

Speech Dialogue Systems — A Pragmatics-Guided Approach to Rational Interaction

Günther Görz

Computer Science Institute
University of Erlangen-Nuremberg, Germany

Abstract—Modelling rational dialogues on the basis of Finite-State Automata has shown to be of limited use when natural speech behaviour and dynamic adaptation to new domains are an issue. To enable such a flexible, robust and generic dialogue management, we rely on an approach characterized by a logically precise modelling of the linguistic and application aspects of the domain in question and a dialogue manager deriving pragmatics-driven discourse strategies from reasoning on evolving belief structures, both of the user and the system. For the linguistic part, we claim that a “pragmatics-first” view on rational interaction provides an appropriate framework for flexible and scalable dialogue modelling. In particular, the plan-based approach offers the means to conduct task- or goal-oriented dialogues which aim at accomplishing concrete tasks. It enables cooperative response behaviour and the ability for negotiation. For the reasoning part, i.e. knowledge representation and inference for the interpretation of dialogue as well as for planning to satisfy user goals in the application domain, we argue for a computational logic framework. By means of a clear functional separation between the language model, the dialogue model, and the domain model, the feedback cycle between the dialogue manager and the applications is of vital importance, as is the close interaction on the linguistic side occurring between the speech parser, the dialogue manager and the text generation module.

I. GENERAL ASSUMPTIONS

Our general goal is to build dialogue systems for rational interaction. What we want to achieve is the satisfaction of user goals in a given (ideally open) domain by conducting spoken dialogues where it should be possible in principle to augment them by other forms of multi-modal interaction like gestures or the selection of items from a menu on a screen. Interactions are called “rational” because we want to apply rationality principles (at the knowledge representation level) to optimally select appropriate communicative actions. We assume that the satisfaction of user goals within the thematic framework of a particular application domain is to be achieved with the help of a dialogue system proper in cooperation with a technical application which we also call the “domain problem solver”. Such a technical application can be an information or reservation system, a system for controlling certain devices, etc.

In such settings, [1] characterize “practical systems” as task- or goal-oriented dialogue systems, where the dialogue is focussed on accomplishing a concrete task. They claim that the conversational competence required for practical dialogues is significantly simpler to achieve than general human conversational competence. We are convinced that this distinction is useful because it provides a realistic starting point and because

the latter – general human conversational competence – is hard to define. Hence, we begin with a certain well-defined conversational competence and with increasingly complex requirements from the application we can try to augment it, i.e., we attempt to proceed in an incremental fashion. But probably this will not be easy; we cannot expect that this can be done in the way of a linear progress: There may be fractions where completely different requirements come in which cannot be integrated seamlessly.

Another general underlying assumption is that for the interpretation of dialogue we insist on a clear commitment to a (computational) logic framework. Of course, humans act incoherently and even inconsistently, and common sense reasoning can only to a certain extent be understood in terms of logic, but we are convinced that a coherent and consistent rational reconstruction is the best we can do about it. Such a constructive perspective has the advantage of enabling us to begin with a well understood framework for knowledge representation and reasoning upon which we can attempt to build rule systems for still idealized, but more realistic patterns of argumentation in specific domains. We believe that there is a potential to succeed in a variety of prevalingly instrumentalized contexts as it is the case with technical applications – that will be discussed in more detail below – or, to take up another example, in forensic argumentation.

II. ON DIALOGUE MODELLING

The question of scalability of natural language and speech dialogue systems has been an issue for quite a long time. And it becomes more and more pressing with the rapidly increasing processing power and memory size of modern hardware and the progress in software engineering techniques. Before we address the dimensions of scalability and the problems connected with it, let us have a look at history to sharpen our awareness.

A. In the Beginning: The Structural Approach

For several decades, natural language and speech systems for information dialogues have been developed on the basis of an approach which led from experimental systems to a commercially available technology. Although successful in many domains, we will argue that those systems have severe limitations which are inherent to the underlying so-called structural approach.

The goal addressed by this first system generation is to accomplish an information task. By conducting more or less strictly guided dialogues they aim at supplying a user with a specific information in a given domain, e.g. train connections, traffic jams, cinema programmes, stock exchange data, weather information, etc. in as few dialogue turns as possible. The technical application they are cooperating with is usually a static database system; in some cases it has been extended with an application server that might be taking orders or reservations. The basic technique these systems use is the extraction of parameter values for a given schema from user utterances (“slot filling”) under the closed world assumption. In general, they have a very limited and domain-specific inferencing capability, if at all. As for dialogue modelling, they rely on a state-based approach, which is also called *structural*. Driven by speech recognition technology, dialogues are modelled by means of stochastic finite-state machines. To do so, the constructor of a structural dialogue model is forced to anticipate future admissible dialogue states. If large annotated corpora are available, stochastic training techniques can help a lot, but the general problem of anticipation remains. In this framework, the general view of dialogue in speech communication is understood as controlling and restricting interaction – and this is fundamentally different from basing human-computer interaction on human conversation (cf. [1]). This “controlling and restricting” view is to a large extent due to robustness requirements that all speech dialogue systems – not only the structural ones – are faced with: recognition errors, speaker adaptation, and “out of vocabulary/domain” problems.

So, what are the inherent limitations of the structural approach? In spite of various improvements to structural systems like the introduction of anchoring into the dialogue history, using dialogue act predictions and defaults, and employing sub-dialogue patterns, they have no means to deal with semantic and pragmatic problems in a flexible way, which is the most difficult part of the anticipation problem with guided dialogues.

