Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-94-34

COSMA – Multi-Participant NL Interaction for Appointment Scheduling

**Stephan Busemann, Stephan Oepen,
Elizabeth A. Hinkelman, Günter Neumann, Hans Uszkoreit**

October 1994

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: + 49 (631) 205-3211
Fax: + 49 (631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: + 49 (681) 302-5252
Fax: + 49 (681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, Sema Group, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct systems with technical knowledge and common sense which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ☐ Intelligent Engineering Systems
- ☐ Intelligent User Interfaces
- ☐ Computer Linguistics
- ☐ Programming Systems
- ☐ Deduction and Multiagent Systems
- ☐ Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Dr. Dr. D. Ruland

Director

COSMA

Multi-Participant NL Interaction for Appointment Scheduling

Stephan Busemann, Stephan Oepen,
Elizabeth A. Hinkelman, Günter Neumann, Hans Uszkoreit

DFKI-RR-94-34

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITWM-94 02).

ISSN 0946-008X

COSMA
Multi-Participant NL Interaction for Appointment Scheduling

Stephan Busemann, Stephan Oepen,
Elizabeth A. Hinkelman, Günter Neumann, Hans Uszkoreit
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbrücken (Germany)
{ busemann | oe | hinkelman | neumann | hansu }@dfki.uni-sb.de

Abstract

We discuss the use of NL systems in the domain of appointment scheduling. Appointment scheduling is a problem faced daily by many people and organizations, and typically solved using communication in natural language. In general, cooperative interaction between several participants is required whose calendar data are distributed rather than centralized. In this distributed multi-agent environment, the use of NL systems makes it possible for machines and humans to cooperate in solving scheduling problems.

We describe the COSMA (Cooperative Schedule Managament Agent) system, a secretarial assistant for appointment scheduling. A central part of COSMA is the reusable NL core system DISCO, which serves, in this application, as an NL interface between an appointment planning system and the human user. COSMA is fully implemented in Common Lisp and runs on Unix Workstations. Our experience with COSMA shows that it is a plausible and useful application for NL systems. However, the appointment planner was not designed for NL communication and thus makes strong assumptions about sequencing of domain actions and about the error-freeness of the communication. We suggest that further improvements of the overall COSMA functionality, espescially with regard to flexibility and robustness, be based on a modified architecture.

The paper is organized as follows. Section 2 discusses properties of the domain of appointment scheduling and some related work. Section 3 presents the COSMA system and discusses its advantages and drawbacks. Section 4 presents a modified approach to COSMA.

Contents

1 Appointment Scheduling

Appointment scheduling is a problem faced daily by many people and organizations, and typically solved using communication in natural language. In general, cooperative interaction between several participants is required. In this paper we are concerned with meetings all participants should attend, e.g. business meetings, thus excluding e.g. invitations for social events the date of which is typically not negotiated.

Since appointments are often scheduled after a sequence of point-to-point connections with several participants, e.g. by phone, this will at times necessitate repeated rounds of communication until all participants agree to some date and place. This is a very time-consuming task that should be automated.

Theoretically the task could be simplified by implementing some central planning facility that has access to all individual and collective spatio-temporal constraints of the individual participants. In practice such a centralized solution is not feasible since it would reduce the circle of participants to the members or employees of one organization. But even for employees of the same organization, privacy considerations should rule out full access to individual calendars. Therefore, a distributed solution based on the notion of cooperating agents is preferred. Such an approach also adds flexibility, for it allows individual human participants to act more freely in the negotiation process.

We have designed and implemented the cooperative appointment scheduling manager COSMA, an extended personal calendar system, that assists its owner in scheduling appointments and reacting to other agents' appointment requests. The interaction of humans and machines in the scheduling process necessitates the use of natural language. Human language is required since it cannot be expected that all partners use the same program for managing their calendars. For quite some time there will be partners who do not use any software system for this purpose. Natural language is used for the communication with these people. Natural language is also relevant for the monitoring of the negotiation process by the human owners of the system in cases where they need to interfere.

