# Combining Task Descriptions and Ontological Knowledge for Adaptive Dialogue

Martin Beveridge and David Milward

Cancer Research UK[1], London WC2A 3PX
{mb, dm}@acl.icnet.uk

This paper investigates the use of abstract task specifications for dialogue management in the medical domain. In most current dialogue systems, possible interactions with the system are hand-coded in the design. This is an expensive process, especially for complex dialogues. This paper motivates the use of a task description language for building flexible and adaptive dialogue systems in ontologically rich domains such as medicine. It describes the components of a task specification, and proposes an architecture for dialogue systems which allows integration of domain reasoning and dialogue. A high-level dialogue specification is used to support multimodal input and output, including generation of HTML pages, and generation of fragments of VoiceXML for spoken interaction.

## Introduction

Natural language dialogue systems seem to have great potential for assisting in delivery of healthcare services, but medical applications of dialogue systems are relatively uncommon. Medical applications providing advice to patients or to practitioners pose particular problems relative to standard information seeking dialogues such as flight booking or banking. In most current commercial dialogue systems, the dialogue designer specifies the exact interactions that can take place. This hand-coding allows precise control of what can occur within a dialogue. However, it is an expensive process, especially for complex dialogues where the number of states can be in the hundreds of thousands. In the medical domain, where the knowledge structures are particularly complex, the number of states could run into millions. Moreover, clinical services must employ flexible dialogues based on tasks that are changing dynamically (for example, during the enactment of a clinical guideline) and make use of expert knowledge, complex reasoning and decision-making in responding to the user. It is therefore necessary to integrate technologies such as medical guidelines and advice systems with the dialogue system so that the dialogue can be closely related to the underlying clinical context and goals.

[1] This work was carried-out as part of the EU-funded HOMEY project (Home Monitoring through an Intelligent Dialogue System, IST-2001-32434). Many thanks to our partners for all their helpful comments and advice.

In this paper we explore an approach where dialogues are generated automatically from an *abstract task specification* (ATS), which includes knowledge about the particular domain, and knowledge about tasks that the user might request or be expected to perform. In our current work, the domain knowledge is provided by a medical ontology [1] and the task knowledge is provided by a plan execution and argumentation engine [2] that sends tasks to the dialogue system during the enactment of a clinical guideline.

Once the dialogue system has received a task it needs to decide how next to act. In conventional systems, the dialogue system might output a single new move e.g. a question to a user. However, one of our aims was to support multi-modality, in which case the output might be a whole HTML form rather than a single question. We therefore introduce the notion of a *high-level dialogue specification* (HLDS) that mediates between the task and dialogue structures and provides all the information required by either a graphical or spoken modality. This is then rendered appropriately. For example, HTML rendering may construct one or more forms. VoiceXML rendering may create the first question on the form, but also provide a language model enabling a user to answer any other question on the form. By automatically generating fragments of VoiceXML from the HLDS we allow for greater reconfigurability and more flexible dialogue, but preserve the advantages of VoiceXML in being recogniser and synthesizer independent. The following sections describe the ATS and HLDS in more detail and discuss how they relate to each other and to dialogue structure. We then briefly outline an implementation architecture for this approach.

## Abstract Task Specification

As stated above, one of the aims of this work is to investigate use of an ATS to exploit existing knowledge representation schemas used in medicine, as a basis for generating a dialogue specification automatically. In order to support flexible dialogue, the ATS must provide information of two functional types: information on what task is to be accomplished, and information on the concepts associated with that task and their relations. The former is derived here from execution of a *domain plan* and the latter from a *domain ontology*. These are described below.

### Domain Plan

Consider an application which is concerned with assisting a clinician to follow a structured guideline in some medical domain, such as breast cancer diagnosis. Since the domain plan determines the overall process to be followed, it can be seen as the basis for deriving the intentional structure of a dialogue concerning that domain. In fact, the domain plan may be seen as imposing certain obligations on the dialogue system in order that the plan execution system can achieve successful completion (e.g. first take a patient history then formulate the diagnosis). These obligations will then give rise to intentions on the part of the dialogue system to engage in particular dialogues with the user in order to meet those obligations [3]. This is consistent with the suggestion that dialogue structure is largely determined by task structure [4].

