Linköping Electronic Articles in
Computer and Information Science
Vol. X(1999): nr X

Automaton-Descriptions and Theorem-Proving: A Marriage made in Heaven?

Maria Aretoulaki
and Bernd Ludwig

FORWISS (Bavarian Research Centre for Knowledge-Based Systems)
Erlangen, Germany

Linköping University Electronic Press
Linköping, Sweden

http://www.ep.liu.se/ea/cis/1999/XXX/

Published on Month XX, 1999 by
Linköping University Electronic Press
581 83 Linköping, Sweden

Linköping Electronic Articles in
Computer and Information Science
ISSN 1401-9841
Series editor: Erik Sandewall

Typeset by the author using LaTeX
Formatted using étendu *style*

Recommended citation:
<Author>. <Title>. Linköping Electronic Articles in Computer and Information Science, Vol. X(1999): nr X. http://www.ep.liu.se/ea/cis/1999/XXX/. *Month XX, 1999.*

This URL will also contain a link to the author's home page.

Abstract

In this paper, Finite-State-Automata (FSA) and theorem-proving approaches to spoken dialogue systems (SLDS) are contrasted to each other. FSA are too rigid to deal with unpredictable user reactions, such as corrections or counter-questions, whereas plan-based approaches are usually too complex to be effectively used, given the unreliability of word recognition and the elliptical and unconventional nature of spontaneous speech. As an alternative, a Dialogue Manager architecture is proposed which uses knowledge on both the possible sequences of dialogue acts and the dynamic representation of the task and requirements for its fulfillment. The behaviour of the specific user is taken into consideration, including their expectations about the system and the service offered, as are instances of miscommunication and disagreement in the course of the dialogue, and the successful completion of sub-plans relevant to the task and the dialogue flow.

1 Introduction

In this paper[1], a hybrid approach is presented to the modelling of dialogue management that is adaptive to the current user and the specific communication context. The approach brings together the ubiquitous finite-state automata (FSAs) that describe types of dialogue grammars, and theorem-proving tools. The advantages and disadvantages of both individual approaches have been weighed and each is being called to perform a function that best suits its abilities. FSAs are used for the domain-independent modelling of discourse unit sequences, and inferencing is used to model the domain-specific knowledge necessary to carry out the task. What is important is the interaction between the two sources of knowledge and their dynamic coupling in the course of the dialogue in order to cover its whole structure as it evolves in real time. Thus, there is no need to specify a priori all the possible dialogue flows, but rather these are identified and fleshed out each time a user interacts with the system.

In Section 2, the motivation for the adoption of the hybrid approach is presented, in the context of a corpus analysis with actual human-machine spoken dialogues. In Section 3.1, an overview of FSA approaches is given followed by the introduction of an alternative in Section 3.2. Then in Section 4, the architecture and the operation of the proposed generic and flexible dialogue manager are presented in detail.

2 Lessons from Corpus Analysis

The analysis of actual man-machine dialogues collected with the German train-information system EVAR [Eckert *et al.*1995, Gallwitz *et al.*1998] has shown that users can behave in unpredicted ways, thereby decreasing the chances of the system to correctly interpret their utterances and intentions. For example, in a corpus of 50 dialogues, the percentage with one **user-initiated correction** was relatively high (27,9%) and that representing the cases with two corrections was also substantial (11,6%). In fact, the user often changed the task after all parameters had been specified and the system had retrieved an appropriate database entry (25,5%). Not all corrections took place in the utterance directly after the one with the mistake, which complicates the word recognition process, but also the modelling of the dialogue and the validity of the system predictions about the subsequent user utterance. Equally problematic for the system can be the occurrence of **repetitions** on the part of the user, motivated either by the desire to inform the system about a new specification of the task or to confirm an already-established value along with making a new one known. 32,5% of the above-mentioned corpus contained one repetition, 18,6% three, and another 18,6% two or four repetitions.

Another frequent problem was the **early specification of a new task** by the user: the system is initiating a new subdialogue within the same phonecall without expecting a user initiative at that point. The prespecified dialogue structure does not allow such an initiative. Thus, the user has to repeat the information at least once more. In the above-mentioned corpus, there are frequent instances of the user asking for an earlier or later train connection than the one retrieved by the system, at that point. Un-

[1] Appeared in Proc. of IJCAI-99 Workshop NLP-2: Knowledge and Reasoning in Practical Dialogue Systems, 2 Aug, 1999, Stockholm, Sweden, J. Alexandersson (Ed.), pp. 9–16.

fortunately, this query represents a new task for the system, the previous task record having been wiped out. As a result, the user has to repeat the values of the other parameters (departure and arrival station), if they want to slightly change a single one (e.g. departure time). What is needed, therefore, is a *task memory* for each user, which is indispensable meta-knowledge for the system to have. Equally important is a record of the *dialogue history*, especially of those parameters that have already been confirmed by the user, so that requests for re-confirmations –caused by word recognition errors– are prevented.

Finally, in designing a Dialogue Manager (DMan) one has to consider that it is always possible that the user may pose a query on an **out-of-domain task** or an **out-of-vocabulary word** [Boros *et al.*1997]. The system, apart from identifying the problem, has to accommodate the exceptional nature of the situation and modify the dialogue strategy and the resulting structuring of the dialogue accordingly. This requirement calls for a more flexible dialogue design that consults the pragmatics of the domain, as well.

2.1 Requirements for Spoken Discourse Analysis

The main challenge for every DMan is to match **high-level predictions** about the next user utterance (generated on the basis of the user profile, the domain model, and the current dialogue state) with the possible interpretations of the meaning of the **word lattice** corresponding to that utterance. Spoken language is a semiotic system, for which two essential components can be distinguished [Eco1993]:

1. *Content*: the information that can be expressed
2. *Expressiveness*: the associated vocabulary, phonetics, and syntax

The key problem is how to associate content and its formulations in order to capture the meaning of an isolated expression. That is also the difficulty with pure plan-based approaches. A DMan can only reason about the content of an expression in terms of hypotheses, as the intended content is unavailable. The same holds for speech acts that are assigned to utterances in attempting to classify their function and determine the possible continuation of the dialogue. In this case, speech acts form a semiotic system that carry information about the user's mental state and attitudes. From this point of view, the semantic representation of an utterance serves to reason about the coherence and consistency of the utterance content with respect to knowledge about the domain and the task to be carried out. Speech acts, on the other hand, justify the coherence of the utterance function within the specific discourse, i.e. they explain *why* something has been uttered and *what purpose* should be served, in the speaker's view.

3 Approaches to Dialogue Management

3.1 Finite State Dialogue Models (FSA)

The processing of spoken language is inherently more difficult and complex compared to its written version. This is because speech signals are difficult to recognise, as the same sound sequence may be segmented in more than a single different way, thereby also denoting different meanings. For this reason, the majority of spoken language dialogue systems (SLDS) are limited to function as easy-to-use interfaces to an underlying database, such as train timetables. The database defines various entity types, along with relations that may hold between their instantiations (individuals). In a train information application, for instance, the types would be `Source-City`, `Goal-City`, `Departure-Time`, and `Arrival-Time`. Relations are used to associate individuals to each other; e.g. `connection(Munich, Hamburg, 10:00, 15:26)` expresses the fact that a train leaves Munich at 10:00 and arrives in Hamburg at 15:26. In order to initiate a database query, the system has to have available specific values for these four task parameters. Consequently, the goal of a SLDS user interface is, usually, to obtain these values, which renders the modelling of dialogue structure a database-oriented slot-filling exercise. In such applications, however, the values are not usually dependent on any preconditions, neither is there a temporal or spatial ordering of database types and individuals.

Experience with graphical user interfaces for the development of dialogue applications, such as the *CSLU Toolkit* [Sutton *et al.*1998], has demonstrated the simplicity of the database approach and its inappropriateness for complex time and space-dependent domains, such as those listed in Section 5. The dialogue flow is represented by a series of ordered states which are interconnected with arrows, denoting the –limited number of– alternative transition possibilities in any one dialogue. Such a model restricts the user's freedom to focus on any theme that is not predicted to be relevant, and can, thus, further complicate and aggravate the speech recognition process. Apart from rendering the system anything but user-friendly, this approach leaves no room for user initiative in the course of the dialogue.

The alternative approach of automatically discovering n-grams of speech acts on the basis of corpora (e.g. [Moeller1997, Passonneau and Litman1997, Reithinger and Maier1995]) is by necessity too domain- and corpus- dependent to be of interest for our goals. The point is that the DMan should not have to be restructured or have more than one of its modules respecified every time it is customised to a new application and domain. Thus, a plausible answer seems to be the one proposed in Section 3.2.

3.2 Inference-Based Approaches

A change is advocated here from the Dialogue-State-Transition model described in Section 3.1 to a more flexible one based on speech act recognition and a well-defined notion of coherence: *coherence* between utterances in a dialogue is determined on the basis of the coherence between their respective *pragmatic effects* (cf. [Jokinen1994, McKevitt *et al.*1992, Smith and Richard Hipp1994]). Consequently, there is no need to categorise each dialogue turn according to a priori labels, as the case is in FSA models. The domain model will provide guidance as to the salience of the various topics and subtopics dealt with. In addition, the advocated approach to dialogue can handle phenomena such as the occurrence of misunderstandings, because system knowledge is not restricted to the application domain, but also extends to the discourse itself and its conventions. Using such a framework, **generic SLDS** can be designed which can accommodate a greater number of applications which are more sophisticated than those implemented by systems currently available. More importantly, more user-oriented systems can emerge, which can better simulate the communicative behaviour exhibited in human-human interaction. In Section 4, this idea is fleshed out in greater detail.

4 The Hybrid Dialogue Manager

Both the rigidity of FSA approaches and the cumbersome and subjective nature of pure plan-based processing have been hinted at. The purpose of this paper is to put forward a hybrid solution to the design of DMans that are more generic, flexible and user-centred.