The domain of cooperative scheduling management has many attractive features for both realistic NL applications and advanced research:[1]

- There is a real demand for the application.
- The application domain is relatively simple and can be handled by existing AI technology.
- The application is much more feasible than most attempted NL applications.
- The application domain exhibits an interesting sublanguage.
- The NL functionality is not mission-critical, messages on which the system fails can be given to the human user. The user can select a set-up under which no decision will be made without user confirmation.

[1]The same domain is used by the project VerbMobil [Wahlster 1993] for the translation of spoken dialogues.

- The functionality of the application can be extended in many ways. The system may be extended to account for e.g. an agent's social status, for travel times, or the function of a user alert system.
- The domain of the application can be extended in many ways. Many logistics problems, e.g. fleet logistics or room utilization, require a very similar functionality.
- Corpora of appointment messages for improving the system and for measuring progress can easily be obtained.
- The application lends itself to easy in-house testing.

1.1 Related Work

While electronic calendars are becoming widespread, computerized support of the scheduling process that includes NL capabilities was not available until recently. Notable among implemented research systems in this domain are those of Mattern [Mattern and Sturm 1989], Dent et al. [Dent *et al.* 1992], and the MECCA system [Lux *et al.* 1992; Lux 1992].

Mattern's system is a (non-AI) distributed systems technology appointment arrangement application. It focuses on fine-tuning the user interface, paying particular attention to convenient expression of recurring constraints, personal priorities, and temporal preferences conceived as scalar values. Dent et al.'s system, on the other hand, is an experimental platform for learning systems. It is used for comparing neural-net and statistical factor analysis methodologies. In stark contrast to Mattern, the user interface is reduced to the prompting structure of a single new Emacs editor function. This function makes entries in the user's online diary, while the system itself provides (learned) intelligent defaults for each field value.

MECCA was developed by the DFKI project KIK-TEAMWARE and Siemens AG, is functionally the most similar to our system, primarily because our calendar planner (provided by the DFKI group AKA-MOD) grew out of the same work. Although in MECCA no attempt was made to incorporate NL capabilities, it, too, is concerned with the mechanics of making and amending appointments between multiple participants. The basic hypothesis of this work, however, was that such cooperative behaviour is *independent* of the communications medium; their system uses communication between agents that is machine readable only, and the planning component is architecturally isolated from the properties of this channel.[2]

COSMA could draw upon much of this previous work, but its focus is different. Most notably, the use of NL makes it possible for machines and humans to cooperate in solving scheduling problems.

[2]Recently a cooperation has been formed between the KIK-TEAMWARE follow-up project COMMA and the COSMA project, aiming at integrating distributed cooperating agent systems and natural language communication.

2 The COSMA Prototype System

2.1 Overall Set-Up

With its NL capabilities COSMA functions as a secretarial assistant in a distributed scenario. It provides the following services:

- storage and organization of a personal calendar;
- graphical display and manipulation of appointment data;
- natural language understanding and generation in communication with other agents via electronic mail.

Correspondingly, the system falls into three major components: (i) an appointment planner (developed by the DFKI project AKA-MOD) that keeps the calendar database, provides temporal resolution and drives the communication with other agents; (ii) a graphical user interface (GUI) (developed inside project DISCO) monitoring the calendar state and supporting the mouse- and menu-driven arrangement of new appointments; and (iii) the DISCO NL core system (enriched with a set of application specific modules) that provides the natural language capabilities (see [Uszkoreit *et al.* 1994] for an overview over DISCO).

In this set-up DISCO is connected to an external application system that comes with a rigid interface, viz. the appointment planner internal representation (IR) language. Because IR is the only interface between the two modules, all knowledge about calendar actions resides within the planner, thus posing several limitations on dialogue and generation (see below).

The COSMA system is connected to the outer world through its GUI (as a convenient and plausible interface for the local user) and an interface to standard e-mail. The use of e-mail is among the most natural ways to schedule appointments; agents act asynchronously and save precious time. Since in talking to remote agents by e-mail it may be unknown whether the addressee runs another COSMA instance or not, outgoing messages generated from the DISCO system always contain both NL and IR representations; this way whatever addressee is free to pick the appropriate form: human(oid) agents read plain NL text, artificial ones IR. Consequently, any COSMA system has to be able to process NL, both in order to understand NL dialogue contributions and to produce them.