Intentional relations such as those proposed by [4] can be derived from relations between tasks in the domain plan [5]. For example, preconditions of tasks within the plan can be considered to give rise to *satisfaction-precedence* relations in the intentional structure ($I_1$ *SP* $I_2$ if $I_1$ must be satisfied before $I_2$). Hence if task $T_2$ has preconditions such that it cannot be started until task $T_1$ has completed (e.g. it is not possible to make a diagnosis decision until you have completed the patient interview) then a satisfaction-precedence relation can be inferred between the associated intentions $I_1$ and $I_2$ such that $I_1$ *SP* $I_2$. Such dependencies can also arise more indirectly through the data flow, rather than control flow, of the plan. For example, if task $T_2$ has preconditions such that it cannot be started until a data item has a particular value and the goal of task $T_1$ is to acquire a value for that data item, then the same relation can be inferred. *Dominance* relations ($I_1$ *DOM* $I_2$ if $I_2$ provides part of the satisfaction of $I_1$) can similarly be inferred from task decomposition. For example, if task $T_1$ is decomposed into tasks $T_2$ and $T_3$ then the satisfactions of the associated intentions $I_2$ and $I_3$ each form part of the satisfaction of $I_1$. Hence, dominance relations can be inferred such that $I_1$ *DOM* $I_2$ and $I_1$ *DOM* $I_3$.

Note that, whilst obligations imposed by the task specification will provide some of these dialogue intentions, others will arise as a result of the dialogue itself in terms of obligations imposed by the user, e.g. to reply to a "clarification" request or meta-level question. Such obligations will also lead to intentions in the dialogue system, which must then be balanced with those deriving from obligations defined by the task specification. In fact, it is generally assumed that obligations imposed by the user should be processed before those derived from the task specification [6]. Hence, the system must respond to questions from the user before it can continue to pursue domain goals.

### Domain Ontology

It is argued in [7] that task-specific knowledge must be augmented with a conceptual model that describes general information concerning the relationships between objects in a domain. For example, in a library system, [7] suggests a conceptual model in which 'book is-a publication', 'author is-aspect-of publication' etc., hence the type of information typically captured by a domain ontology.

The domain ontology can be seen as forming the basis for deriving rhetorical information relations such as those proposed by [8], e.g. elaboration, circumstance, cause/result etc. These linguistically-motivated relations are, however, domain-independent and are generally at a more abstract level than domain ontological relations. For example an *elaboration* relation between two concepts, $C_1$ and $C_2$ such that $C_1$ *elaborate* $C_2$, might arise from various more specific relations between $C_1$ and $C_2$ such as [8]:

- $C_1$ denotes an attribute of the object denoted by $C_2$
- $C_1$ denotes a member of the set denoted by $C_2$
- $C_1$ denotes a part of the whole denoted by $C_2$

Hence, the required information relations can be inferred from the domain-specific ontological relations between the entities, states and events being discussed.

## High-Level Dialogue Specification

The ATS, as described above, provides a useful abstraction over the task and conceptual knowledge contained in the underlying medical technologies, which can therefore differ between implementations. In order to make use of that knowledge for dialogue, a communication language is required which permits a mapping to be made between task/conceptual structures and structures appropriate to dialogue specification. This is provided here by an HLDS that is based on conversational game theory (CGT).

CGT represents dialogue at two functional levels: at the plan-based level are conversational *games* which are associated with intentions, and at the structural level are sequences of conversational *moves* which specify the linguistic structure required to satisfy those intentions [9]. Dialogues are thought of as being comprised of a series of games each aiming to achieve some sub-goal of the dialogue. Each game itself consists of a series of moves starting with an opening move and finishing with an end move. For example, a request game (such as a request for information) may consist of a request move from the speaker, followed by a reply move by the hearer and optionally an acknowledgement from the speaker to ground the reply. Additionally, a game can have nested sub-games or a break. Sub-games account for phenomena such as clarifications, side sequences etc in which the sub-game contributes to the goals of the parent game. Breaks account for misunderstandings and indicate that either repair is needed in order to continue, or that the current game may have to be abandoned [9].

