Dialog Acts from the Processing Perspective
in Task Oriented Dialog Systems

Markus Berg
University of Kiel &
University of Wismar
mail@mmberg.net

Bernhard Thalheim
University of Kiel
Technical Faculty

Antje Düsterhöft
University of Wismar
Faculty of Engineering

1 Introduction
The formulation "I'd like to know what time it is" has the same aim as "What's the time?". Thus, we can easily see that different formulations can have the same intention. Consequently we learn that it is not possible to infer a one-to-one relationship between form and function. When developing a dialogue system, the main interest is what the user expects from the system, and not how he formulates his concern. So we propose a backend-oriented scheme for the description of dialog utterances. This scheme applies for three basic types of mixed-initiative systems that often have to be modeled: control systems (e.g. for controlling the lights in a room by speech), question-answering systems and information-seeking/booking-dialogues (i.e. the system asks questions in order to gain information that is necessary to fulfil the user's request). All of these systems have a task-related information exchange in common. Thus we don't classify by initiative (they are all mixed initiative) but by purpose and call those systems, according to (McTear, 2004, p.45), "task-oriented" dialog systems.

2 Modeling of Dialogs
While most dialog models start with linguistic aspects, we specify the model bottom-up. We have a backend and we know what it is able to do. Then we can find out how to address these functions, i.e. what linguistic form triggers which function.

2.1 Backend Functions
In the introduction we have mentioned three basic system types. This leads to three different categories of user aims:

- the user gives a command in order to make the system realize the request
- the user asks a question in order to retrieve an information
- the user gives information in order to enable the system to provide him with information

We now introduce appropriate functions that model these capabilities: do, getInfo and setInfo. The following examples are annotated with these basic functions and by this means indirectly describe the users aim, or the intended perlocutionary effect.

- Could you please switch on the light? → do
- Play some music → do
- How is the weather in London? → getInfo
- I'd like to start on May 4th → setInfo

2.2 Utterance Role and Speaker
We already observed that form is not function. Thus we should avoid the terms question and answer as they extremely relate to the form. So we replace them by the introduction of the terms concern and reply. A concern comprises all types of utterances that have the aim of causing a system reaction. This can be a regular question, a command, a request or just a wish. We summarize both a command and a question under the same category as they both constitute a form of system request. A reply is any possible response which satisfies the concern, i.e. an answer or an acknowledgement. Furthermore we introduce the speaker of an utterance leading to four base units (in combination with the utterance role): user concern (UC), user reply (UR), system concern (SC) and system reply (SR). After analysing several dialogs, we realized that the combination of SC and UR equals a UC: The system concern "Tell me your destination" and the user reply "San Francisco" equals the user concern "I'd like to go to San Francisco". So if the system initiates a question, by answering it, the user states his own concern. For the
dialog manager it is important to know of the utterance role in order to infer the next dialog step. For the backend itself it does not matter if the request was a UC or a UR in consequence of a SC.

### 2.3 Selection of Dialog Acts

In order to model the user’s intention we use dialog acts. While many dialogue act schemata suffer from the fact that form is mixed with function, we apply Bunt’s second-level general-purpose functions (Bunt and others, 2010): information seeking functions, information providing functions, commissives and directives. In the types of dialog system described in this paper we don’t need commissives, as we do not concentrate on human-human-like conversations. Of course the system could produce utterances like “I will look for that”, but from the backend processing perspective we don’t need to understand promises, invitations, oaths or threats. From the directives we only use the instructions-category and rename it to action requesting in order to delimit from the form (instructions are often associated with imperatives). These dialog acts can now be related to our backend functions: an information-seeking dialog act will initiate the getInfo function, an information-providing act initiates the setInfo function and an action-requesting act leads to the do function. Apart from these acts, we also have to do with what in DIT++ (Bunt and others, 2010) is called social obligations, like greeting/return greeting. These acts often don’t need any backend access, which means that they bypass it. Moreover they form symmetric adjacency pairs as the reaction always belongs to the same dialog act category as the request. Hence we name them copy dialog acts.

### 2.4 Description of Dialog Utterances

We now have described two different classification approaches: the distinction into concern and reply as well as the differentiation between information-seeking, information-providing, action-requesting and copy acts. The attempt to integrate both into a common taxonomy fails as we have to do with different, independent dimensions. While the first approach describes the role \( r \) of an utterance, the second approach describes its primary illocution \( i \) and its derived intended action \( a \). The role is important to enable the system to differentiate between “I’d like to go to New York” as a concern or as a reply to the question “Where do you want to go?”. Moreover an utterance is described by the speaker \( s \) and the domain \( d \) of the utterance, i.e. task oriented, dialog handling or social. It is further characterized by its form \( f \) (roughly equivalent with the secondary illocution) and the range \( R \) (only in case of info-seeking acts) of the resulting answer. Because range and action can be inferred from the primary illocution, we only have five independent attributes. Thus an utterance can be described by the following quintuple: \( U = (s, r, i, d, f) \) where \( s \in \{\text{user, system}\} \), \( r \in \{\text{concern, reply}\} \), \( i \in \{\text{inf.seeking, inf.prov., act.req., copy}\} \), \( d \in \{\text{task1, ..., taskN, dialog, social}\} \) and \( f = (\text{sentence type, mode, verb, style, } \ldots) \). So the sentence “Could you please close the window?” can be described as: \( U = (\text{user, concern, action requesting, smart room, (question, subjunctive, close, formal)}) \). This would result in a do backend call and no range because requests don’t expect an answer.

### 3 Conclusion

In this paper we have discussed dialog acts from the processing perspective in mixed-initiative task-oriented dialogs. For the system it is most important to recognize what the user wants in order to be able to accomplish his needs. There is no need for the backend to know whether he formulated a request as an instruction or as a question. We identified the role of an utterance and three classes of backend functions which build the basis for the top level of a backend-motivated and formulation independent taxonomy of illocutionary acts. It comprises information-seeking, information-providing, action-requesting and copy acts. An extension of these attributes results in a quintuple for the description of utterances in a dialog which is a compact way of representing the user’s aim and the intended system reaction in task-oriented mixed-initiative dialogs.

### References