Figure 1 gives an overview of the system architecture. We describe those parts of the COSMA installation that are relevant to the application task at hand.

The DISCO system is a reusable NL core machine that relates NL sentences to surface-level semantic expressions and vice versa. Analysis and generation processes are based on a declarative and constraint-based HPSG-style grammar [Pollard and Sag 1987]. This allows for a highly modular, although strongly interconnected, representation of linguistic knowledge [Netter 1993]. Although the analytical depth of the core machinery was restricted in order to preserve its domain and task independency, the uniform constraint formalism adopted [Krieger and Schäfer 1994] makes it possible to include, on demand, further linguistic strata in a modular way. For the scheduling task at hand, linguistic surface speech

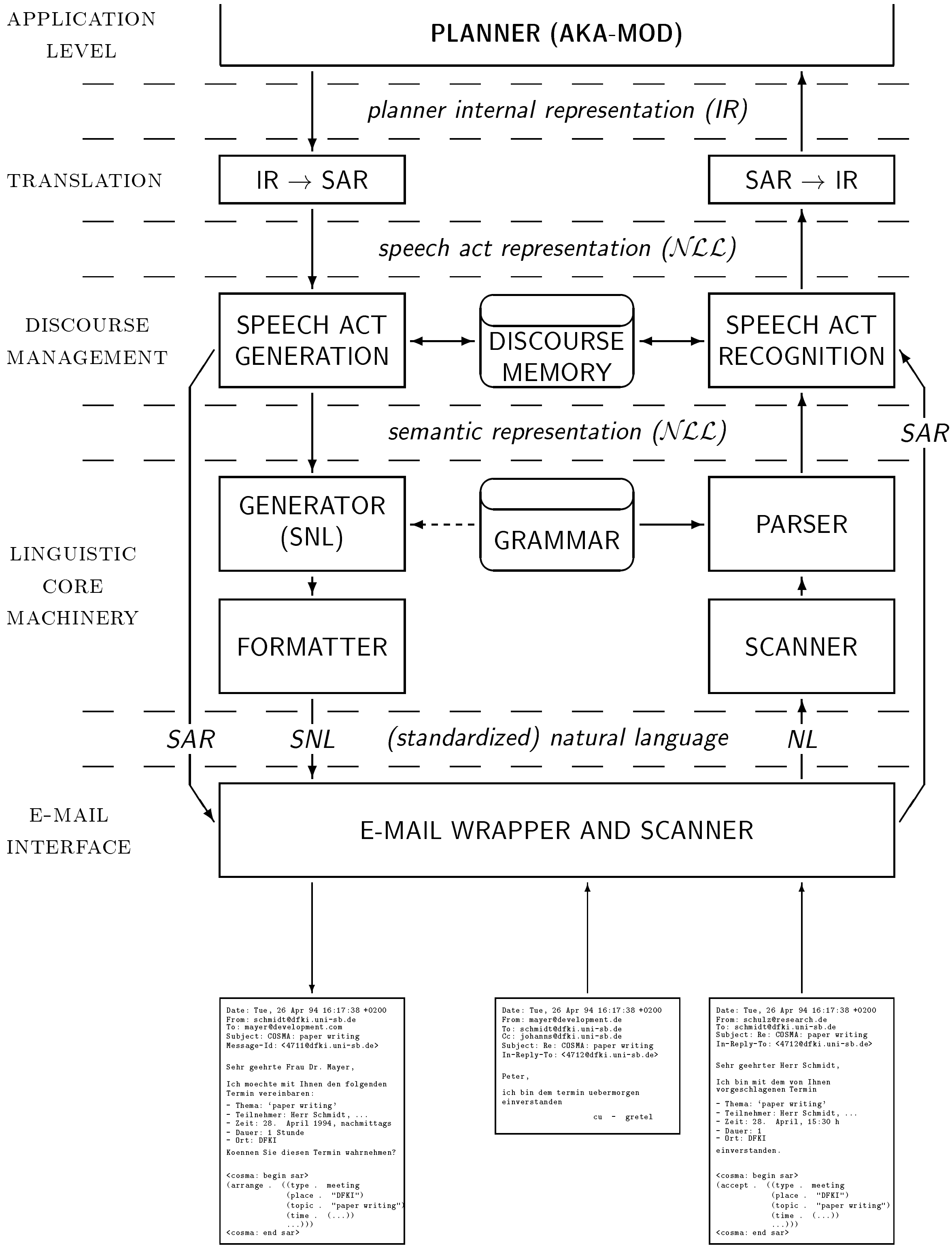

Figure 1: Sketch of the COSMA internal architecture.

```
((from . "schmidt@dfki.uni-sb.de")
 (to . ("mayer@development.de" "schulz@research.de"))
 (message-id . 4711)
 (actions . ((arrange ((type . :meeting)
                       (topic . "ANLP paper writing")
                       (participants . ("schmidt@dfki.uni-sb.de"
                                        "mayer@development.de"
                                        "schulz@research.de"))
                       (time . ((year . 1994)
                                (month . :april) (day . 28)
                                (time-of-day . :afternoon)))
                       (place . "DFKI"))))))
```

Figure 2: Internal Representation of the COSMA appointment planner.

act representations were integrated to constrain the set of possible readings as early as possible.

The specialized meaning representation formalism $\mathcal{NLL}$ (originally developed at Hewlett Packard and substantially extended at DFKI) is used in DISCO for semantic reasoning and as a flexible interface to various application systems. $\mathcal{NLL}$ is a logical form semantic representation language borrowing from various schools of linguistic semantics (e.g. the generalized quantifier theory and the logic of plurals). It is designed to provide a general-purpose semantics module, thus extending the scope of domain- and application-independent linguistic processing [Nerbonne *et al.* 1993b]. In the COSMA application $\mathcal{NLL}$ was extended to cover the interface between NL semantics and the COSMA appointment planner.

