is a valid frame, and the recognition score is lower than 1.0, which means that a question should be sent to the user to request the missing information.

Finally, the Show Results obligation is generated when we have only one interpretation and one domain, and both the recognition and answer scores are 1.0, that is, the frame is completely filled and results can be shown to the user.

In the example requesting a ticket to Lisbon, the Request Slot obligation is generated, since we are working with only one domain, only one interpretation was formed, and the answer score is 1.0. The system may then ask the user for the value of one of the empty slots. In this case, the system may ask the question “what is the departure city?”. This DO contains the possible answer values, which will enable the system to, later on, recognize that a sentence uttered by the user is an answer to the generated question.

References

1. J. Allen, G. Ferguson, A. Stent: An architecture for more realistic conversational systems, in Proc. of Intelligent User Interfaces (IUI-01), Santa Fe, NM, Jan, 14–17 (2001).
2. D.R. Traum: Speech Acts for Dialogue Agents. UMIACS, Univ. of Maryland, pp. 17–30 (1999).
3. J. Kreutel, C. Matheson: Modelling Questions and Assertions in Dialogue Using Obligations, 3rd Workshop on the Semantics and Pragmatics of Dialogue. Univ. of Amsterdam (1999).
4. D.R. Traum, J.F. Allen: Discourse Obligations in Dialogue Processing, in Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 1–8 (1994).
5. P. Madeira, M. Mourão, N. Mamede: STAR FRAMES – A step ahead in the design of conversational systems capable of handling multiple domains, ICEIS, Angers, France (2003).