# Form-based Dialogue Structure for Task-oriented Conversations

*Ananlada Chotimongkol and Alexander I. Rudnicky*

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
{ananlada,air}@cs.cmu.edu

## Abstract

Task-oriented conversations have a clear structure that relates to the goal of the interaction. We propose a representation for such conversations based on forms and associated operators. We evaluate its adequacy through analysis of conversations in four different domains: air travel planning, bus schedule enquiry, map reading and UAV flight simulation. The proposed representation consists of two main components a task structure (featuring a three-level organization of task, sub-tasks and concepts) and a set of dialogue mechanisms that advance the dialogue. We show how the proposed framework accommodates two phenomena in task-oriented conversations: managing domain constraints and grounding referents. As these representation elements are observable in surface dialogue phenomena, we believe the approach can be used to build systems that learn how to converse in new domains.

## 1. Introduction

Informal observation of goal-oriented human-human dialogues shows that such dialogues have a clear structure. When two or more people engage in a conversation that has a specific goal, they organize their conversion so that key ideas are clearly communicated and that progress towards the goal is observable by both parties. This characteristic is reflected in the choice of language, which will be instrumental, and will reference shared representations of the task. If we can specify the characteristics of the representation, its related operations, and how they map to language behavior, we may be able to build systems that can learn to carry out (goal-directed) conversations.

For our purpose, we need a dialogue structure that is sufficient, general and learnable. The structure has to capture **sufficient** information for the development of a new dialogue system and be able to describe key phenomena in a conversation. It has to be **general** enough to describe goal-oriented conversations in dissimilar domains. Finally, it must be **learnable** from a corpus of human-human conversations. Many theories and models have been proposed to explain the structure of human-human conversations. However, the existing structures lack at least one of these desired properties. Theoretical-oriented dialogue structures, which focus on discourse understanding, can describe most phenomena in a dialogue, but are usually too complex to be modeled in a dialogue system or learned from a corpus of conversations. An example of theoretical-oriented structures are Grosz's Theory of Discourse Structure [1], a structure of discourse segments, discourse segment purpose relations and participants' focus of attention, and Discourse Representation Theory (DRT) [2], a logical representation of utterance meaning and the context, and HCRC dialogue structure [3], a hierarchical representation of sub-dialogues, initiation-response exchanges and utterances.

Engineering-oriented dialogue representations, which are designed for being used in automatic systems, on the other hand, are more suitable for actual applications. But how a dialogue system should treat each dialogue structure component is unclear. Only the information from an utterance level has been widely used to aid the understanding of user utterances. One example of engineering-oriented structures is Conversation acts [4], a structure of four levels action which extends the speech act theory [5] by adding both the level that is higher than the utterance level and the levels that are lower.

In this paper, we describe a dialogue framework that has these desired properties along with the concrete mapping between dialogue structure components and dialogue system behavior. The proposed structure is based on a form architecture that has been used successfully in many dialogue systems, such as the Philips train timetable information system [6] and the CMU Communicator system [7].

## 2. Task-oriented domains

To make the proposed dialogue structure generalized for conversations in different domains, we considered conversations in four different domains. The first two domains are information-access tasks, the others are a problem-solving task and a command-and-control task. Table 1 shows the number of conversations available in each corpus as well as those used in our analysis.

| Domain | Available | Analyzed | |
|---|---|---|---|
| | #Dialogues | #Dialogues | #Utterances |
| Bus schedule | 12 | 5 | 90 |
| Air travel | 43 | 4 | 273 |
| Map reading | 128 | 4 | 498 |
| UAV simulation | 2 | 1 | 224 |

*Table 1*: The amount of data used in the analysis

### 2.1. Let's Go Bus Information corpus

The Let's Go corpus consists of calls to the Pittsburgh Port Authority Transit system enquiring about bus schedules. Each conversation is a telephone conversation between a help desk operator and a client. We selected those conversations that involved enquiries about bus schedules (e.g., asking for the time that a specific bus leaves given a bus stop location, or asking for the buses that run between two locations). The Let's Go corpus is described in [8].

### 2.2. CMU Travel Agent corpus

This is a collection of human-human conversations between an experienced travel agent and clients arranging trips that include plane, hotel and car reservations. The corpus was collected during the development of the CMU Communicator system. The corpus is described in [9].

### 2.3. HCRC Map Task corpus

The HCRC Map Task corpus [10] from the Universities of Edinburgh and Glasgow consist of conversations between two participants: a route giver and a route follower. A route giver's map has a route printed on it while a route follower's one does not. The goal of the conversation is to reproduce the giver's route on the follower's map (which may have differences).