The planner follows the paradigm of [Lux *et al.* 1992] in that it implements a finite state protocol based on a fixed set of negotiation primitives (e.g. `arrange()`, `accept()`, `reject()`, `refine()` etc.) taking full or partial descriptions of appointments as arguments. Typically, a scheduling interaction is initiated by broadcasting an appointment suggestion (`arrange()`); remote agents will then reply to this message (`accept()`, `refine()`, `modify()` or `reject()`) and it is the task of the initiator to decide whether all agents have committed to a compatible set of constraints (regarding the appointment parameters) or whether further pairwise e-mail exchange is required. Once all participants have come to an agreement, the final appointment data is broadcast (`confirm()`) and personal calendars updated. Figure 2 shows a sample IR expression with its `actions` field resulting from DISCO analysis.

Generation from IR expressions demands a compromise between theoretical and practical requirements. Since IR does not make a distinction between the knowledge of the planner and its communicative goal and DISCO has no access to planner internal data, the full-fledged linguistic competence system is not needed to verbalize IR structures faithfully. Instead a template-based approach has been adopted generating the bulk of appointment

```
Date: Tue, 26 Apr 94 16:17:14 +0200
Message-Id: <4711@cl.dfki.uni-sb.de>
Received: by cl.dfki.uni-sb.de; Tue, 26 Apr 94 16:17:14 +0200
From: schmidt@dfki.uni-sb.de (Peter Schmidt)
To: mayer@development.de, schulz@research.de
Subject: ANLP paper writing

Sehr geehrte Frau Dr. Mayer, sehr geehrter Herr Schulz!

Ich moechte mit Ihnen den folgenden Termin vereinbaren:

  Zeit:         Donnerstag, der 28. April 1994 nachmittags
  Ort:          DFKI
  Teilnehmer:   Dr. Mayer, Herr Schulz, Prof. Dr. Schmidt
  Dauer:        1 Stunde
  Thema:        ANLP paper writing

Mit freundlichen Gruessen,
Peter Schmidt

--- Dieser Text wurde automatisch durch COSMA generiert ---
```

Figure 3: COSMA e-mail message requesting an appointment.

data in tabular form (see Figure 3 for a sample message[3]).