In our approach, the HLDS describes the games to be played to fulfill obligations imposed by tasks in the ATS. Games seem an appropriate level of description as they represent joint plans for achieving a communicative goal [10] and so can be mapped onto both the task specifications in the ATS, and a low-level specification of the series of communicative acts (moves) that may be made at a given point in the dialogue. This low-level specification can be described as a finite-state machine where moves cause state transitions. Hence, it can be represented by finite-state approaches to dialogue specification such as VoiceXML. The HLDS must therefore contain all the information required to generate a low-level description for the next segment of dialogue. Recent research has proposed three types of information to account for the surface (linguistic) structure of discourse, namely: *intentional structure*, *information structure* and *attentional state*. The representation of these types of knowledge in the HLDS is described below.

### Intentional Structure

Intentions derived from tasks in the ATS are implicitly captured by the structure of games [3, 11], and dominance and satisfaction-precedence relations can be treated as relations between games. The type of a game is in turn determined by its opening (characteristic) move, e.g. a query game is one that starts with a query move. The set of initiating moves employed here are based on those described by [9] and are listed below:

- *Explain*: provide information that wasn't previously requested
- *Instruct*: provide instruction

- *Query-yn*: ask yes/no question for unknown information
- *Query-w*: ask complex question (wh-question) for unknown information

In addition to these initiating moves, the games they initiate will also have response moves. These constitute moves towards satisfying the intention underlying the game. The response moves discussed here are:

- *Acknowledge*: acknowledge and signal continuation
- *Reply-yn*: yes/no reply (in response to a *query-yn*)
- *Reply-w*: reply supplying a value (in response to a *query-w*)

In addition to these response moves, either participant may also respond with the initiating move of a new sub-game whose intention subserves that of the parent game.

### Information Structure

The information structure of discourse can be derived from domain ontological relations specified in the ATS. One of the problems associated with applying information relations to dialogue is determining the appropriate units that such relations should apply to. In text generation, they are applied to successive utterances, but, as Stent points out [12], in dialogue they may span more than one utterance and speaker. For example, consider the dialogue given below in the context of a medical dialogue system which is trying to determine whether a patient with suspected breast cancer should be referred to a specialist or not.

1. *S*: Is there any nipple discharge? [*Query-yn*]
2. *U*: Yes [*Reply-yn*]
3. *S*: Ok… [*Acknowledge*]
4. *S*: And is it bloodstained? [*Query-yn*]
5. *U*: No [*Reply-yn*]
6. *S*: Ok. [*Acknowledge*]

In this example, the second *Query-yn* game (utterances 4, 5 and 6) seeks to elaborate the information provided in the first *Query-yn* game (utterances 1, 2 and 3). This elaboration relation arises because the topic of the second game (bloodstained nipple discharge) elaborates the topic of the first game (nipple discharge). Hence, information relations can be seen as relating games based on domain relations between their associated topics. This relation is, in turn, the basis for the selection of the cue word "and" in utterance (4), and licenses the use of the anaphor "it" to refer to the topic being elaborated. It can also be used to determine the order in which games should be played, in order to ensure the semantic coherence of the dialogue (as described next).

### Attentional State

Since there are likely to be several playable moves at any point in the dialogue, the HLDS should indicate which is the preferred move. Choosing the appropriate move requires a strategy that must take into account the constraints imposed by the intentional and information structures. For example if the intentional structure specifies that $I_1$ *SP* $I_2$ then $I_2$ should not be chosen until $I_1$ is satisfied. Similarly, if $I_1$ *DOM* $I_2$ and $I_1$ *DOM* $I_3$ then it makes sense to address $I_2$ and $I_3$ together as they satisfy a common goal (either by presenting them in succession, or by aggregating them e.g. "what are the patient's age and sex?").