This assertion is corroborated by the results of an evaluation of more than 1100 recorded train information dialogues with the EVAR system, a state-of-the-art structural system developed at our university [2], [3]: 68.5% of the dialogues were completed successfully. Of the remaining 31.5% incomplete dialogues no less than 83.1% failed because there was no (proper) integration into the context; 14.7% were cancelled by the user, and 2.2% failed because of a system crash.

To resolve semantic and pragmatic problems which result from a high degree of dependency on the dialogue and application context, a system must be able to draw inferences based on a dialogue model and an application model. This requirement goes far beyond the expressive means provided by a structural model, which usually integrates dialogue and application knowledge in a finite-state machine – for theoretical reasons.

At this place, another aspect of the close integration of dialogue and application features must be addressed, although it is not a fundamental theoretical limitation of the structural approach: Because these systems usually are tailored to a

specific task type, mostly static database querying, they lack of a sufficient modularization. The need to factor out proper dialogue features is hardly addressed. Therefore, attempts to build in extensions or to port to new domains within the same task type are faced with a considerable technical effort, which constitutes a bottleneck to scalability. Porting to a new domain with different task types will often result in a complete reimplementing of the structural dialogue model.

Let us come back to the previous argument: The need for reasoning becomes even more apparent if we have to deal with meta-discourse, a phenomenon which is frequent in corpora, and with dynamic domains.

To give an example, we may consider the scenarios the EMBASSI system has to deal with. EMBASSI is a joint national project sponsored by the German Ministry for Research and Technology with 19 partners from industry, research institutes and universities¹. Its goal is to develop a system which is able to control devices by means of multi-modal dialogues, e.g. an audio-video theatre in the home or a radio and navigation device in cars. Particular emphasis is put on flexible assistance concepts in the realization of the EMBASSI system. For the first application area, there is the explicit requirement that the application system is reconfigurable, i.e., that new devices can be integrated in a plug-and-play fashion on the fly.

The technical answer of the EMBASSI consortium to the quest for such a high level of flexibility and scalability was to design a highly modular agent-based system architecture where the modules communicate in a uniform agent communication language (KQML/ACL). In the following we will address the question of the theoretical foundations and basal decisions to make it happen.

B. Reconsidering Basal Design Decisions

The discussion of the state-based approach to natural language and speech dialogue systems indicated a need to reconsider basal design decisions. This means, we have to provide arguments and reasons which help in the search for solutions, in other words, we are looking for theoretical justifications for such decisions. For the following, “theory” will address the spheres of language and reasoning. As for system architecture, this seems still to be rather an art and engineering practice, and we will be happy if we were able to derive at least some constraints on architectural designs.

What we will do in the following sections, is to set up a general perspective on dialogue systems which can be characterized as a “pragmatics-first” view on rational interaction².

For dialogue modelling, we will follow the **plan-based** approach which has its roots in natural language processing and Artificial Intelligence. It provides the means to conduct task- or goal-oriented dialogues which are focussed on accomplishing concrete tasks as mentioned in the introduction. We claim that only a general planning approach enables cooperative response behaviour (pragmatic adequateness, overanswering) and the ability for negotiation.

¹Grant No. 01 IL 904 F 8

²From a theoretical perspective see the pioneering work by Cohen and others, cf. the contributions in the volume [4]. As an example for a system strictly based thereon cf. Sadek’s ARTIMIS, [5]–[7]

For the reasoning part, i.e. knowledge representation and inference for the interpretation of dialogue as well as for planning to satisfy user goals in the application domain, we will refer to a computational logic framework, in particular description logics.

There is no doubt that a minimal prerequisite for scalable systems is that they have a modular structure. We will argue that a clear functional separation between the language model, the dialogue model and the domain model provides a sufficient condition to address scalability.

Some remarks on our current work on dialogue management within the EMBASSI project will illustrate the practical implications of this theoretical framework³.

C. Dimensions of Scalability

Often the requirement for scalability of dialogue systems is seen primarily as an engineering problem. At a closer look, it becomes apparent that there is more to it: Scalability must be considered in the context of underlying assumptions and specifications, which of course concerns various technical aspects, but also includes the question how we understand the interaction between user and system.

At least, the following issues have to be addressed, but, of course, the list is open for extensions:

- vocabulary size,
- linguistic coverage and utterance types, in particular varieties of reference, e.g. anaphor and ellipsis resolution,
- multi-linguality,
- mixed-initiative dialogues,
- multi-modality,
- single or multiple simultaneous goals,
- multi-party dialogue,
- size and fundamental properties of the application domain(s)
- extensibility of applications by new subdomains, e.g. train information plus ticket sale plus hotel information and reservation,
- switching to new (sub-)domains,
- reconfigurability of the application,
- system portability to new domains.

We are far from being able to provide operational answers to these questions, not even to a part of it. The best we can do at the moment, is to present some theoretical considerations which indicate the directions in which to search for answers. Of course, many authors have addressed problems mentioned in this list. Interestingly, in most cases we know of, if the authors share our assumption that scalability is not only a technical problem, there is no fundamental disagreement w.r.t. the theoretical premises.

III. THEORETICAL CONSIDERATIONS

The term “theory”, in particular linguistic theory and logic, will be understood in a very general and explicitly non-formalistic sense, which means that foundational issues are part of our theoretical reflection. For the latter, the hardest

part lies in the beginnings: We have to become clear about the basal assumptions we are starting with and we have to make them explicit in order to avoid argumentative cycles in the construction of our terminology. Considerations of the actual, developed level of theory should include an understanding of its genesis – how did we get to where we are? –, and in particular where opportunities for alternative development paths had been.