The COSMA architecture was implemented on top of the DISCO development shell, which serves as the architectural platform for the integration of system components [Neumann 1993]. The generic frame system neither depends on specific components to be used or a particular flow of control. Using an object-oriented design together with multiple inheritance increases the system's modifiability, extensibility and incremental usability.

2.2 Sample e-mail Interaction

Although the restricted appointment task language of the COSMA planner was not designed for NL communication, it provides for a realistic functionality in multi-participant appointment scheduling, as is examplified by the following interaction between three agents, Dr. Mayer, Mr. Schulz and the COSMA of Prof. Dr. Schmidt running autonomously.

[3] The German message body translates into "I would like to arrange the following meeting with you: ..."

(1) **Schmidt's COSMA to Mayer and Schulz (NL plus IR):**
[I would like to arrange with you the following meeting: ...]
(see Figures 2 and 3)

(2) **Schulz to Schmidt (NL):**
"Ich kann an dem Termin übermorgen teilnehmen."
[I can participate in the meeting on the day after tomorrow.]

(3) **Mayer to Schmidt (NL):**
"Können wir uns am Donnerstag um 15:30 Uhr treffen?"
[Can we meet on Thursday at 3:30 pm?]

(4) **Schmidt's COSMA to Mayer and Schulz (NL plus IR):**
[All participants have agreed to the following meeting: ...]

Schmidt uses the graphical user interface of his COSMA to request a one hour meeting with Mayer and Schulz for the afternoon of April, the 28$^{\text{th}}$ to discuss a joint paper writing. Once he has finished the input, the appointment planner reserves the time in question in Schmidt's calendar and uses the DISCO system to verbalize the `arrange()` request in e-mail to both Mayer and Schulz. Note that this initial request is vague with respect to the meeting time (message (1)). Schulz reads the NL part of the incoming e-mail and replies affirmatively in plain text (message (2)). Mayer checks her old-fashioned (leather-bound) calendar, and since she has a meeting until 3 pm, she suggests 3:30 as a meeting time (message (3)).

Schmidt's COSMA receives the incoming e-mail and has the DISCO system analyze its body as either a request for another meeting or the refinement of one under negotiation thus leaving it to the appointment planner for disambiguation. Since 3:30 is within the time slot initially suggested by Schmidt, his COSMA commits to the appointment and (by a message analogue to the one in Figure 3) notifies all the participants of the scheduling success and the refined appointment data (message (4)). Finally, the planner updates Schmidt's calendar.

2.3 Interfacing DISCO to the COSMA Appointment Planner

The interface between the DISCO core machinery and the COSMA appointment planner internal representation language (IR) extends the approach successfully employed in the DISCO semantics module already (see above). Based on the assumption that programming languages and meaning representation languages share a number of common design goals, semantic processing in $\mathcal{NLL}$ and interface issues are heavily inspired from traditional compiler technology. Manipulation of $\mathcal{NLL}$ expressions and translation into and out of the $\mathcal{NLL}$ semantics module are therefore viewed and implemented as tree transformations on the $\mathcal{NLL}$ *abstract* syntax (see presently) that are specified as *meta-syntactic* rewrite rules over the *concrete* $\mathcal{NLL}$ syntax ([Nerbonne *et al.* 1993a] argue for the parallelism between

$\mathcal{NLL}$ and traditional compiler technology).

The $\mathcal{NLL}$ transformations implemented so far in the COSMA prototype can be classified according to their theory vs. domain dependence and the target language ($\mathcal{NLL}$ vs. IR):

(I) **core $\mathcal{NLL}$ inference**: purely logical (theory-independent) equivalence transformations, e.g. flattening of nested conjunctions, removal of non-constraining variables;

(II) **simplification**: meaning preserving transformations generating a 'canonical' form, e.g. the grouping of multiple restrictions on the same variable;

(III) **disambiguation and domain-specific inference**: transformations using domain knowledge, e.g. scope resolution, anchoring of underspecified temporal locations (in rudimentary implementation by now);

(IV) **translation**: mapping into a *different* abstract syntax — mediation of difference in expressive power, e.g. translation to SQL or the COSMA planner internal representation.