Information relations can also help to preserve dialogue coherence by ensuring that the next move made by the system is as semantically relevant as possible to previous moves in the dialogue history. For example, the intentional structure might describe two intentions $I_1$ and $I_2$ that must both be satisfied, but not specify which should receive focus first. However, given an information structure with a specification such as $I_2$ *elaborate* $I_1$ then it can be seen that $I_1$ is the nucleus of a relation of which $I_2$ is the satellite, and hence $I_1$ should receive focus first. Furthermore, if at any point in the dialogue the user takes the initiative and addresses $I_1$, then $I_2$ should receive focus next, allowing the system to shift to the topic introduced by the user and hence preserving dialogue coherence. E.g. in the dialogue below, once the user introduces the topic of nipple discharge, the system chooses its next move so as to continue the topic:

1. *S*: What is the patient's age?
2. *U*: They're thirty and they have severe nipple discharge
3. *S*: Ok…
4. *S*: And is the nipple discharge bloodstained?
5. *U*: No.
6. *S*: Ok…

In addition, the component which generates the appropriate language model for the speech recogniser will need to know the whole set of games that are playable by either participant, in order to allow mixed-initiative (where the user can address other games than the focused one, or aggregate several reply moves into a single utterance, e.g. "the patient is female, thirty-five"). The notion of attentional state here, unlike that in [4], therefore represents all playable games simultaneously so that the participants can make moves in several games in parallel and move between games [13].

## Implementation

In order to make use of these representations in a practical system we have employed a multi-level architecture (similar to 3-layer hybrid agent architectures [14]) as shown in Figure 1. The deliberative layer is provided by a Domain Manager, which creates an ATS based on the associated domain knowledge components (plan execution and ontology engines). The sequencing layer includes a Game Engine, which determines the games to be played in order to complete the tasks in the ATS and orders them to achieve maximal coherence based on the dialogue history and intentional and information relations. The resulting HLDS is then used by a Move Engine to generate the sequence of low-level communicative acts (moves) that can be made by either participant at the current point in the dialogue. The move specification is expressed in an appropriate mark-up language for the client browser. The reactive layer interprets the move specification in order to play the current game. It also handles low-level behaviour required to support communication (e.g. repeating questions, changing the volume etc).

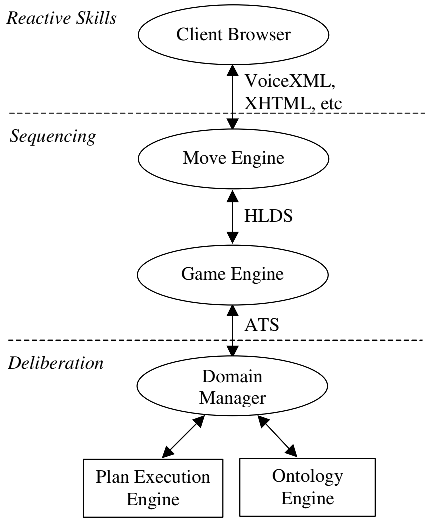


**Fig. 1.** Dialogue system components and their correspondence with 3-Layer Architecture

This architecture is similar to some other recent approaches to providing dialogue based on pre-existing high-level components, e.g. TRIPS [15]. In particular, the Domain Manager and Game Engine here seem to correspond to the Task Manager and Behavioural Agent respectively in [15]. The main difference between this approach and TRIPS, however, is that here the overall dialogue structure is only *partially* determined by task structure - ontological knowledge is also used. This allows the dialogue system to employ information relations between games in order to ensure dialogue coherence, in a similar way to the use of discourse coherence relations in text planning systems [8].

So far, we have employed this approach in the development of a system which interacts with a medical General Practitioner in order to provide advice as to whether a patient with suspected breast cancer should be referred to a specialist or not. The ontology engine in this implementation is provided by Language & Computing n.v. [1] and the plan execution engine is provided by Cancer Research UK's PRO*forma* system [2]. The ATS is implemented via method calls on a Java class whilst an XML language has been developed for representing the HLDS [16]. The client browser can be any voice or web browser (for spoken or visual interaction respectively).

## Conclusions

This paper has described an approach to building spoken dialogue systems that makes use of task and domain knowledge to provide flexible dialogue, but also facilitates the use of current voice-based standards, such as VoiceXML, which are widely employed in commercial systems. Because the high-level dialogue representation can be derived from a domain plan and ontology, there is no need to author dialogues directly, hence providing reconfigurability, as well as allowing greater integration with the application domain and non-dialogue tasks. One further outcome of this work is the development of a (tentative) general framework for adding dialogue capabilities to pre-existing back-end components such as planners, schedulers, databases etc via a principled communication language comprising an abstract task specification and high-level dialogue specification.

## References


1. Ceusters, W., Martens, P., Dhaen, C., Terzic, B.: LinkFactory: an Advanced Formal Ontology Management System. In: Proceedings of KCAP-2001 Interactive Tools for Knowledge Capture Workshop, Victoria B.C., Canada (2001)
2. Fox, J., Das, S.: Safe and Sound: Artificial Intelligence in Hazardous Applications. AAAI Press, Menlo Park, CA, and MIT Press, Cambridge, MA (2000)
3. Kreutel, J., Matheson, C.: Obligations, Intentions, and the Notion of Conversational Games. In: Proceedings of Gotalog '00, 4th Workshop on the Semantics and Pragmatics of Dialogues, Gothenburg, Sweden (2000)
4. Grosz, B., Sidner, C.: Attention, Intention and the Structure of Discourse. Computational Linguistics, Vol. 12, No. 3 (1986) 175-204
5. Young, M. R., Moore, J. D.: Does Discourse Planning Require a Special Purpose Planner? In: Proceedings of the Annual Meeting of the AAAI, Seattle (1994)
6. Traum, D. R., Allen J. F.: Discourse Obligations in Dialogue Processing. In: Proceedings of the 32nd ACL, Las Cruces, NM (1994) 1-8
7. Dahlbäck, N., Jönsson, A.: Integrating Domain Specific Focusing in Dialogue Models. In: Proceedings of EuroSpeech '97, Rhodos, Greece (1997)
8. Mann, W. D., Thompson S. A.: Rhetorical Structure Theory: Towards a Functional Theory of Text Organization. Text, Vol. 8, No. 3 (1988) 243-281
9. Kowtko, J. C., Isard, S. D.: Conversational Games Within Dialogue. Research Paper 31, Human Communication Research Centre, Edinburgh (1993)
10. Hulstijn, J.: Dialogue Games are Recipes for Joint Action. In: Proceedings of Gotalog '00, 4th Workshop on the Semantics and Pragmatics of Dialogues, Gothenburg, Sweden (2000)
11. Maudet, N., Evrard, F.: A Generic Framework for Dialogue Game Implementation. In: Proceedings of the 2nd Workshop on the Formal Semantics and Pragmatics of Dialogue, Twente, Netherlands (1998)
12. Stent, A.: Rhetorical Structure in Dialog. In: Proceedings of the 2nd International Natural Language Generation Conference, Mitzpe Ramon, Israel (2000)
13. Burton, M., Brna, P.: Clarissa: an Exploration of Collaboration through Agent-Based Dialogue Games. In: Proceedings of EuroAIED, Lisbon (1996)
14. Gat, E.: On Three-Layer Architectures. In: Kortenkamp, D., Bonasso, R. P., Murphy, R. (eds): Artificial Intelligence and Mobile Robots. AAAI Press, Menlo Park, CA (1998)
15. Allen, J., Ferguson, G., Stent, A.: An Architecture for More Realistic Conversational Systems. In: Proceedings of Intelligent User Interfaces, Santa Fe, NM (2001)
16. Beveridge, M., Milward, D.: Definition of the High-Level Task Specification Language. Deliverable D11, EU 5th Framework HOMEY Project, IST-2001-32434 (2003)