A. Linguistic Theory and Linguistic Processing

1) *Linguistic Theory and the Pragmatic Turn*: The traditional division of the theory of language into the investigation of its structure, meaning, and use is rather uncontroversial. This way of modularizing linguistic knowledge has been taken up by most NLP system constructors as a basis for system modularization. But, of course, for system building this is only one part – what can linguistics say about processing? Chomsky’s claim that linguistics considers only competence, not performance [10], is of little help if we aim at functional, practical systems. If we have to deal with real users’ input the integration of performance issues is inevitable. The big question whether there is a theory of “human language processing” has been taken up by cognitive scientists, and although their research led to a broad variety of interesting results, it is still debatable whether there is yet a general framework in cognitive science which really deserves this name. We will come back to this point in the section on system architecture.

So many, if not most, systems today still map in their architecture the construction of linguistic theory in the formalistic tradition as outlined by Chomsky and many others working within the same paradigm: Signals are turned into symbol strings which first are segmented and scanned lexically and then analyzed into phrase structures, in some cases also dependency structures. Later on, these grammatical structures are transformed into some kind of logical form which is supposed to express their meaning. For conversational systems, the pragmatic level is in most cases represented by the use of speech acts and the representation of intentions as the last and subordinate processing step in analysis. Independent of the adequateness question for sequential processing, this class of systems can be understood as realizations of a theoretical conception. That they have deficits on the performance side – although in many practical systems a lot of strategies have been implemented to cope with performance issues – is in principle a clear consequence of that conception.

Of course – for whatever theory – what system builders can achieve at best is a clear operationalization of theoretical constructs and their implementation under the limitations of problem decidability and complexity. Hence, certain shortcomings of NLP systems on the performance as well as on the pragmatic level are in a way a consequence of a decision on the meta-theoretic level, i.e. the choice of a certain linguistic theory type. So, what is the alternative? In our conviction, philosophy of language has given an answer already a while ago with what is called the “pragmatic turn”. So, when we push the introduction of a “pragmatics-first” perspective in NLP, we consider *communication as action* – which in fact means

³Further information about our approach can be found in [8], [9].

a radical departure from the formalistic mainstream: Taking up the “pragmatic turn” in NLP means to turn Chomsky upside down. Methodologically, pragmatics is put at the beginning; on that basis semantic and syntactic categories are understood as pragmatically founded distinctions. This is different from the traditional view where meanings come up as abstract objects which are linked to purely linguistic objects in a functional way; in this way pragmatics is just the investigation of dependencies between meaning functions and functions of the use of linguistic expressions. In other words, pragmatics – the use of language, requirements of communicative functionality – determines semantics and syntax.

To be historically precise, this methodological stance of looking at language primarily as a means of communication even predates Chomsky a lot, because it had already been introduced by the Prague school of linguistics in the 1920s under the term of “functionalism”. Their starting point for analysis is the speaker’s intention as expressed by a linguistic utterance, i.e. the analysis begins with the function of an utterance in order to describe its form. The “functional sentence perspective” then establishes the thematic/rhematic structure of utterance sequences or texts as the main structural principle. This approach is complemented by the theory of speech acts according to Austin and Searle⁴ who share the communicative view on semantics and syntax.

Whereas at the first glance it may seem a bit strange if we do not understand the choice of a particular linguistic theory along with an appropriate representation formalism as the fundamental theoretical question for NLP in the first place, we hope it became clear that a comprehensive theoretical attitude is the issue. It sums up to rather understanding language in its social context – language as action – than departing from a particular grammatical framework.

But, of course, one has to ask – in the same way as for programming languages – how far a certain formalism supports modularization, on the level of grammar itself as well as between syntax proper, semantics, and pragmatics. And furthermore, linguistic processing is constrained by various technical factors among which, in the case of speech, is the quality of recognition, i.e. of the transformation of signals into a symbolic representation. Up to now, there is no alternative to the success of stochastic methods for speech recognition. With considerable vocabulary sizes, the best that speech recognition technology can offer is not a “best string”, but a lattice of competing scored word hypotheses. For parsing, this means that in most cases we will not be able to find a single spanning syntactic description ranging over whole utterances or dialogue turns [12], but that we must expect sets of syntactic fragments which can be combined into bigger units by employing constraints from linguistic semantics as well as the semantics of the application domain and foremost from discourse pragmatics.

2) *Linguistic Processing in the Erlangen Dialogue System:* It is worthwhile to point out that the overall processing control, of which linguistic analysis – and generation as well – is a

⁴For its influence in the domain of dialogue systems cf. Traum’s and Allen’s theory of conversational acts [11].

part, is the duty of the dialogue manager⁵. For the analysis of utterances, in general we have to take complex speech acts into account, where the interpretation of the utterance’s propositional content is determined by its (local) linguistic-pragmatic context in the first place. This, in turn, is to a large extent influenced by (global) discourse-pragmatic features which provide constraints based on the dialogue history and the actual place of the utterance in the dialogue, as, e.g., being the expected answer to a question. Furthermore, the application provides further constraints by restricting the meaning of words and phrases to their particular use within a given thematic framework. Therefore, we have to distinguish several – interleaved – levels in the analysis of user utterances:

- Linguistic analysis on the utterance-local level, which in turn consists of several levels of syntactic and semantic construction;
- Semantic evaluation, i.e. evaluation of semantic operators, reference resolution, and additional transformations of the logical form, augmented by specific computations;
- Application-domain specific specialization of the evaluated semantic representation, where the general lexical concepts are replaced by domain concept structures according to the formal ontology of the application (cf. IV-B);
- Discourse-pragmatic analysis – a proper function of the dialogue manager (cf. IV-B).