While a set of core logical rewrite rules comes built-in with the $\mathcal{NLL}$ semantics module, the second class of transformations listed above depends on the concrete linguistic theory of semantics assumed and again was implemented in the compiler approach. Its main purpose is to abstract away from (mostly syntactical) surface conditions — i.e. the structure derived from the DISCO grammar — in the initial semantics (in feature structures) as it is input to the semantics module thus yielding a quasi-normalized semantic representation.

Likewise, in adapting the compiler approach to the $\mathcal{NLL}$ to IR interface, the translation module is construed as a set of rewrite rules that are applied in post-oder (bottom-up) tree traversal just like the other classes of transformations given before.[4] The tree transformation mechanism has turned out to be suitable in handling larger and more complex structures as well and to effectively support the mediation of different expressive power needed in mapping $\mathcal{NLL}$ into COSMA IR and allows for rapid prototyping.

2.4 Experiences with COSMA

With COSMA we have shown how a NL core system can serve as an interface to an existing application system, thus enhancing the communicative capabilities of the latter considerably. Work on COSMA was successful also in other respects:

- The object-oriented framework proved astonishingly flexible, permitting the final integration of about ten modules in one two-week period.

[4] While in the current prototype all four classes of $\mathcal{NLL}$ transformations are still implemented as sets of hand-coded rewrite functions that are applied by use of an optimized tree traversal driver, meanwhile a rule compiler has been developed allowing to generate the appropriate rewrite functions from declarative meta-syntactic rewrite specifications.

- We were able to utilize the economies of uniform feature-structure formalism and unification to avoid the overheads introduced by translation, incommensurateness of formal bases, and synchronization. In particular, simultaneous availability of lexical, syntactic, and semantic features was a great asset in implementation of the speech act recognition method of [Hinkelman and Allen 1989]. This method takes advantage of both semantic representations and other features to which no formal semantics has been assigned.
- New research was stimulated by the necessity of connecting DISCO to such a rigid interface as IR. General methods of linking competence-based NL core systems to application systems are being developed, including methods for deriving performance-oriented sublanguages [Uszkoreit *et al.* 1994] and configurable generation components [Busemann 1993] that can work with simple techniques, but have available a full linguistic knowledge base as a fall-back.
- The decision to allow desired speech acts to be chosen by the application was fundamentally sound, since the application has a better grasp of its own state and may potentally have many other goals which are not communicative ones.

Besides these positive and stimulating experiences, we were also able to show that some extensions and refinements are necessary preconditions for further advancement in using NL in this application.

Most importantly, specific finite-state protocols do not allow for sufficient flexibility and robustness in NL dialogue. A corpus of e-mail exchanges on appointment scheduling collected at DFKI reveals that sequences of actions are much less regular than predicted by finite state protocols. In addition, the protocols assume an error-free communication, but NL utterances are often erroneous.

We suggest that flexibility and robustness can be sufficiently increased by a fully compositional model of dialogue where sequencing of performed actions is based on an explicit and formal semantics of speech acts and where theories of repair are used, which reveal the recovery strategies naturally followed in human-human communication. These techniques would require more sophisticated exchange of information between domain-independent and application-specific components. A rigid and simple interface such as IR would not suffice anymore; rather, we suggest to redraw the borderline between the reusable core engine and the application system by incorporating application-independent reasoning and knowledge representation into the core system.

3 An Extended Approach to COSMA

In order to increase the range of utterances that can be processed, an inferential component that makes use of context is desireable. Consider asking someone if they can meet you in Saarbrücken on the 28^{th}. If you get the answer “I'll be in Berlin then.”, you want to conclude that your suggestion has been rejected. How could this be handled? The desired inference steps result into something like

- We believe Y will be in Berlin then.
- We believe people can't be in two nonoverlapping places at once.
- We believe Berlin and Saarbrücken don't overlap.
- We believe Y can't be in Berlin and Saarbrücken then.
- We believe Y won't be in Saarbrücken then.
- We believe Y can't attend the meeting as suggested.
- We believe Y rejects the suggestion.