### 2.4. CERTT UAV corpus

This corpus consists of interactions among a pilot, a navigator and a payload operator, tasked to fly a simulation of an Unmanned Air Vehicle (UAV) on missions to get photographs of specified targets. The corpus was collected by Cognitive Engineering Research on Team Tasks (CERTT) Laboratory, New Mexico State University [11].

## 3. Form-based dialogue structure

To create a dialogue system for a particular domain, we need to specify the structure of the task and describe the necessary dialogue mechanisms. A *task structure* describes domain information that is necessary in order to achieve the task goal such as steps in the task and domain keywords. The *dialogue mechanisms* describe how the participants communicate to perform the task, by relaying information and managing their performance.

### 3.1. Task structure

A task can be decomposed into a series of steps or sub-tasks. In order to accomplish each sub-task, a set of related pieces of information has to be acquired. We model a structure of a task by a three-level organization that consists of task, sub-tasks and concepts.

A *task* corresponds to a subset of conversations that has a particular goal. Conversations in the same domain can have different goals. For example, in the bus schedule enquiry domain, a user can ask different kinds of questions, all related to the bus schedule, such as asking for a departure time given a bus number and stop location or asking for the buses that run between two locations. Each type of enquiry has a different goal and corresponds to a different task.

A *sub-task* (also *episode* or *topic*) is a step in a task that contributes toward the task goal. In the air travel planning domain, where the top-level task is to create a trip itinerary, the sub-tasks are flight reservation, hotel reservation and car rental reservation. A sub-task can be further decomposed into smaller steps. To obtain a consistent structure, we need to specify the appropriate level of sub-task decomposition. For this reason, we define a sub-task to be the smallest unit of a dialogue that contains sufficient information to execute a specific domain action. Examples of actions might be querying a database in the air travel planning domain, and drawing a line in the map reading domain. A sub-task can be represented by a form in a dialogue system.

A *concept* is key domain information needed to complete a task or sub-task. A concept is an abstraction or a type of information while a *concept member* is a specific entity. In the air travel planning domain, examples of concepts are city_name, and airline_name, while the members of the concept city_name would be "Pittsburgh", "Boston", etc. A concept is equivalent to a slot in a form.

### 3.2. Dialogue mechanisms

Dialogue mechanisms consist of operations invoked by participants to communicate and to perform the task. An *operation* corresponds to an utterance or a part of an utterance that has a unique consequence on the state of the conversation. Operation category models utterances' roles in a more concrete way compare to the category used in speech act theory [5] and DAMSL [12]. The role of each operation closely relates to a task structure. For example, an init_form operation causes a system to create a new form.

There are some dialogue mechanisms that require more than one utterance to complete e.g. a confirmation mechanism. This type of mechanism is called a *dialogue procedure*. We model a dialogue procedure by dividing it into sub-steps that fall within utterance boundaries, and representing each step as an operation. For example, a confirmation procedure can be divided into two operations: a confirmation_request and a respond.

Normally, an operation is activated by a verbal expression. For instant, an utterance "I would like to fly to Houston", activates a fill_form operation. However, there is no explicit expression that associates with an access_database operation; it is activated by the completion of the query form instead. An operation that activated by a state of the representation rather than a verbal expression is called a *non-verbal operation* while an operation that is activated by an utterance is called a *verbal operation*. A criterion that activates a non-verbal operation is called a *trigger*.

Operations can be classified in to two classes according to the characteristic of their contribution toward the goal: task-oriented operations and discourse-oriented operations.

#### *3.2.1. Task-oriented operations*

A task-oriented operation contributes directly to the completion of the task. Since a task is represented by a form, task-oriented operations manipulate a form in three different ways: 1) modify a form or a slot, 2) execute a form, and 3) discuss the result of a form execution.

The consequence of the same operation is the same regardless of the domain. For example, a fill_form operation, which fills a concept value into a slot in a specific form, has the same behavior in every domain; only the parameters of the operation, a concept value, a slot and a form, that are different. Figure 1 and Figure 2 show the consequence of a fill_form operation on the corresponding form in the bus schedule enquiry domain and the map reading domain respectively.

A consequence of a form execution operation, on the other hand, is related to the action associated with the sub-task represented by the form. A consequence of a form

execution operation for a query form is database access where as for a line_segment form it is a drawing action.

**USER:** *fill_form:* i wanted to take the **BusNo:[**28X**]** bus from /um/ **DepartLocation:[**forbes avenue**]** to **ArriveLocation:[**the airport**]**

**Query Form**
**DepartLocation:** forbes avenue **DepartTime:**
**ArriveLocation:** the airport **ArriveTime:**
**BusNo:** 28X

*Figure 1*: A fill_form operation and the corresponding form in the bus schedule enquiry domain.

**GIVER:** *fill_form:* when you get to **Origin:[**the other side of the rope bridge**]** go **Direction:[**down**]** about **Distance:[**a centimetre**]** immediately.

**Line Segment Form**
**Origin:** the other side of the rope bridge
**Orientation:** down
**Distance:** a centimetre
**Destination:**

*Figure 2*: A fill_form operation and the corresponding form in the map reading domain.

*3.2.2. Discourse-oriented operations*

A discourse-oriented operation manages the flow of the conversation rather than directly manipulates the content of the form. This type of operations is not crucial for the completion of the task and may not occur in every conversation. The examples of discourse-oriented operations are acknowledgement and greeting.

## 4. Discourse phenomena

This section describes how two discourse phenomena: domain constraint management and a grounding process, can be described using the proposed form-based dialogue structure.

### 4.1. Domain constraint management

In addition to a task structure that describes a list of forms and slots in each task, a domain constraint describes acceptable value for each slot. For example, a task structure specifies that a departure_date and a departure_month are slots in a flight_query form while a domain constraint specifies that a valid value range of a departure_date is [1st - 30th] when a value of a departure_month is "April".

The simplest domain constraint is a *single-slot constraint*, a constraint that specifies possible values for a slot independent of the values of other slots. It might, for example, be a list of valid concept members; valid values for a date are ordinal numbers from 1st to 31st.

A *multiple-slot constraint* captures dependencies between slots, either within the same form or in different forms. The dependency between the arrival date of the first leg and the departure date of the second leg of a flight is an example of dependency between forms. Multiple-slot constraints are handled by non-verbal operations. When a participant says, "I would like to depart on April 31st", a semantic parser recognizes "April" as a depature_month and "31st" as a departure_date, and passes this information to a dialogue manager. Since the utterance implies a fill_form operation, a dialogue manager fills the concept values into appropriate slots. When date-related slots are updated, a non-verbal operation check_date is activated to verify the values against date-related constraints. In this case, a check_date operation detects that the value "31st" in a departure_date slot is invalid and notifies a dialogue manager about the error.

### 4.2. Grounding processes

A grounding process is a process that creates mutual understanding between conversation participants. For example, in the map reading domain, a route giver may check the existence of the landmark on a route follower's map before using it to describe the route, as shown in Figure 3. A grounding process is not required in order to complete a task, but rather it makes a conversation more robust and efficient.

A grounding process can be used for two purposes 1) to verify the correctness of the communication or 2) to create new knowledge for future reference such as defining a new term. The grounding process that verifies the correctness of the communication is related to existing information in the form and uses a confirmation mechanism. The grounding process that creates new knowledge, on the other hand, is related to a new set of information and a new sub-goal which can be represented in a separate grounding form as illustrated in Figure 3. The information about the existence of the landmark is represented by two slots, whether it is on a giver map or on a follower map.

**GIVER:** *ask_fill_form*: have you got a **Landmark:[**secret valley**]**?
**FOLLOWER:** *respond*: yes
*store_info:*

**Grounding Form**
**Landmark:** secret valley **Location:**
**On Giver Map?:** yes **On Follower Map?:** yes

*Figure 3*: A grounding sub-dialogue and a grounding form in the map reading domain.

The grounding form is similar to the form that represents a sub-task in both the structure of the form and a set of operations that manipulates the form. The only difference is the action that associates with the form. A form that represents a sub-task associates with an observable action or an external action with respect to a dialogue manager while a grounding form associates with an internal action, which is observable only within a dialogue manger, e.g. a store_info action.

## 5. Dialogue structure evaluation

To evaluate the utility of the form-based annotation scheme, we annotated conversations from the four task-oriented domains described above. This consisted of coding each conversation on an utterance-by-utterance basis using form representations along with operators. Examples of markup are

shown in Figures 1-3. Using the proposed markup we found that 93% of utterances could be coded.

Utterances that could not be described by the proposed framework were classified into 4 categories: *out of domain (OOD)*, *out of scope (OOS)*, *indirect* and *task management (TM)*. An OOD utterance contains information that does not relate to the domain (and thus would not need to be accounted for), while an OOS utterance contains information that while in-domain, falls outside the scope of the defined conversation goal. An indirect utterance is an utterance that requires substantial reasoning or world-knowledge to interpret and act upon. For example, in the bus schedule enquiry domain, instead of specifying the time that he would like to arrive at the airport, a user may provide the departure time of his flight as an implicit arrival time of the desired bus. An additional interpretation process is required in order to represent this type of utterances in the current structure. A TM utterance is an utterance that manages the overall state of the dialogue, rather than manipulates a particular form, for example, suspending or resuming a task. TM utterances can be accommodated in the form-based structure if the definition of the domain is expanded.

| **Domain** | **Rejected utterances (%)** | | | | |
|---|---|---|---|---|---|
| | **OOD** | **OOS** | **Indirect** | **TM** | **Total** |
| Bus schedule | 4.4 | 4.4 | 6.7 | 0.0 | 15.6 |
| Air travel | 1.8 | 4.4 | 0.4 | 2.6 | 9.2 |
| Map reading | 0.0 | 0.0 | 2.2 | 0.0 | 2.2 |
| UAV simulation | 1.0 | 0.0 | 1.0 | 4.0 | 5.9 |

*Table 2*: The number of rejected utterances of each type.

Percentage of rejected utterances in Table 2 reveals that the characteristics of the task affect both the type and the number of rejected utterances. A real-world application (e.g. bus schedule enquiry and air travel planning) has higher rejected utterance rate than a simulated task (e.g. map reading task and UAV flight simulation task). The main difference comes from the scope of the task as reflected in the number of OOS. A simulated task has a limited scope that is well defined while a real task is embedded in the world and appears to have an indefinite domain boundary. Since, a dialogue system applied to a real task will have limited functionality, we would expect it to elicit more OOS utterances.

## 6. Conclusions

We describe a representation for task-oriented conversations based on forms and associated operators, and evaluate its adequacy based on an analysis of conversations in four different domains. The proposed framework accounts for 93% of utterances in task-oriented conversations and can be used to describe two discourse phenomena: domain constraint management and a grounding process.

Acquiring domain information is a resource intensive effort but is a necessary part for building natural language applications on specific domains. The result in [13] shows that one component of the proposed structure, a concept, can be identified in an unsupervised fashion. We are interested in investigating the approaches that would be suitable for learning additional aspects of dialogue structure from a corpus of conversations.

## 7. Acknowledgements

This work was supported under DARPA grant NBCH-D-03-0010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

## 8. References

[1] Grosz, B. J., and Sidner, C. L., "Attentions, Intentions and the Structure of Discourse", *Computational Linguistics*, 12(3):175-204, 1986.

[2] Kamp, H. and Reyle, U., *From Discourse to Logic: Introduction to Model Theoretic Semantics of Natural Language, Formal Logic, and Discourse Representation Theory*, Kluwer Academic Publishers, Dordrecht, 1993.

[3] Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., Anderson, A., *HCRC Dialogue Structure Coding Manual*, Technical Report HCRC/TR-82, HCRC, 1996.

[4] Traum, D. R. and Hinkelman E. A., "Conversation Acts in Task-Oriented Spoken Dialogue", *Computational Intelligence*, 8(3):575-599, 1992.

[5] Searle, J. R., "A taxonomy of illocutionary acts", *Expression and Meaning: Studies in the Theory of Speech Acts*, Cambridge University Press, Cambridge 1979.

[6] Aust, H., Oerder, M., Seide, F. and Steinbiss, V., "The Philips automatic train timetable information system", *Speech Communication*, vol. 17, pp. 249--262, 1995.

[7] Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu W., Oh, A., "Creating Natural Dialogs in the Carnegie Mellon Communicator System", *Proc. of Eurospeech-1999*, Budapest, Hungary, 1999.

[8] Raux, A., Langner, B., Black, A. and Eskenazi, M., "LET'S GO: Improving Spoken Dialogue Systems for the Elderly and Non-natives", *Proc. of Eurospeech-2003*, Geneva, Switzerland, 2003.

[9] Eskenazi, M., Rudnicky, A., Gregory, K., Constantinides, P., Brennan, R., Bennett, C. and Allen, J. "Data Collection and Processing in the Carnegie Mellon Communicator", *Proc. of Eurospeech-1999*, Budapest, Hungary, 1999.

[10] Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G. M., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H. S. and Weinert, R., "The HCRC Map Task Corpus", *Language and Speech*, 34(4):351-366, 1991.

[11] Gorman, J. C., Foltz, P. W., Kiekel, P. A., Martin, M. J., Cooke, N. J., "Evaluation of Latent Semantic Analysis-based measures of team communications content", *Proc. of the 47th Annual Human Factors and Ergonomics Society Meeting*, Denver, Colorado, 2003.

[12] Core, M. and Allen, J., "Coding Dialogs with the DAMSL Annotation Scheme", *Working Notes of AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, Massachusetts, 1997.

[13] Chotimongkol, A. and Rudnicky, A. I., "Automatic Concept Identification in Goal-Oriented Conversations", *In Proc. of ICSLP-2002*, Denver, Colorado, 2002.