In this section, we will briefly describe the techniques we chose for linguistic processing, which comprises the first and second level. For parsing, we build upon “chunks” which provide a first grammatical segmentation of utterances. Following [13], a chunk consists of a syntactic head, which is a content word, surrounded by a constellation of function words in fixed patterns. In the case of speech, we also take prosodic boundaries into consideration. Each chunk has an internal structure according to the X-Bar schema: head + complement + adjunct + specifier. For chunk parsing, we use a chart parser which operates in three phases. Lexical scanning is performed by employing a fairly traditional lexicon together with a morphological analyzer. Parsing phase one is the recognition of chunks in a lattice of word hypotheses on the basis of a unification chunk grammar. In the second phase, the syntactic functions of chunks are analyzed and used for the combination of chunks into bigger units, finally resulting in the construction of a dependency tree. To succeed, semantic and pragmatic constraints must be satisfied, the applicability of which is determined by valency and case frames of the lexical units. So, constraints on complements and modifiers of German words are expressed in terms of case frames which state the valencies of a word and their possible semantic filler types.

Of course, we have to cope with lexical and grammatical ambiguity and the ambiguity introduced by recognition uncertainties. We implemented an ambiguity selection mechanism that should help to increase robustness. [14], [15]

Grammatical structure analysis is incrementally tied with the third parsing phase, the semantic interpretation of chunks.

⁵cf. IV-B; the system architecture is depicted in fig. 1

This part in turn consists of three phases: First, we identify word and intra-chunk semantic information, which are domain-independent. The second step consists of the grammatical determination of inter-chunk relations. The third part is to perform semantic construction by means of construction operations associated with the chunk grammar rules into Discourse Representation Structures (DRSs). For the latter we use λ -DRT, a fully compositional derivative of Kamp’s Discourse Representation Theory (DRT) [16]. In correspondence with the syntactic amalgamation of chunks their DRSs are incrementally combined by substitution, the evaluation of DRS operators, and discourse referent resolution which allows to build up DRSs which transcend sentence limits. For anaphor resolution, we developed a computational framework based on linguistic and pragmatic heuristics in [17]⁶. DRSs may contain logical operators, e.g. disjunction and conditional expressions and so called “duplex conditions” representing natural-language quantifiers. Evaluating such DRSs means to apply certain transformations to them. Disjunction will lead to two alternative DRSs. For quantifiers, the scope ambiguity problem can be resolved by applying the “Cooper storage” algorithm to several DRSs representing different readings.

The ultimate goal in this part of analysis is to transform the domain-independent semantic representation into a description of the discourse situation which is specialized to the respective application domain of our dialogue system. Therefore, the processes mentioned above are incrementally combined with the execution of further rules in order to achieve the application of domain-specific constraints as soon as possible. To do so, first of all we need to access the domain-specific concepts which are available through a link between the general lexical semantic information and the specific semantics of the application domain in the lexicon. Some more details are given below (IV-B). We then have to instantiate the respective domain concepts with discourse referents of the extensional semantics by mapping chunk structures into relations between concept instances. Finally, some further domain-related rule-based transformations have to be executed as, e.g., calendrical calculations to provide the domain-specific discourse situation representation with absolute time specifications.

3) *On the Theoretical Basis for Rational Dialogues:* Under the assumption that the “language as action” perspective provides a flexible and extensible framework for rational dialogues, whose aim is to satisfy user goals in a given application context, we need means to identify such goals and to represent them formally within an explicit representation of an initial situation. We also need methods to decompose a goal into subgoals to be satisfied by the application system, and to control the satisfaction process. Interactions are called “rational” because we want to apply rationality principles (at the knowledge representation level) to optimally select appropriate communicative actions. In other words, we formulate a complex planning problem which comprises at least two levels: planning on the dialogue level w.r.t. interactions between the dialogue system and the user, and planning on

the level of the application system.

For planning on the dialogue level, we need an explicit representation of dialogue situations which on the one hand include statements representing what the system could extract from the interaction with the user so far and on the other hand assumptions about the user’s knowledge about the actual situation as well as on goals, their subgoals and the actual state of their satisfaction. From the system engineering point of view we are dealing with the epistemic level and there is no need for stronger mentalistic claims as far as the user is concerned. The planning process consists in the application of dialogue operations which have preconditions defining their applicability and assertions about their effect, i.e., how the dialogue situation develops when they are applied.

A general logic-based approach for representing and processing dialogue situations on the epistemic level has been developed by Cohen, Levesque and others⁷. How rationality principles can be integrated in such a framework has been shown by Asher et al. (e.g. in [22]). Grice’s conversational maxims as e.g. cooperativity and sincerity are represented axiomatically in a modal logic formalization.

A comprehensive framework for discourse planning has been established by Grosz and Sidner in their pioneering investigations [23], [24], who in fact proposed three levels for modelling task-oriented discourse structure⁸. The *intentional* level records the beliefs and intentions of the dialogue partners regarding the tasks and subtasks to be performed. The *attentional* level captures the changing focus of attention in a dialogue using a stack of so-called “focus spaces” organized around the dialogue tasks. The *linguistic* level represents “segments”, i.e. contiguous sequences of utterances, which contribute to a particular task.

These theoretical studies have been very influential for a lot of systems although we are not aware of comprehensive implementations yet. To quote only a few examples, cf. Rich’s et al. COLLAGEN system [25], Allen’s et al. TRIPS [1] or Sadek’s et al. ARTIMIS [5]–[7]. In ARTIMIS, Sadek uses a modal logic theorem prover, which introduces a high degree of flexibility and extensibility, but – probably – for the cost of losing completeness. Beyond the recognition of user intentions Rich et al. show how plans can be recognized by inferring intentions from actions. Needless to say that there is still a huge need for research into dialogue strategies as clarification, negotiation, and other subdialogues, and on metadialogue.