This calculation requires general knowledge about space and time, knowledge of the Saarbrücken suggestion in this specific context, and knowledge of the definitions of communicative and domain-specific actions. It also requires general theorem proving and context manipulation facilities.

It may be objected that this kind of indirect speech act is of little relevance to application systems. But it will occur in conversations dominated by human beings, such as in the VerbMobil scenario [Wahlster 1993]. There are also hosts of smaller problems which can individually be papered over in the lexicon, but will not be addressed with any generality until the lexicon or its inferential partner contains enough information to show the semantic relationships between words. Consider "I can't come on the 28^{th}", where we want the system to interpret *come* in the sense of *attend*.

The new architecture makes reasoning about actions accessible to the linguistic system, placing it between the propositional semantics level and the application programs. The familiar parsing and generation components from DISCO are organized around a blackboard's central controller.

The new inference-based level receives as input a set of speech act interpretations in propositional form, which are translated into action representations with preconditions and effects. The interpretations found are then filtered for consistency with context, according to the method of [Hinkelman and Allen 1989].

Each context contains the beliefs of some individual or group as modelled by the system. Our belief model is fully implemented [Hinkelman and Spackman 1994], and has the unique property of attributing beliefs directly to groups rather than sets of individuals. In particular, the conversational record is also a group belief context. Speech acts have their effects directly on these conversational group belief contexts, which then propagate to participant contexts through default inheritance.

In particular, the system itself establishes, after each utterance, its current goals. It plans to resolve these goals; the resulting hierarchical plan structure may involve actions of back end programs, e.g. a calendar database or an e-mail addressbook. These application program actions are modelled within the system in just enough detail for planning, and are invoked via procedural attachment.

Note that while it may be possible to plan several application program actions in advance, the dialogue aspects of appointment arrangement must be planned on a reactive basis. It is simply not known what speech act other conversational participants will perform next.[5]

Planned speech acts are also realized by invocation of the generation process through procedural attachment.

In this modified approach the dialogue component has access to an ontology of domain concepts as well as domain actions and state. It thus has the necessary prerequisites for analysis of such utterances as “I'll be in Berlin then.”

Our experiences with the new architecture thus far have shown that while great potential flexibility is obtained through this architecture, further work is required in the areas of uniformity of representation and of application-independent external interfaces.

4 Conclusion

We have presented the COSMA system for appointment scheduling in a distributed multi-agent environment. It adds NL communication capabilities to a external appointment planner, thus allowing humans and machines to cooperatively solve scheduling problems.

Since NL poses additional requirements on the flexibility and robustness of the system that could hardly be met within the first system's framework, we suggested a modified approach that investigates what amount of reasoning about actions needs to be incorporated into the NL core machine. This work is continued in a DFKI project on COSMA.

[5] Various studies have found that even with a carefully limited domain and speech act set, a three-way ambiguity must be allowed, to obtain 80% reliability.

References

[Busemann and Harbusch 1993] Stephan Busemann and Karin Harbusch. DFKI Workshop on Natural Language Systems: Re-usability and Modularity. Proceedings. Document D-93-03, Saarbrücken, Germany, 1993.

[Busemann 1993] Stephan Busemann. Towards Configurable Generation Systems. Some Initial Ideas. In: [Busemann and Harbusch 1993].

[Dent *et al.* 1992] Lisa Dent, Jesus Boticario, John McDermott, Tom Mitchell, and David Zabowski. A Personal Learning Apprentice. In: *Proceedings of the National Conference on Artificial Intelligence*, 1992.

[Hinkelman and Allen 1989] E.A. Hinkelman and J.F. Allen. Two Constraints on Speech Act Ambiguity. pages 212–219, Vancouver, 1989.

[Hinkelman and Spackman 1994] