Abstract

The paper outlines our dialogue model which comprises utterance analysis, processing and generation in a DRT framework. We present and discuss in detail an example from the EMBASSI domain\(^1\).

1 Introduction

The challenges for speech dialogue systems are manifold: spontaneous speech phenomena at telephone quality, robustness at the language and interaction levels, partial interpretation, etc. There are also requirements for cooperative behaviour and negotiation abilities. Nearly all working systems, including the Erlangen System EVAR (see [Gallwitz et al. 1998]), use a structural approach, i.e.

the dialogue is modelled by some kind of finite state machine. An evaluation of more than 1000 train timetable dialogues with EVAR showed that out of 31.5% of incomplete dialogues, 83.1% failed be-
dcause of a failure to integrate utterances into the con-text! There is an obvious lack of flexibility for dynamic tasks with a high degree of context-
dependency. Furthermore, it is an open question how meta-discourse, which occurs frequently in dia-
logue corpora, can be accommodated for. And there are further limitations: The dialogue models are tai-
lored to a certain application type which makes port-
ing and scaling up nearly impossible. So, a lot of
work has concentrated on analysing communicative
plans to overcome these shortcomings. In the fol-
lowing, we show the advantages of plan-based di-
alogue modelling for rational dialogues, stressing
the usefulness of distinguishing between proper dia-
logue features and task-specific ones. For the system

architecture, our dialogue model enables, orthogo-
nally, a division between an application-dependent problem solver and a generic dialogue manager – a
long-term goal of our research.

In a dialogue, every user utterance must be analy-
sed in different aspects, in the assumption that it simultane-ously establishes coherence on several dis-
tinct levels of abstraction. As a consequence, these aspects help to understand in which way a partic-
ular utterance contributes to the current dialogue.

A salient feature of our approach is that it takes
a “pragmatics-first” perspective, in the sense that
discourse pragmatic and application-specific con-
straints play a key role in controlling the syntactic
and semantic levels of analysis and generation.

2 An Example Dialogue

The EMBASSI project delegates the knowledge of how to use electronic devices from the user to soft-
ware, as in the following typical dialogue:

U\(_1\) : Welche Filme kommen heute abend? (What films are on this evening?)
S\(_2\) : Krimis oder Spielfilme? (Thrillers or popular films?)
U\(_3\) : Ich würde gerne eine Komödie sehen. (I'd like to watch a comedy.)
S\(_4\) : Um 21:45 beginnt “Spion in Spitzenhöschen”. (“Spion in Spitzenhöschen” is on at 21:45.)
U\(_5\) : Und Krimis? (and thrillers?)
U\(_7\) : Bis wann läuft der? (How long is that on for?)
S\(_8\) : [Er dauert] Bis 22:15. ([It is on] Till 22:15.)
U\(_9\) : Dann möchte ich den Spielfilm aufnehmen. (In this case, I’d like to have the film taped.)
S\(_10\) : Ok. Der Videorecorder ist programmiert auf Sender BR3 von 21:45 bis 23:05. (Ok. The recorder has been set to tape channel BR3 from 21:45 to 23:05.)

---

\(^1\)EMBASSI is a joint project sponsored by the German Federal Ministry of Research with the goal to develop systems for multimodal assistance for device operation. Our group works on the speech dialogue control of an A/V home theatre. Only the authors are responsible for the content of this paper.
The next section introduces a dialogue model which is applied to the dialogue above in Sect. 5.

3 Levels of Utterance Analysis

Some of the utterances cohere with the dialogue (as far as it has evolved up to that point) for application pragmatic reasons (e.g. $S_4$, $U_2$), while some utterances are coherent in terms of dialogue pragmatics (e.g. $U_5$, $S_9$)$^2$. In each case, a particular level of the utterance explains how and why it is coherent. In the following sections, the different levels will be introduced and characterised. They are used in a computational dialogue model as follows: Given a new utterance, its satisfaction state for each level is computed on the basis of the information available and the background knowledge about the dialogue and the application. Under the assumption that all knowledge is partial, three different possible satisfaction states (unique, ambiguous, none) for each level suffice to explain how the new utterance coheres with the current dialogue. By processing a dialogue it is attempted to obtain a unique satisfaction state for each level in order to define and execute a unique task and to update the information state.

3.1 Syntactic Level

The syntactic level involves parsing a lattice of word hypotheses, annotated with prosodic features as provided by the speech recogniser. Instead of deriving complete syntactic derivations spanning the whole input, edges in the lattice are grouped in syntactic chunks [Abney 1991] in a first step.$^3$ Then the combination of these fragments into bigger units is governed by semantic and pragmatic constraints, the applicability of which is checked by valency and case frames of the lexical units. So, the assignment of grammatical functions is determined by the valency of the respective lexical units.

3.2 Semantic Level

A phrase such as the film on ZDF is assumed to be syntactically unique if we use appropriate grammar rules and lexical entries. However, these are two possibilities for the attachment of prepositional phrase on ZDF in the sentence Tape the film on ZDF! On ZDF may be attached either to the film or to tape. Such ambiguities induce different semantic readings. Analogously, lexical entries are ambiguous if they belong to more than one category or to different semantic types as in the example “What films are on this evening?”. Here film means “audio/video-event of certain genres”. As Discourse Representation Structures (DRS)$^4$, the different readings can be formalised as:

\[
\begin{align*}
&\text{takeplace}(a) \quad \text{partofday}(a, \text{ev}) \\
&\text{date}(a, \text{today}) \quad \text{agent}(a, f) \\
&\text{avevent}(f) \quad \text{genre}(f, \text{thriller})
\end{align*}
\]  

(1)

\[
\begin{align*}
&\text{takeplace}(a) \quad \text{partofday}(a, \text{ev}) \\
&\text{date}(a, \text{today}) \quad \text{agent}(a, f) \\
&\text{avevent}(f) \quad \text{genre}(f, \text{feature})
\end{align*}
\]  

(2)

The used concepts are defined in the ontology for EMBASSI and considered primitive here.

3.2.1 Intensional Level

The semantics of an utterance may be unique, ambiguous, or impossible to construct. We have to distinguish between the meaning to be assigned to an utterance in terms of a terminology defined for an application (intension) and the enumeration (i.e. extension) of all objects satisfying the intension.

Consider the phrases I like films and I like the film “Goldfinger”: In the first case, the concept of film is contrasted to book, music, or anything else one could like. However, no particular film is mentioned, only the genre is referred to. So, What films do you like (most)? asks for the elements in the set film with the property I like them. The second example, on the other hand, refers to a particular film, namely “Goldfinger”, and not to films in general.

We represent intensions with Description Logics (DL)$^5$. The intension of the film on ZDF is:

\[
\text{avevent} \cap \forall \text{genre.feature} \cap \forall \text{agency.avchannel}
\]

The first restriction $\forall \text{genre.feature}$ is introduced by the lexical semantics of film: A film is an event

---

$^2$This will be discussed in Sect. 5.

$^3$This step is performed by a chart parser with a chunk grammar, working primarily with a head-driven bottom-up strategy.

$^4$See [Kamp and Reyle 1993] for an introduction to Discourse Representation Theory (DRT)

$^5$In DL, one can define primitive concepts as unary predicates. From them complex ones may be derived using concept forming operators, e.g. $\cap$ and $\forall$. Given concepts $A$ and $B$, $C$ may be defined as $C := A \cap B$, i.e. $C$ is the concept containing all elements in $A$ as well as in $B$. Using the binary predicate (or role) $R$, one can define $C := A \cap \forall R.B$: An element is in $C$ if and only if it is in $A$ and $\forall R.B$. This may be interpreted as a type restriction in the following way: An element $e$ is in $\forall R.B$ if and for each $b$ with $R(e, b)$ (i.e. any element related to $e$ via $R$) $b \in B$ holds. Consequently, elements in $C$ have the property to be $A$-elements with the restriction that any element related to them via role $R$ must be $B$-elements. So, $C$ is a “specialised” subset of $A$. For an introduction to DL see e.g. [Donini et al. 1996]
with a genre feature. The second restriction is imposed by the PP on ZDF whose semantics is ∀agency.achievement.

As shown in (1) and (2), intensions may be ambiguous: There, one has to distinguish between two different concepts for the film: aevent ∩ ∀genre.feature and aevent ∩ ∀genre. thriller. If such ambiguities are detected, clarification dialogues are necessary to obtain a unique reading.

3.2.2 Extensional Level

When a unique interpretation on the terminological level has been found, the elements in the set represented by this description must be determined. These are the objects the utterance is potentially about. The set is enumerable by the objects of the application domain to which the discourse refers and in (2) refer. As the semantics of the DRS is

∃a, f  
takeplace(a) ∩ agent(a, f)  ∩  
avevent(f) ∩ genre(f, feature)  ∩  
date(a, today) ∩ partofday(a, ev)

when determining the extension of the original utterance we are trying to satisfy the concept

takeplace  ∩  ∀agent.avevent  ∩  ∀genre.feature  
 ∩  ∀date.today  ∩  ∀partofday.ev

which is subsumed by the intensity of the utterance:

takeplace  ∩  ∀agent.avevent  ∩  ∀genre.feature  
 ∩  ∀date.today  ∩  ∀partofday.ev

On the application pragmatic level, it must be validated whether the extension (i.e. the interpretation of the DRS) constructed for the utterance is consistent with what is known about the state of the application scenario. There are three possible outcomes of this analysis: for each case, the dialogue starting with the utterance above has to be continued in a different way: If the set is empty, a coherent response is “Sorry; no films are on tonight.”, whereas in the case of a singleton set an acknowledgement may be prompted: “Goldfinger is on from 20:15 to 22:30 on ZDF.” If the set contains several elements, the user should be allowed to choose by initiating a clarification: “There are several films on this evening: Goldfinger from 20:15 to 22:30 on ZDF or Dangerous Minds from 23:45 to 1:05 on RTL.”

3.3 Discourse Pragmatic Level

Up to now, our discussion has focussed on the propositional content of an utterance. But, as utterances are actions in a given discourse situation, the analysis of speech acts is an equally important task.

Our approach to speech act recognition is based on two main ideas: First, speech acts are actions in a discourse. Therefore, basically, the recognition of (possible) speech acts depends on whether the preconditions for a speech act(ion) are satisfied in the current discourse situation. Speech acts are defined independently from applications, and their number is kept small in order to enable robust analysis. Second, preconditions are formalised by using the notion of expectation and obligation introduced into a discourse by previous utterances, by describing how the propositional content of an utterance contributes to the recognition of a speech act (e.g. by discourse markers and cue phrases). Finally, assumptions about the knowledge of the dialogue participants are taken into account. Again, there are three possible outcomes of the analysis: If no speech act can be identified, it is unclear in what way the utterance is intended to contribute to the current discourse. A possible reaction could be: “Why do you say that ...?”. If several speech acts are possible, one could ask: “Do you want to say that ... or do you want to know whether ...?”. Third, a unique speech act is recognised.

3.4 Satisfaction of an Utterance

If a unique speech act has been recognised, a unique dialogue goal consisting of the speech act and the content of the utterance can be established. To continue the dialogue coherently, it is necessary to know whether the goal can be achieved. For this purpose, we introduce the notion of plans as a means to abstract from application-specific algorithmic solutions. The dialogue goal can be achieved uniquely if a single plan satisfies it. The goal is unsatisfiable, if no plan could be developed, or ambiguously satisfiable, if several plans have been found. In this case, they should all be presented to the user.

3.5 Result

In the case of a unique satisfaction of the dialogue goal, a problem solver executes the corresponding action and reports the status of the execution back to the dialogue manager (DM). If the action has been executed successfully, the result is communicated. In the case of an error, the user should be informed about the possible cause. Otherwise, the result consists of a number of different alternatives which are presented to the user.

Sometimes the information available is insufficient for executing the required task. If so, the problem solver may ask for additional information.
Such clarifications delay the completion of the initial achieve\textsuperscript{6} call. For example, a clarification could be: “Should the video in the living room be used or the one in the kitchen?”

4 Language Generation Issues

In this section, we discuss Natural Language Generation (NLG) issues relevant to our dialogue system on the basis of the example dialogue presented in Sect. 2. On a sufficiently abstract level, the NLG module faces a situation similar to the one described in the preceding sections: First, if the input provided cannot be uttered at all, an error message has to be generated to inform the DM that the current dialogue goal could not be achieved; the user has to be informed as well and an explanation dialogue has to be initiated. Second, if the input can be uttered in exactly one way, generation is straightforward; but with increasing generative capacity of the NLG module, this situation becomes less frequent. And, third, if several utterances are possible, the NLG module has to constrain the generation process appropriately using knowledge sources other than the logical proposition delivered by the DM, such as the discourse history and the user model.

According to [Stent 1999], planning and realisation of utterances can be decomposed into the planning of intention, content and form. In our dialogue system, the dialogue manager serves as the central component and determines both the intention and the content of the utterance to be generated, leaving only the form to the NLG module. Hence, a sentence planner is required that takes as input a logical proposition plus the system intention and produces as output a sentence plan which figures as input for the realiser. The disjunction found on a high level of the sentence LF is passed on to the lexical level to produce an elliptic realisation.

Our NLG approach is based on the theory of conversational acts, originally developed by Traum (cf. [Traum 1994]) for descriptive purposes only. For NLG purposes, the theory has to be slightly revised and extended (cf. [Stent 1999]). The three levels of speech acts can be nicely mapped to NLG depth levels: Core speech acts can be expressed by canned text, primary speech acts by templates\textsuperscript{8}, and secondary speech acts require deep NLG.

Our idea of efficient NLG is to use the generative capacity of deep NLG only when it is desirable (to produce pragmatically adequate utterances with varying complexity, choice of register, lexical choice etc.) and to rely on templates as much as possible\textsuperscript{9}. This includes incrementally enhancing the template database by storing sentences generated by the deep NLG branch together with the conditions that lead to their application\textsuperscript{10}. We intend to search the template database first\textsuperscript{11} and use deep NLG only if no matching\textsuperscript{12} template can be found.

5 Analysis of the Example

U\textsubscript{1}: This utterance is ambiguous with respect to its intensional meaning. The reason for this is the lexical semantics of the word Film: The concept a\textsubscript{1}event is assigned to it. In the domain model, the application specific knowledge of the dialogue system is represented. a\textsubscript{1}event has got two subconcepts: film a\textsubscript{1}event := a\textsubscript{1}event \cap \text{genre}.feature and also thriller a\textsubscript{1}event := a\textsubscript{1}event \cap \text{genre}.thriller. These subconcepts induce the ambiguous readings of U\textsubscript{1} that are shown in (1) and (2). This means that the satisfaction state for the intensional view is ambigu-

\textsuperscript{6}The various modules in our implementation communicate with KQML ([Finin 1997])

\textsuperscript{7}which is, of course, part of the NLG module as well

\textsuperscript{8}In our approach, templates and canned text are being implemented in a single component, as templates can be viewed as generalised canned text augmented with phrasal variables.

\textsuperscript{9}cf. [Busse 1998] for further elaboration on this approach.

\textsuperscript{10}not just the logical proposition and the dialogue act submitted by the DM, but also pragmatic constraints derived from the knowledge sources mentioned in Sect. 6.

\textsuperscript{11}Efficient storing and searching techniques are the key to this approach.

\textsuperscript{12}in the extended sense mentioned above: The state of the knowledge sources that were accessed to generate the sentence is also stored in the database.
ous. The system disambiguates by initiating a clarification subdialogue in $S_2$.

$S_2$: In $S_2$, the system addresses the ambiguity of $U_1$ by asking the user a question that should clarify the meaning of $U_1$. Hereby, the system expects the user to answer the query and an obligation for the user is introduced in the dialogue situation.

This clarification question is realised with an elliptic sentence that leaves out the predicate and passes the disjunction operator on to the lexical level, as mentioned in Sect. 4.

$U_3$: The DRS for $U_3$ is:

$\begin{align*}
\text{want}(w) & \text{ agent}(w, u) \text{ user}(u) \\
\text{content}(w, a) & \text{ watch}(a) \text{ patient}(a, f) \\
\text{avevent}(f) & \text{ genre}(f, \text{comedy})
\end{align*}$

(3)

As there is an obligation for the user pending, the system first tries to understand $U_3$ as if it were meeting its expectation. As comedy \subseteq feature, the system may infer (using the domain model) that $\text{avevent}(f) \cap \text{genre}(f, \text{comedy}) \subseteq \text{avevent}(f) \cap \text{genre}(f, \text{feature})$. This means that the user chooses film/avevent to address the obligation. Consequently, the system specialises (2) as follows:

$\begin{align*}
\text{takeplace}(a) & \text{ agent}(a, f) \\
\text{avevent}(f) & \text{ genre}(f, \text{comedy}) \\
\text{date}(a, \text{today}) & \text{ partofday}(a, \text{ev})
\end{align*}$

(4)

(1) remains as an unaddressed hypothesis for $U_1$. Still it has to be decided whether $U_3$ is coherent with $U_1$. To achieve this, the system applies the following rule about intentions (expressed by I want): If something is intended to be done, then its preconditions have to be satisfied. In this sense, $U_1$ has to be a precondition for $U_3$. As this is the case, $U_3$ is seen as a specialisation of $U_1$; the user has addressed the pending obligation which is no longer active now. Instead, the system is obliged to answer (4) which is the unique interpretation of $U_1$. To do this, the programme database (epg\(^{13}\)) is queried:

$\begin{align*}
\text{ask-all} & \quad \text{:sender dialogue :receiver epg} \\
\text{content} & \quad \text{(4)}
\end{align*}$

The answer is:

$\begin{align*}
\text{tell} & \quad \text{:sender epg :receiver dialogue} \\
\text{content} & \quad \text{(4)}
\end{align*}$

$S_4$: As the answer from the database is unique, it is communicated directly to the user. The system’s obligation addressed now and no longer active. The dialogue goal of $U_1$ is processed completely.

With a reference to the discourse history, the NLG module is caused to use a verb different from the usual “kommen” for lexical variation purposes. Besides, the starting time is topicalised, as this information is the one to be immediately communicated to the user.

$U_5$: The function word and in $U_5$ is a binary operator whose first operand is missing. Krimis alone is ambiguous on the intensional view as different operations may be carried out on avevents.

To obtain a unique reading of $U_5$, the system searches the user’s previous utterances whether they specialise or generalise $U_5$ in order to get the missing operand. $U_1$ is found as the unique generalisation of $U_5$ in the dialogue so far. $U_1$ can be specialised to the reading shown in (1). Adopting $U_1$’s speech act (query) is not contradictory with information about $U_5$. This results in a unique interpretation of $U_5$ on all levels. Now, the system tries to satisfy $U_5$. Again, the epg is consulted; the answer is:

$\begin{align*}
\text{tell} & \quad \text{:sender epg :receiver dialogue} \\
\text{content} & \quad \text{(4)}
\end{align*}$

$S_6$: From the DM point of view, this is completely analogous to $S_4$. The NLG module, however, has to generate an elliptical sentence with the default word order (SVO), because the most important information is contained in the subject this time.

$U_7$: The pronoun der (that) is resolved to “Tatort” as the closest antecedent. So, the DRS for $U_7$ is:

$\begin{align*}
\text{takeplace}(a) & \text{ agent}(a, f) \\
\text{avevent}(f) & \text{ genre}(f, \text{thriller}) \\
\text{title}(f, \text{“Tatort”}) & \text{ date}(a, 2000/05/22) \\
\text{partofday}(a, \text{ev}) & \text{ starttime}(a, 20:15)
\end{align*}$

(5)

This is a unique interpretation and $U_7$’s speech act

\(^{13}\)for Electronic Programme Guide
is identified as a question. So, again the system tries to satisfy the user’s utterance by consulting the epg which responds with

\[
\text{tell :sender epg :receiver dialogue :content}
\]

\[
\begin{align*}
\text{takeplace}(a) &\text{ agent(a, f) aevent(f)} \\
\text{genre}(f, \text{thriller}) &\text{ title(f, "Tatort"}) \\
\text{date}(a, 2000/05/22) &\text{ partofday(a, ev)} \\
\text{starttime}(a, 20:15) &\text{ endtime(a, 22:15)}
\end{align*}
\]

\(S_4\): Nothing new from the DM perspective, but the NLG module has some choice points to master: Either, a very short and elliptic sentence is uttered (this would probably be the first choice), or an anaphoric reference to the referent introduced in a preceding user utterance has to be generated, which requires access to the dialogue history.

\(U_9\): As in the previous user utterance, anaphora resolution is necessary to compute a unique semantic representation for the utterance to be interpreted. This time, “Spion in Spitzenhöschen” is the nearest antecedent for the anaphoric noun phrase den Spielfilm. Consequently, the semantics for \(U_9\) is:

\[
\text{want(w) agent(w, u) user(u)} \\
\text{content(w, a) record(a) patient(a, f)} \\
\text{aevent(f) title(f, "Spion in Spitzenhöschen")} \\
\text{date(a, 2000/05/22) partofday(a, ev) starttime(a, 21:45)}
\]

Up to the discourse pragmatic view, \(U_9\) is unique. Again (as in \(U_3\)), an intention is expressed, but this time, there are no pending obligations, so the system may directly consult the appropriate problem solver whether the user’s intention is satisfiable (as the intention introduces obligation for \(S\), and \(S\) is cooperative, it tries to satisfy the user’s request):

\[
\text{ask-all :sender dialogue :receiver recorder :content}
\]

\[
\begin{align*}
\text{record(a) patient(a, f)} \\
\text{aevent(f) title(f, "Spion in Spitzenhöschen")} \\
\text{date(a, 2000/05/22) partofday(a, ev)} \\
\text{starttime(a, 21:45)}
\end{align*}
\]

The problem solver answers:

\[
\text{tell :sender recorder :receiver dialogue :content}
\]

\[
\begin{align*}
\text{status(a, ok) record(a) patient(a, c)} \\
\text{channel(c) channelid(c, BR3)} \\
\text{date(a, 2000/05/22) starttime(a, 21:45) endtime(a, 23:05)}
\end{align*}
\]

\(S_{10}\): As the result from the problem solver is unique, the system utters \(S_{10}\) to communicate the successful execution of the user request \(U_9\). The answer is preceded by a summary “OK” introduced by the NLG module to give the user a very fast feedback on the completion of his request.

6 Linguistic Knowledge Sources

Quite a few knowledge sources have to be incorporated into a speech dialogue system which can be useful, among other modules, for the parser, the DM, and the NLG module in particular: Considerable work has been done on grammar reversibility (cf. [Strzalkowski 1994]); lexicon and morphology can relatively easily be used or at least adapted for parsing and generation\(^{14}\); the discourse context and the dialogue history are knowledge sources we try to re-use as well.

The idea of re-use is pursued by splitting up the domain model horizontally and vertically, into linguistic and application domain and into upper and lower model (on the application side), respectively.

7 Related Work

Our approach to modelling rational dialogues and to dialogue planning has a lot in common with the work of Sadek et al. [Sadek 1999]. But whereas Sadek uses an explicit (meta level) representation of the rationality principles as modal logic axioms and a modal logic theorem prover for the planning of dialogue steps, in our case these principles are implicit in the object level interpretation, i.e. the use of the partial logic FIL\(^{15}\). The rationality principles are present, but in a “compiled” form by means of so called dialogue game rules. Our understanding of dialogue games is due to [Cooper 1998]. The issue of how utterances contribute to a dialogue on various levels of locality, has been studied in detail by Traum and Poesio [Traum 1994, Poesio and Traum 1997]. Stent describes an NLG system architecture based on Traum’s and Allen’s work. The approach to use a combination of deep and shallow NLG is found elsewhere as well (cf. [Busemann 1998]), but the idea

---

\(^{14}\)Efficiency is, of course, a different question

\(^{15}\)First Order Inference Logic. [Abdallah 1995]
to use the theory of conversation acts for NLG planning purposes is presented in [Stent 1999] for the first time.

8 Conclusion and Further Work

We have sketched a generic dialogue model which overcomes many of the limitations of current finite-state data-base inquiry systems. Our flexible dialogue manager is adaptable to different applications because it separates between language specific, but application independent and domain specific components. The problem solver capsulates application-dependent actions and the algorithms for their execution. The knowledge sources used in our system are available to each module as well as partially reusable for new domains. We have prototypes running for each described module and also demonstrated the adaptability of the DM to various domains by alternating knowledge sources.

As a next step we will complete implementation and evaluation of our system. We intend to develop methods to semi-automatically extract the information needed to build domain-specific knowledge sources from application specifications.

9 Acknowledgements

We thank our colleagues Kerstin B"ucher and Maria Aretoulaki for their helpful comments on draft versions of this paper.

References