Our own work builds up on the insight that planning in dialogues is based on partial knowledge. Each contribution of a dialogue turn is differential w.r.t. the present dialogue situation. Therefore we use a monotonic partial logic [26] – which allows a certain kind of defaults – for reasoning in dialogue situations, including the dialogue context, in order to establish common knowledge and conduct dialogue action(s). Instead of making explicit use of modal logic we instead, according to a suggestion by John McCarthy, refer to a realization of it as interpretation in context. Rationality principles serve as constraints on the planning process.

⁶For a general theoretical introduction with a similar computational solution, which covers also presupposition resolution, cf. vol. II of [18]

⁷cf. [19] and further contributions in the volume [4]; cf. also [20], [21]

⁸cf. [25]

Discourse planning, i.e. the determination of a sequence of dialogue steps, has to take into account that the application subsystem influences the sequence of dialogue steps by reacting on preconditions of operations, and generating effects which change the actual state. So, a description of the dialogue step sequence requires representations of time, the “actual state”, the terminology of the application, and the operations, their preconditions and effects.

B. Reasoning

As Pat Hayes has remarked a while ago, it does not make sense to speak of “non-logical” representations of knowledge. In one way or another, for any knowledge representation schema there is a corresponding logic calculus, even if its authors are not aware of it. Therefore, we argue for a clear commitment to (computational) logic for the dialogue manager, i.e. for the interpretation of dialogue, as well as for the application system it is connected to, i.e. for domain knowledge representation and reasoning. As the application may be any software system in general as a database system, a robot controlling system, etc., we will need to introduce a logical layer between it and the dialogue manager in which the domain model is represented and in which inferences, e.g. in planning, are drawn.

If we speak about logic, we do that in its traditional understanding, which is far more extensive than modern formal logic: In particular, we address the traditional branches of concept formation, proposition, and inference. Modern formal logic primarily deals with the latter aspect, but in our field of interest the other two are of equal importance.

All kinds of knowledge representation can be understood best as rational reconstruction. As we already mentioned in the introduction, it is a commonplace that humans act incoherently and even inconsistently, and that common sense reasoning can only to a certain extent be understood in terms of logic, but we are convinced that a coherent and consistent rational reconstruction is the best we can do about it. Such a constructive perspective has the advantage of enabling us to begin with a well understood framework for knowledge representation and reasoning upon which we can attempt to build rule systems for still idealized, but more realistic patterns of argumentation in specific domains. So, we always have to struggle for compromises: We will not be able to represent everything in common-sense argumentation. There is no formal theory of common sense – understanding is hermeneutics.

In the modern theory of argumentation, an important impetus was given by Toulmin’s investigations on the uses of argument [27]. Considering the practice of factual argumentation, Toulmin criticizes formal logic used as a critical tool for its insufficient adequateness. Going back to the traditions of topic and rhetorics – which have been parts of traditional logic – he develops a program of a practical informal logic which is supposed to be useful for various kinds of argumentation. An important observation is that all practical argumentation is dependent on the domain of discourse. Toulmin tries to explain the function of expressions relevant for argumentation,

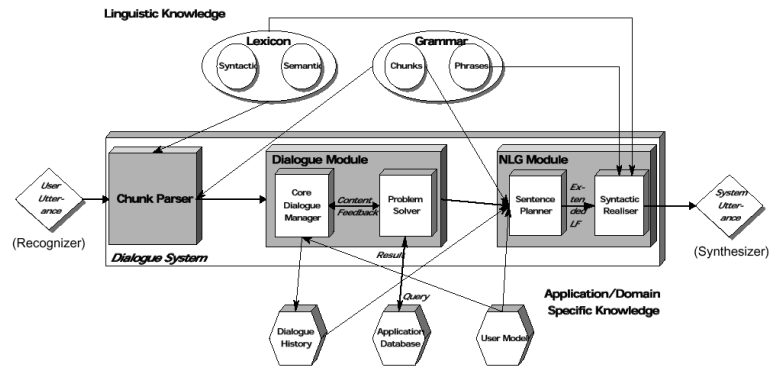


Fig. 1. The Erlangen Dialogue System Architecture

e.g. the logical operators, by the elaboration of argumentation schemata. In the generic case, the distinction between domain-dependent conventions, which guarantee the transition from reasons to the conclusion (warrants), and the justification for those conventions (backing), is important. Keeping Toulmin’s observation of the domain-dependency of argumentation and the requirement of specific argumentation schemata in mind will help us in the search for a flexible, modular decomposition of dialogue system functionality.

C. Selecting a Suitable Representation and Reasoning System

Referring to computational logic has three aspects: First we have the formal logical language, i.e. its syntax and semantics which define a certain expressive power, secondly the level of the reasoning problem, where we have to deal with decidability and computational complexity, and finally the inference procedure with the properties of soundness and completeness. The latter issue also imposes a need for compromises: Often we want to express more than a well-understood complete and sound reasoner can deal with, but then must know exactly what we are doing and we have to keep in mind to stay always as close as possible to completeness and soundness.

Therefore we decided to use description logics [28] as our representational framework. As we will point out in the next section in more detail, we use it for the representation of lexical concepts, as well as for dialogue and application modelling. Using a uniform representation schema on several system levels has the advantage that we need not translate between different level-specific schemata, but there may be a tradeoff.

But as long as we stay compatible with evolving web standards as XML/RDF(S) on the syntactic and DAML+OIL on the semantic level, we have the additional advantage that we can use publicly available resources (thesauri, formal ontologies) and tools.