In general, the role of every DMan should be threefold:

1. to **interpret** the user utterance
2. to **generate expectations** about the continuation of the dialogue
3. and to **plan an appropriate response**, both in terms of content and form

In order for the DMan of a SLDS system to be able to carry out these tasks, a number of knowledge sources have to be available to the system: information about the application and the domain, on the one hand, and the current user profile (goals, plans, expectations) and their behaviour in the course of the interaction with the system, something that includes awareness of the frequency of misunderstandings, user-initiated corrections, and indispensable repetitions (dialogue history). Moreover, the use of specific utterance formulations rather than others should be exploited to interpret user intentions and attitudes. In Section 4.2, a DMan architecture is proposed which uses these types of knowledge effectively and efficiently. First, in Section 4.1 some key concepts are introduced on which part of the implementation is based.

4.1 Knowledge Representation and Reasoning

4.1.1 Formal Grammars

In analogy to the parsing of spoken language, formal context-free or even regular grammars using constraints of various types are employed within the proposed system to represent knowledge about clear ordering in time (e.g. sequences of speech acts). Such grammars form the FSA part of the DMan and are processed by appropriate chart parsing algorithms, as will be explained later.

4.1.2 Description Logics (DL)

For the representation of the semantics of natural language, DL is employed, a specific sublanguage of first-order logic. In DL there exist unary relations for the definition of concepts related to objects and actions, whereas binary relations describe how concepts are associated with each other (roles). Such a logics-based language should be well-suited for this task, because natural language functions in a similar way: noun phrases define objects, verbal phrases actions, modifiers give more specific meaning to phrases. Grammatical relations (i.e. syntax) define how phrases depend on each other, thereby taking semantic and pragmatic constraints into account. In order to interpret a user utterance in a dialogue context the DMan, in close cooperation with the speech recogniser and the parser, has to construct a semantic representation of the current utterance on the basis of knowledge about the syntax, the semantics, and the pragmatics of the application. This task can be carried out effectively and efficiently using the reasoning processes provided by DL (see DRT and Section 4.4).

4.1.3 Discourse Representation Theory (DRT)

DRT [Kamp and Reyle1993] offers a way to store the semantic representation of a whole discourse, and not just a single utterance. For this reason, it is employed in the system as the representation formalism for all utterances. In DRT, a discourse is stored in a Discourse Representation Structure (DRS) of the form

$$\left[\frac{d_1, d_2, ..., d_n}{\text{Cond}(d_1, d_2, ..., d_n)}\right]$$

where $\text{Cond}(d_1, d_2, ..., d_n)$ is a set of DL propositions about the set of *discourse referents* d_1, d_2, ..., d_n. The (model theoretic) semantics of a DRS defined by

$$\exists d_1, d_2, ..., d_n : \text{Cond}(d_1, d_2, ..., d_n)$$

is an extension of the corresponding terminological DL concept definitions. The possibility to store discourse referents of the whole discourse (in contrast to a single utterance) provides a dialogue history and a number of processes for anaphora and ellipsis resolution. The meaning of a DRS is that the propositions related to the introduced discourse referents are true, i.e. extensions are made to the application-dependent concepts and roles, defined in the Domain Module (see TM in Section 4.2). The parser transforms every word lattice into a lattice each of whose edges is annotated with a DRS corresponding to the phrase represented by that edge.

4.2 The Architecture

A DMan that is sufficiently flexible to deal with unpredictable user reactions and intelligent to react in congruence with the pragmatics of the domain and the specific interaction history should consist of the following modules:

- a *Dialogue Module* (DM)
- a *User Model* (UM)
- a *World Knowledge Module* (WM)
- a *Task / Domain Module* (TM)
- a *Message Planner* (MP)

In Fig. 1, Linguistic Interface is the module that intermediates between the parser and the DMan. It provides the Dialogue Module with a DRS representing the syntax and the semantics of the current utterance. It also furnishes a history of linguistic expressions of the current discourse.

4.2.1 Dialogue Module (DM)

DM is the most central part of the whole DMan. It is here that the dialogue flow is modelled in the form of actions and reactions. It is also here that the current Dialogue State is recorded. What is important, however, is that the resulting dialogue structure is not too restrictive and can allow for unpredictable user reaction (Section 4.3). For example, it should be possible for a system question about a task parameter (action) to be followed by a user counter-question about the available value options (reaction) (cf. Fig. 2). Apart from such option listings, a counter-question can be a request for help, clarification, or explanation. It should be noted that DM provides predictions to the LI module, in order to facilitate the word recognition process for the next utterance.

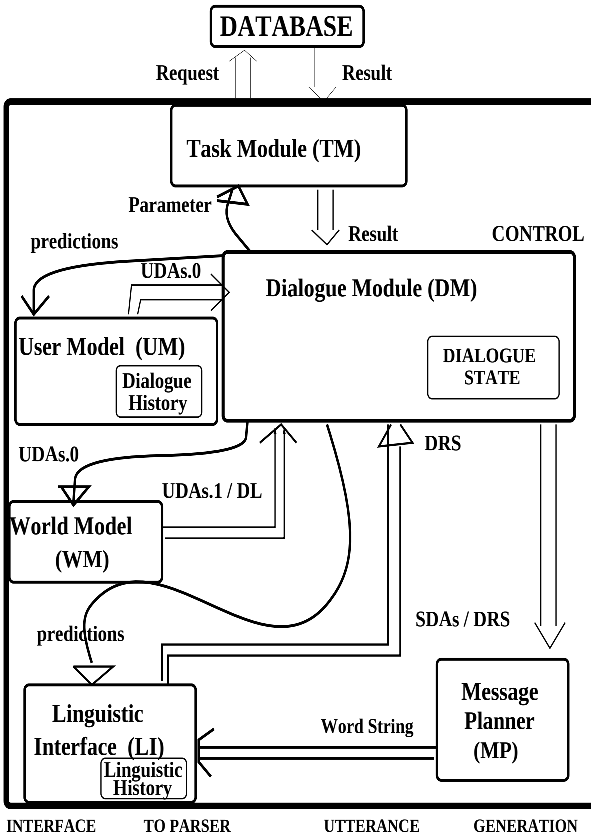

Figure 1: The Generic Dialogue Manager

DM is also where a *Task Memory* is kept, so that slight changes in the user's requirements do not result in a re-specification of all parameters in a new sub-dialogue (Section 2).

4.2.2 User Model (UM)

UM contains knowledge about users of the system, in general, as well as information about the current user as this can be dynamically inferred and augmented in the course of the specific transaction. Thus, this module contains data about the following: **(1) General domain knowledge** that the user may or should have. Again, this knowledge is represented in DL terms. Therefore, the same reasoning procedures can be applied to information from both UM and the World Model (see below). The knowledge stored here is not static, but is updated with information about the user's assumptions gathered while interpreting their utterances. Thus, the DMan can determine differences between the user's and the system's domain knowledge (potentially leading to misunderstandings). **(2) Current task requirements** are represented in UM in the form of DRS (i.e. extensions of the terminological knowledge about the domain), as well as the resulting dialogue obligations that the user has towards the system (e.g. to provide information about other related aspects of the task in order for the system to be able to fulfill the user requirements). It is here that *corrected* or changed task parameter values are recorded, thereby making up a Task Memory for the current sub-dialogue (Section 2). On the whole, the exis-

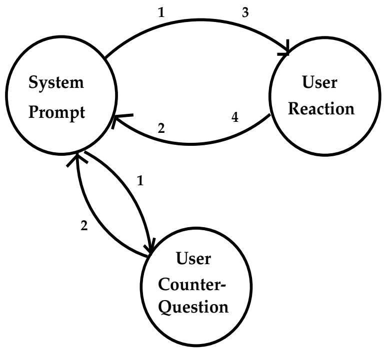

Figure 2: The Dialogue Strategy

tence of UM is indispensable in identifying communication problems and dissatisfaction on the part of the user, which in turn can activate appropriate dialogue strategies for their solution. This functionality of UM stems from its constant interaction with the corresponding data in the Task and Dialogue Modules of the DMan. In Fig. 1, it is shown how DM predictions help in the generation of potential User Dialogue Acts (*UDAs.0*), which will be later assessed by the World Model.

4.2.3 World Knowledge Module (WM)

WM contains facts and rules about application-independent commonsense knowledge for disambiguation and pragmatic interpretation purposes. This knowledge is represented using DL terminology. More specifically, the following are covered: **(1) Time** and **location** specification, i.e. the translation of anaphoric or underspecified expressions into their exact referents (e.g. "In four days' time" would become "On Friday the 6th of August"). **(2)** A **generic model of communicative behaviour**: here information about the overall goals and requirements of the system is kept, as well as about the overall goals and needs of users in general, their assumptions about the system, and their expectations about the service provided by the system (cf. the so-called 'Background Knowledge' in [Dybkjaer *et al.*1996]). In addition, assumptions about the communicative behaviour of dialogue participants are also modelled in WM; e.g. the principles of sincerity and cooperation (cf. Section 4.3). Assumptions about interaction behaviour have already been used by other researchers to integrate new utterances in a existing dialogue structure (cf. [Jokinen1994]). WM is generic in the sense that it does not contain any information about actual dialogues and their participants. This information is represented in the User Model and the Task Module. As shown in Fig. 1, the preliminary UDAs.0 provided by DM are interpreted using world knowledge within this module to result in the 'actual' or intended User Dialogue Acts (UDAs.1).

4.2.4 Task / Domain Module (TM)

TM is customizable to the current application and is not reusable in the system. It contains a definition of the domain concepts involved, as well as links to the related vocabulary, as the basis for associating expression (of utterances) and content (of the domain model). Also included in TM is information about how the terminology corresponds to the system's problem solver (e.g. a database) and how there can be a translation between

DL statements and problem solver expressions (e.g. SQL queries). It has already been noted that task and domain knowledge is closely coupled with the discourse structure in this system, so that dialogue management takes the pragmatics of the situation into consideration and not just preset expectations about the flow of the dialogue. How this takes place is further explained in Section 4.4.

4.2.5 Message Planner (MP)

MP is where the next system utterance is planned in terms of both content and actual formulation. The content is established on the basis of information about the required System Dialogue Acts (SDAs) retrieved in the form of a DRS from the Dialogue Module (DM), which acts like a "blackboard" where all modules can record their intermediate processing results for consultation by the other modules. DM establishes whether the task specification is complete and a database search can be started or the planning of its fulfillment can be initiated. MP then activates the realisation of an appropriate system sub-goal, which can be a suggestion, a further question, or the actual search of the application database and the presentation of the result to the user. It is not so important whether generation is based on actual synthesis or concatenation (although the prosodic features of the system utterance can have a decisive effect on the user and their subsequent behaviour). What matters most is the planning of the content of the system utterance, so that it appears to be coherent and cohesive to the context of the preceding dialogue. The coupling of the discourse with the task ascertains this coherence, as is explained in Section 4.4.

4.3 Grounding and Dialogue Segmentation

The modelling of the possible structuring of the dialogue in DM thus depends on a number of different factors. The quality of word recognition is the first one. If the system cannot understand what the user has said, then special repair and clarification strategies have to be activated in order to re-establish communication with the user (Meta-communication principle in [Dybkjaer *et al.*1996]). A failure to do so at that point can mean that the user loses patience and hangs up. This is the process of **grounding** [Traum1994]. Repair strategies can range from requests for repetition or confirmation, to requests for spelling of the controversial task parameter. Another way to establish the possible continuation of the dialogue is the *generation of predictions* on the part of the system. These expectations can augment the word recognition process. Associated with this last factor is some type of dialogue record that holds a history of conflicts and misunderstandings between the system and the specific user in the course of the dialogue (*dialogue behaviour history* in Section 2). In this case, too, the system has to be more conservative in its reactions and more attentive to user initiatives after a misunderstanding has already occurred.

Every grounded utterance has to be integrated in the discourse. As often noted in the literature on discourse theory and SLDS implementations, for this integration speech acts have to be assigned to the utterance so that it can be hypothesised which part in a discourse plan about communication the user is probably executing at this point in time. A limited number of speech acts is used in this system: **greeting, query, inform, accept, reject, suggest** (command to the hearer). This set suffices to model the discourse structure of information retrieval dialogues. The underlying idea was to use speech acts that are mainly motivated linguistically and not overloaded with pragmatic information, thereby rendering their automatic identification more straightforward. Perlocutionary and illocutionary effects of utterances are better determined by looking how the content of an utterance coheres with domain-related concepts and pragmatic plans and with communicative behaviour, as defined in TM and UM.

In the literature, sequences of speech acts are often called *conversational games* which prescriptively and descriptively explain the coherence of several turns in a dialogue providing the motivations of a dialogue participant in acting in a specific way at the given point in time. In this system, a context-free grammar is used to model the set of all permissible conversational games. This grammar has to be context-free in order to allow counter-questions and clarifications (Fig. 2). The grammar rules can be used to generate predictions about the speech acts to follow as the next step in the currently applied discourse plan. The grammar is annotated with constraints that incorporate the knowledge represented in UM and TM in order to select a conversational game for the current utterance.

To clarify this process an example game rule will be discussed here. An initial definition of the *information retrieval game* (irg) would be:

$$\begin{array}{lll} \text{irg} & \longrightarrow & \text{query}_I(\phi)\,\text{inform}_R(\xi) \\ & & \mathcal{B}_R(\xi \rightarrow \phi) \end{array}$$

This rule states that a Query act by participant I(nitiator) with propositional content ϕ can be followed by an Inform act by R(esponder) with content ξ. Such an act sequence, however, is only permissible, if the constraint $\mathcal{B}_R(\xi \rightarrow \phi)$ holds, that is R has to believe that ξ implies ϕ, i.e. that their utterance is an answer to I's query. The validity of $\xi \rightarrow \phi$ is determined on the basis of the knowledge R is supposed to have, i.e. on information held in UM. Thus, the coherence between the content of two utterances is a precondition for the coherence between the speech acts expressed therein. In addition, by evaluating the rule constraints, the context of the current dialogue is also taken into account.

What motivates R to react with an Inform act to I's query? DM contains the knowledge that the reason for $\text{query}_I(\phi)$ is $\mathcal{W}_I(\mathcal{K}_I(\phi))$, i.e. that I wants to know whether or under which circumstances ϕ is true. Thus, the DMan understands ϕ as a query for information. Being **co-operative** (as defined in DM), the DMan assumes the user's discourse goal (to receive an answer to ϕ) as its own: $\mathcal{W}_R(\mathcal{K}_I(\phi))$. As the DMan is a **sincere** dialogue participant that has no information to hide, it will generate an $\text{inform}_R(\xi)$ speech act, if reasoning that has taken place in TM and WM identified a ξ with $\xi \rightarrow \phi$, namely if an answer can be found for ϕ. The same basic assumptions about communicative behaviour can also be found in the UM, where they are used to generate hypotheses about the possible speech acts expressed by various utterances. Consequently, a revised definition of the information retrieval game (irg) would be:

$$\begin{array}{lll} \text{irg} & \longrightarrow & \text{query}_I(\phi)\,\text{inform}_R(\xi) \\ & & \mathcal{B}_R(\xi \rightarrow \phi) \\ & & \mathcal{W}_R(\mathcal{K}_I(\phi)) \end{array}$$

For a detailed account of how the system computes its own dialogue goals, see [Ludwig *et al.*1998].

4.4 The Dialogue Manager at Work

In this section, a detailed example is presented of how the DMan processes an information retrieval dialogue in a flight information application.

α **User:** I want a flight from Athens.
β_1 **System:** Where do you want to go to?
γ_1 **User:** To Rome.
β_2 **System:** On which day?
γ_2 **User:** On Monday.
β_3 **System:** When do you want to depart?
γ_3 **User:** At 12 o'clock.
ρ **System:** There are several connections: AZ717 at 06:55, OA233 at 09:05, AZ725 at 09:20, OA235 at 12:55, AZ721 at 16:00, OA239 at 17:10, AZ723 at 20:20.

Utterance a refers to the following concept defined in the domain model (here as a DL representation):

$$\begin{aligned}\textsc{FlightFromTo} \quad = \quad & \forall\textsc{From.airport} \sqcap \\ & \forall\textsc{To.airport} \sqcap \\ & \forall\textsc{At.Time} \sqcap \\ & \forall\textsc{On.Day} \sqcap \\ & \textsc{Flight}\end{aligned}$$

We ignore the parsing process that constructs DRS a (Fig. 4) and go on to discuss how the discourse is interpreted by the DMan.

The performative modal verbal phrase *I want* explicitly expresses a desire and is therefore represented as $\mathcal{W}_I(\alpha)$. Reasoning about communicative behaviour and its connection with speech acts produces: $\mathsf{query}_I(\alpha)$. According to conversational games rules, this induces a response expectation for R (the system). This obligation is expressed by $\mathcal{W}_R(\mathcal{K}_I(\alpha))$. In order to fulfill this desire, the DMan tries to evaluate the consequence status of α. This evaluation is performed on the basis of the domain model. Its result is that the status of α depends on information about the concepts $\{\textsc{At}, \textsc{To}, \textsc{On}\}$; i.e. departure day and time, as well as the destination are still unknown. Thus, the mental attitudes of DMan change: a new desire $\mathcal{W}_R(\mathcal{K}_R(\beta_1))$ is added which implies $\mathsf{query}_R(\beta_1)$; a new discourse obligation is introduced leaving the old one pending. The user's response is constative(γ_1), implying $\mathcal{W}_I(\mathcal{K}_R(\gamma_1))$. From γ_1 DMan infers $\gamma_1 \to \beta_1$. Consequently, it assumes $\mathcal{B}_I(\gamma_1 \to \beta_1)$ and $\mathsf{inform}_I(\gamma_1)$. As there is no evidence against $\mathsf{inform}_I(\gamma_1)$ and no other hypothesis for a possible speech act, $\mathsf{inform}_I(\gamma_1)$ is assumed. This completes $\mathsf{query}_R(\beta_1)$, whereby DMan obtains information about the flight destination. The processing of β_2, γ_2 and β_3, γ_3 is performed analogously. After that, the justification context inferred during the interpretation α has been instantiated fully. Having acquired all this information, DMan can compute a solution for α, formulated in ρ. Fig. 3 sketches the coherence structure of all utterances of the example dialogue, represented in a DRS in Fig. 4.

5 Future Work

At the level of semantics and pragmatics, a generic way of achieving a tight coupling with pragmatic components apart from databases has still to be

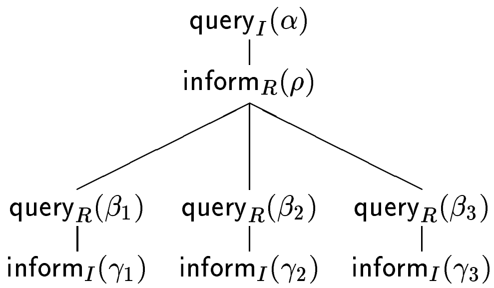

Figure 3: Coherence structure of the Example Dialogue

worked out. The overall aim is to enable people other than the original developers of DMan to customize it to new applications based on a number of heterogeneous problem solvers. Another area of future work is the further exploitation of the modelling of the dialogue participants integrated in the system. In particular, the user model should be extended to cope with prosodic information, which is useful both for syntactic and semantic analysis and for determining the "non-logical" (e.g. the emotional) state of the user. Related research will be fundamental for the incorporation in the system of a **Dialogue Behaviour History**, i.e. a record of the specific interaction style of the current user, as well as of the degree of co-operativity and agreement (as opposed to conflict and rejection) that the user has been exhibiting in the course of the specific dialogue with the system. This would enable the system to adapt its communicative behaviour according to that exhibited by the user and to always try to react in a polite and considerate manner. Finally, further work is planned on the DMan's discourse strategies. It is very important to communicate misunderstandings, corrections, and alternative interpretations or responses to utterances in a transparent and easily understood way. If the user understands precisely what the system is "thinking", they will be able to cooperate more towards the desired answer by being flexible. Transparent dialogue behaviour is indispensable in increasing the robustness of a dialogue system in real-world applications for which a great demand already exists. For example:

- a Communication Aid for disabled persons which initiates a message sending process or a dialogue with someone, depending on the owner's current mood, physical condition, topic and communication preferences in order to select the speaking / writing style to be used;
- a Language Learning System which keeps a record of the pupil's rate of progress, current knowledge and abilities, and learning goals, in order to select appropriate teaching material (set exercises vs newspaper cuttings) and means of presentation (text-only, pictures, animation).

6 Conclusion

In this paper, an integrated approach to dialogue management was presented that ensures flexibility and naturalness, as well as portability for the resulting spoken dialogue systems (SLDS). The simplistic Finite-State-Automaton (FSA) approach to the modelling of communicative acts is combined with a dynamic domain space walkthrough for constraint satisfaction and coherence establishment.

The outlined approach presents a number of advantages over the FSA

$$
\left[\begin{array}{l}
\alpha\,\beta_1\,\beta_2\,\beta_3\,\gamma_1\,\gamma_2\,\gamma_3\,\rho \\
\hline
\alpha = \left[\begin{array}{l}
\mathrm{u}\,\tilde{\alpha} \\
\hline
\tilde{\alpha} = \left[\begin{array}{l}
\mathrm{f}\,\mathrm{Athens} \\
\hline
\textsc{Flight}(\mathrm{f}) \\
\textsc{From}(\mathrm{f}, \mathrm{Athens})
\end{array}\right] \\
\mathrm{want}(\mathrm{u}, \tilde{\alpha})
\end{array}\right] \\
\beta_1 = \lambda l. \left[\begin{array}{l}
\mathrm{u} \\
\hline
\textsc{Go}(\mathrm{u}) \\
\textsc{Location}(l) \\
\textsc{To}(\mathrm{u}, l)
\end{array}\right] \\
\gamma_1 = \beta_1(\mathrm{Rome}) \\
\beta_2 = \lambda d. \left[\begin{array}{l}
\mathrm{u} \\
\hline
\textsc{Go}(\mathrm{u}) \\
\textsc{Day}(d) \\
\textsc{On}(\mathrm{u}, d)
\end{array}\right] \\
\gamma_2 = \beta_2(\mathrm{Monday}) \\
\beta_3 = \lambda t. \left[\begin{array}{l}
\mathrm{u} \\
\hline
\textsc{Go}(\mathrm{u}) \\
\textsc{Time}(t) \\
\textsc{At}(\mathrm{u}, t)
\end{array}\right] \\
\gamma_3 = \beta_3(12\ \mathrm{o'clock}) \\
\rho = \left[\begin{array}{l}
\mathrm{c}\ \mathrm{AZ717}\ \mathrm{OA233}\ \mathrm{AZ725}\ \mathrm{OA235}\ \mathrm{AZ721} \\
\mathrm{OA239}\ \mathrm{AZ723} \\
\hline
\textsc{Flight}(\mathrm{AZ717})\ \textsc{At}(\mathrm{AZ717}, 06:55) \\
\textsc{Flight}(\mathrm{OA233})\ \textsc{At}(\mathrm{OA233}, 09:05) \\
\textsc{Flight}(\mathrm{AZ725})\ \textsc{At}(\mathrm{AZ725}, 09:20) \\
\textsc{Flight}(\mathrm{OA235})\ \textsc{At}(\mathrm{OA235}, 12:55) \\
\textsc{Flight}(\mathrm{AZ721})\ \textsc{At}(\mathrm{AZ721}, 16:00) \\
\textsc{Flight}(\mathrm{OA239})\ \textsc{At}(\mathrm{OA239}, 17:10) \\
\textsc{Flight}(\mathrm{AZ723})\ \textsc{At}(\mathrm{AZ723}, 20:20)
\end{array}\right]
\end{array}\right]
$$

Figure 4: Discourse Representation Structure (DRS) for the Example Dialogue

approach. Firstly, the structure of the dialogues processed by the dialogue manager is no longer predefined by the designer down to the level of word sequences that evoke the transition from one state of the dialogue flowchart to the next. In the proposed approach, the structure grows dynamically on the basis of a coherence relation between utterances. The structure is restricted at the level of speech acts, a more abstract level of discourse compared to that used in the FSA approach. It is exactly because of this design choice that the system is able to process and appropriately react to user counter-questions. Equally important is the fact that the separation between function and content allows for the design and implementation of at least some knowledge sources that are independent of the specific domain and application of the system. At the same time, content will always be coupled to dialogue function in the course of utterance and discourse interpretation by means of the pre-defined conversational games.

High-level knowledge (such as user profiles, repair and conflict history, conversational games, and task plans and goals) is associated with its surface expressions, thereby setting up the groundwork for the development of systems that are adaptive to the current user and communication situation, but still sufficiently efficient to operate in real time. As a result, all modules in this dialogue manager are reusable across different dialogue applications, with the obvious exception of the knowledge base defining the domain world and the corresponding entities and their interrelations.

The proposed Dialogue Manager is currently under development. Preliminary work has shown that although the inference processes are computationally intensive, the DL algorithms employed are sufficiently fast for the system to operate in real time. Equally importantly, efficiency is further increased by the massive reduction of the number of generated hypotheses, which results from the application of these reasoning procedures.

References

[Boros *et al.*1997] M. Boros, M. Aretoulaki, F. Gallwitz, E. Nöth, and H. Niemann. Semantic Processing of Out-Of-Vocabulary Words in a Spoken Dialogue System. In *Proceedings of the 5th European Conference on Speech Communication & Technology (EUROSPEECH '97)*, volume 4, pages 1887–1890, Rhodes, Greece, 1997.

[Dybkjaer *et al.*1996] L. Dybkjaer, N. O. Bernsen, and H. Dybkjaer. Grice incorporated: Cooperativity in spoken dialogue. In *Proceedings of COLING'96*, volume 1, pages 328–333, Copenhagen, Denmark, Aug. 5–9 1996.

[Eckert *et al.*1995] W. Eckert, E. Nöth, H. Niemann, and E.-G. Schukat-Talamazzini. Real users behave weird – experiences made collecting large human-machine-dialog corpora. In P. Dalsgaard, L. B. Larsen, L. Boves, and I. Thomsen, editors, *Proceedings of the ESCA Tutorial and Research Workshop on Spoken Dialogue Systems*, pages 193–196, Vigso, Denmark, June 1995.

[Eco1993] U. Eco. *La Ricerca della Lingua Perfetta nella Cultura Europea.* Laterza, Roma, 1993.

[Gallwitz *et al.*1998] F. Gallwitz, M. Aretoulaki, M. Boros, J. Haas, S. Harbeck, R. Huber, H. Niemann, and E. Nöth. The Erlangen spoken dialogue system EVAR: A state-of-the-art information retrieval system. In *Proceedings of the 1998 International Symposium on Spoken Dialogue (ISSD'98)*, pages 19–26, Sydney, Australia, 1998. Invited paper.

[Jokinen1994] K. Jokinen. *Response Planning in Information-Seeking Dialogues.* PhD thesis, Centre for Computational Linguistics, University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., October 1994.

[Kamp and Reyle1993] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer, Dordrecht, 1993.

[Ludwig *et al.*1998] B. Ludwig, G. Görz, and H. Niemann. User models, dialog structure, and intentions in spoken dialog. In *Proc. von 4. Konferenz zur Verarbeitung natürliche Sprache (KONVENS'98), Thema: Computer, Linguistik und Phonetik zwischen Sprachen und Sprechen, 5–7 Oktober*, Universität Bonn, 1998.

[McKevitt *et al.*1992] P. McKevitt, D. Partridge, and Y. Wilks. Analysing Coherence of Intention in Natural Language Dialogue. Technical Report 227, Dept. of Computer Science, University of Exeter, Exeter, U.K., 1992.

[Moeller1997] J-U. Moeller. Dia-Mole: An unsupervised learning approach to adaptive dialogue models for spoken dialogue systems. In *Proceedings of the 5th European Conference on Speech Communication & Technology (EUROSPEECH '97)*, volume 4, pages 2271–2274, Rhodes, Greece, 1997.

[Passonneau and Litman1997] R. J. Passonneau and D. J. Litman. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139, March 1997.

[Reithinger and Maier1995] N. Reithinger and E. Maier. Utilizing statistical dialogue act processing in VERBMOBIL. Technical Report VM Report 80, DFKI GmbH, Saarbruecken, Germany, May 1995.

[Smith and Richard Hipp1994] R. W. Smith and D. Richard Hipp. *Spoken Natural Language Dialog Systems: A Practical Approach.* Oxford University Press, New York, 1994.

[Sutton *et al.*1998] S. Sutton, R. Cole, J. de Villiers, J. Schalkwyk, P. Vermeulen, M. Macon, Y-H. Yan, E. Kaiser, B. Rundle, K. Shobaki, P. Hosom, A. Kain, J. Wouters, D. Massaro, and M. Cohen. Universal speech tools: The CSLU Toolkit. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, volume 7, pages 3221–24, Sydney, Australia, 1998.

[Traum1994] D. Traum. *A Computational Theory of Grounding in Natural Language Conversation.* Ph.D. thesis, Dept. of Computer Science, University of Rochester, Rochester, 1994.