IV. SYSTEM ARCHITECTURE

As indicated in the introduction, there is no doubt that a minimal prerequisite for scalable systems is that they have a modular structure. Our fundamental design decision consists of a clear functional separation between the language model,

the dialogue model and the domain model⁹, and we claim that it provides a sufficient condition to address scalability. We agree with Allen’s domain-independence hypothesis [1]: “Within the genre of practical dialogue, the bulk of complexity in the language interpretation and dialogue management is independent of the task being performed.”

In addition to the structural aspect of modularization which lays down which components a system has and how they are connected with one another, we have also to consider the temporal structure of their mutual interaction. Although for analysis there is a clear direction of data flow “bottom up”, from signal to action (and back to signal for generation), we already indicated the shortcomings of a purely sequential processing direction. Therefore we will have to introduce feedback loops which allow to make use of predictions from “higher-level” components by “lower-level” ones, hence providing opportunities for incremental processing.

A. Functional Separation of Dialogue Management and Application

The decision to introduce a clear functional separation between dialogue management and application implies the following interaction steps:

- the dialogue manager “formulates a task” for the application;
- the application executes the task;
- the application decides whether it is necessary to inquire the user;
- the application sends task results and further inquiries to the dialogue manager such that it can execute appropriate dialogue operations.

The division of labour between dialogue manager and application in this way is quite radical, but it allows a transparent separation of application and dialogue functions and control flows¹⁰. As far as the administration of application-specific user goals, and in particular the conflict resolution among them is concerned, this has to be provided by the application – as opposed to the administration of dialogue goals cared for by the dialogue manager. Application and dialogue manager are planning separately. The exchange of data must guarantee consistency between the application and the dialogue situation which, of course, requires semantic compatibility. This in turn presupposes that both, dialogue manager and application, have access to the same domain model. Another consequence of the separation is that it leads to a classification of utterances w.r.t. their functionality to change the dialogue situation.

Scalability is supported because the functionality of the application can be augmented without requiring changes to the dialogue manager. Furthermore, by separating the linguistic base (“language model”), the dialogue model, and the application model the reuse of resources is facilitated.

⁹represented as two boxes with the “Dialogue Module” box in the center of fig. 1. The “Problem Solver” box is a placeholder; in fact, of course it is a technical application system which is not a part of the dialogue system, but accessible through a suitable interface.

¹⁰cf. our first paper on functional separation and coordination [29]

1) *Application and Dialogue Knowledge*: From a structural point of view, the conceptual knowledge (concepts or classes, and roles, i.e. binary relations, for their properties) about application and dialogue is represented in two separate, but formally similar terminological hierarchies. They must be inserted as parallel, but disjoint branches into the system’s global conceptual model, which in our case is represented in a description logic.

In particular, the application knowledge, which is used in application situation descriptions, consists of

- concept descriptions of domain objects, and
- concept descriptions of domain actions.

These concepts are instantiated in application situation descriptions that are used to represent which objects of which types currently exist and which actions are possible in the current situation.

So, the application concept hierarchy represents formally reconstructed technical or scientific knowledge, combined with elements of common sense under a technical perspective. In specific application domains it may be possible – as it is the case for EMBASSI – that a considerable part of the application concept hierarchy, i.e. the device-specific concepts, can be gained automatically from a source provided by the application engineers, which in this special case was given as a Java class hierarchy implementing the device control system.

Analogously, the dialogue knowledge used in dialogue situation descriptions, is built up from

- concept descriptions of dialogue objects (utterances, enumeration of alternatives, dialogue goals), and
- concept descriptions of dialogue actions (speech acts).

Dialogue situation descriptions contain instances of those objects; they are extracted from the DRT representation.

The common roof for both hierarchies consists of a generic base model, for which we chose the IEEE SUMO formal ontology, into which both are plugged in. Furthermore, a third branch, which contains lexical concepts, is inserted in this global model. The lexical concepts are derived from a structured lexicon, in our case EuroWordNet¹¹, and they are linked via a specialization role with concepts of the application and dialogue subhierarchies. To establish this mapping from lexical to domain concepts is a rather labor-intensive process and has to be taken up anew whenever the system is configured for a new application. So, (semi-) automatic knowledge acquisition remains as a big problem. Future research should aim at methods for controlled semiautomatic acquisition by supervised learning.

B. Challenges for the Dialogue Manager

An important task for the dialogue manager is to process the interaction between semantics and the domain model. As already outlined at the end of section III-A.2, the last analysis step is the transformation of the domain-independent semantic representations into descriptions of discourse situations which are specialized to the respective application domain of the dialogue system. Within the description logic framework, this

¹¹based on WordNet, cf. [30].

means to map DRSs composed during parsing to ABoxes representing propositions on the current application situation. The general idea is that discourse referents in the domain of discourse refer to instances in the application domain. The interpretation of thematic roles in terms of roles in the application domain is encoded as the application specific part of the word's case frames describing the language usage in the application domain. This means that ABoxes have to be consistent with respect to the given application concept hierarchy (TBox). The rationale for using description logics for constructing natural language semantics is the possibility to eliminate hypotheses constructed by the parser if the corresponding ABoxes are inconsistent. For further technical details, cf. [31], [32].

Usually, semantic representations of natural language utterances cannot directly be mapped into extensional terms of the formal ontology: The meaning of many utterances is not defined by operations of an application. Some utterance parts are not relevant within the domain ontology as e.g. "I would like to..." or "Shall I...". But they are relevant for the dialogue manager to determine the utterance's intended function. "I would like to..." expresses that the user pursues an intention. The dialogue manager must be able to recognize and process this fact. Indeed, it represents a state which can be valid in a given dialogue situation. Hence, a cooperative system must search for circumstances under which the intention can be satisfied.

To process dialogue situations in a flexible (and also extensible) way the dialogue manager has access to a repository of dialogue operations. They are characterized by preconditions and effects w.r.t. the dialogue situation – in analogy to the operations of an application. Dialogue operations are associated to speech acts (or performatives) as "must", "can", "shall", "may", "want", and those expressing conventions (thanking, greeting), etc. The inventory of defined dialogue operations defines a complexity limit for dialogues.

In goal-oriented dialogues, a cooperative system aims at a successful execution of the user's dialogue goals. So, in a given dialogue situation the dialogue manager has to determine the executability conditions of an actual dialogue goal. If it is not able to determine the conditions due to missing information, it has to ask the user – otherwise satisfaction of the dialogue goal fails. This (missing) information is a necessary condition for the satisfiability of the user goal. This means that coherence of an utterance w.r.t. a dialogue goal is determined via the satisfiability relation. Of course, the ability to initiate and conduct clarification subdialogues is a general requirement to the dialogue manager, for example in cases like misunderstandings, recognition errors, or ambiguities in utterances which may occur on all linguistic levels.

Performatives can be explained in terms of dialogue operations and operations on the dialogue system data base, e.g., questions aim at checking satisfiability, statements aim at adding facts to the actual situation description. Furthermore, we have to specify appropriate responses to recognized dialogue operations; in the example of questions it has to be determined under which circumstances the question will be answered.

Regarding application and dialogue states from a technical point of view, in the description logic framework the current state of an application is represented as a set of propositions, a so-called "A-Box" ("possible world"). It contains assertions about instances which refer to objects of the application, and instances for all actions which have been executed by the application up to now. The current state of the dialogue is also represented as an A-Box containing assertions from the dialogue and linguistic domain, i.e. linguistic objects and actions (performatives). Both are related by the requirement that assertions in the dialogue A-Box must be satisfiable w.r.t. the application A-Box. In other words, the dialogue manager "knows" only, what the dialog situation represents, and the actions it can perform are dialogue actions, but as soon as domain content is referred, the application situation description comes into play.

As outlined above, in reasoning we must be able to deal with partial information. For this purpose, Nait Abdallah developed in [26] a reasoning algorithm based on a tableau calculus. Situation knowledge is being processed in a tableau: Its contents can be modified by user or system messages; leaves of open branches represent possible information states of the dialogue manager at a given time. Inferences on tableau consistency are drawn using domain concept definitions. For our current system, we implemented a special prover to handle partial information. Although operational, a more elegant and better integrated solution is desirable. Since modern description logic reasoners are tableau-based as well, an extension to implement hypothetical reasoning in A-Boxes could provide a solution. The fundamental technical requirement is a facility to deal with multiple A-Boxes – where in our case each would represent a different extension of the actual situation description –, is already available.

V. CONCLUSIONS

In the search for answers to the question how theory can contribute to the construction of scalable speech dialogue systems, we considered theoretical aspects from linguistics and logic.

For the linguistic part, we claimed that "pragmatics-first" view on rational interaction provides an appropriate framework for flexible and scalable dialogue modelling. In particular, the plan-based approach offers the means to conduct task- or goal-oriented dialogues which aim at accomplishing concrete tasks. It enables cooperative response behaviour and the ability for negotiation.

For the reasoning part, i.e. knowledge representation and inference for the interpretation of dialogue as well as for planning to satisfy user goals in the application domain, we argued for a computational logic framework.

There is no doubt that a minimal prerequisite for scalable systems is that they have a modular structure. We argued that a clear functional separation between the language model, the dialogue model, and the domain model provides a sufficient condition to address scalability.

VI. ACKNOWLEDGEMENTS

The present article would not have been possible without the work of our project collaborators Kerstin Bücher, Yves Forkl, Martin Klärner, Bernd Ludwig, Peter Reiss and Iman Thabet. I am also very grateful to them for valuable remarks on a draft of this paper. All remaining mistakes are of course in the author's responsibility.

REFERENCES

- [1] J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent, "Toward conversational human-computer interaction," *AI Magazine*, vol. 22, no. 3, pp. 27–37, 2001.
- [2] W. Eckert, G. Görz, G. Hanrieder, W. Hiltl, H. Niemann, and G. Schukat-Talamazzini, "A robust information system for spoken human-machine dialogues," in *Proceedings of ICASSP-94*, Adelaide, April 1994.
- [3] M. Boros, E. Wieland, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann, "Toward understanding spontaneous speech: Word accuracy vs. semantic accuracy," in *Proceedings of ICSLP-96 – International Conference on Spoken Language Processing*, Philadelphia, October 1996, pp. 1005–1008.
- [4] P. R. Cohen, J. Morgan, and M. E. Pollack, Eds., *Intentions in Communication*, ser. System Development Foundation Benchmark Series. Cambridge, Mass. and London: A Bradford Book, The MIT Press, 1990.
- [5] M. Sadek, "A study in the logic of intention," in *Proceedings of ECAI-96*, 1996, pp. 462–473.
- [6] M. Sadek, B. Bretier, and F. Panaget, "Artimis: Natural dialogue meets rational agency," in *Proceedings of IJCAI-97*, 1997, pp. 1030–1035.
- [7] M. Sadek, "Design considerations on dialogue systems: From theory to technology – the case of artimis –," in *Proceedings of the ESCA Workshop "Interactive Dialogue in Multi-Modal Systems*, Kloster Irsee, 1999, pp. 173–187.
- [8] B. Ludwig, G. Görz, and H. Niemann, "An inference-based approach to he interpretation of discourse," *Language and Computation*, vol. 1, no. 2, pp. 261–276, 2000.
- [9] G. Görz, K. Bücher, Y. Forkl, M. Klärner, and B. Ludwig, "Speech dialogue systems – a "pragmatics-first" approach to rational interaction," in *Natural Language Processing between Linguistic Inquiry and System Engineering. Festschrift für Walther von Hahn*, W. Menzel, Ed., Hamburg, 2002, in print.
- [10] N. Chomsky, Ed., *Aspects of the theory of syntax*. Cambridge, Mass.: The MIT Press, 1965.
- [11] D. Traum and J. Allen, "Discourse obligations in dialogue processing," in *Proceedings of ACL 94*, Les Cruces, 1994, pp. 1–8.
- [12] G. Görz, *Strukturanalyse gesprochener Sprache – Ein Verarbeitungsmodell*. Bonn: Addison-Wesley, 1988.
- [13] S. Abney, "Statistical methods and linguistics," in *The Balancing Act. Combining Symbolic and Statistical Approaches to Language*, ser. Language, Speech, and Communication, J. L. Klavans and P. Resnik, Eds. Cambridge, Mass. and London: The MIT Press, 1986, ch. 1, pp. 1–26.
- [14] K. Bücher, M. Knorr, and B. Ludwig, "Ambiguity reports for flexible dialog management," in *Advances in Natural Language Processing. Third International Conference, PorTAL 2002*, ser. Lecture Notes in Artificial Intelligence, E. Ranchhod and N. Mamede, Eds., vol. 2389, Lyon, 2002, p. 151ff.
- [15] —, "Anything to clarify? report your parsing ambiguities!" in *Proceedings of ECAI-02*, Lyon, 2002, pp. 465–469.
- [16] H. Kamp and U. Reyle, *From Discourse to Logic*. Dordrecht: Kluwer, 1993.
- [17] I. Fischer, B. Geistert, and G. Görz, "Incremental semantics construction and anaphora resolution using lambda-drt," in *Proceedings of DAARC-96 – Discourse Anaphora and Anaphor Resolution Colloquium*, S. Botley and J. Glass, Eds., Lancaster, July 1996, pp. 235–244, poster in: Proceedings of the Fourth International Workshop on Parsing Technologies, Universitas Carolina, Prague and Karlsbad, 1995, 87–88.
- [18] P. Blackburn and J. Bos, Eds., *Representation and Inference for Natural Language*, 2 vols. Saarbrücken: <http://www.comsem.org>, 1999.
- [19] P. R. Cohen and H. J. Levesque, "Persistence, intention, and commitment," in *Intentions in Communication*, ser. System Development Foundation Benchmark Series, P. R. Cohen, J. Morgan, and M. E. Pollack, Eds. A Bradford Book, The MIT Press, 1990, ch. 4, pp. 33–69.
- [20] M. Poesio and D. Traum, "Towards an axiomatisation of dialogue acts," in *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues*, J. Hulstijn and A. Nijholt, Eds., Enschede, 1998, pp. 207–222.
- [21] —, "Conversational actions and discourse situations," *Computational Intelligence*, vol. 13, no. 3, pp. 309–347, 1997.
- [22] N. Asher and A. Lascarides, "Cognitive states, discourse structure and the content of dialogue," in *Proceedings of the Amsterdam Dialogue Workshop, Amstelogue-99*, Amsterdam, 1999.
- [23] B. L. Grosz and C. L. Sidner, "Attentions, intentions, and the structure of discourse," *Computational Linguistics*, vol. 12, no. 3, pp. 175–204, 1986.
- [24] —, "Plans for discourse," in *Intentions in Communication*, ser. System Development Foundation Benchmark Series, P. R. Cohen, J. Morgan, and M. E. Pollack, Eds. A Bradford Book, The MIT Press, 1990, ch. 20, pp. 417–444.
- [25] C. Rich, C. L. Sidner, and N. Lesh, "Collagen – applying collaborative discourse theory to human-computer interaction," *AI Magazine*, vol. 22, no. 3, pp. 15–25, 2001.
- [26] M. Nait Abdallah, *The Logic of Partial Information*. New York: Springer, 1995.
- [27] S. E. Toulmin, *The Uses of Argument*. Cambridge: Cambridge University Press, 1964.
- [28] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, "Reasoning in description logics," in *Foundations of Knowledge Representation*, G. Brewka, Ed. Stanford: CSLI Publications, 1996, pp. 191–236.
- [29] A. Brietzmann and G. Görz, "Pragmatics in speech understanding – revisited," in *COLING 82. Proceedings of the Ninth International Conference on Computational Linguistics, Prague, July 5–10, 1982*, ser. North-Holland Linguistic Series, J. Horecky, Ed., vol. 47. Amsterdam: Academia and North-Holland Publishing Company, 1982, pp. 49–54.
- [30] C. Fellbaum, *WordNet – An Electronic Lexical Database*, ser. Language, Speech, and Communication. Cambridge, Mass.: MIT Press, 1998.
- [31] K. Bücher, G. Görz, and B. Ludwig, "Corega tabs: Incremental semantic composition," in *KI-2002 Workshop on Applications of Description Logics, Proceedings*, ser. CEUR Proceedings, G. Görz, V. Haarslev, C. Lutz, and R. Möller, Eds., vol. 63, Gesellschaft für Informatik e.V. Aachen: CEUR, September 2002.
- [32] B. Ludwig, K. Bücher, and G. Görz, "Corega tabs: Mapping semantics onto pragmatics," in *KI-2002 Workshop on Applications of Description Logics, Proceedings*, ser. CEUR Proceedings, G. Görz, V. Haarslev, C. Lutz, and R. Möller, Eds., vol. 63, Gesellschaft für Informatik e.V. Aachen: CEUR, September 2002.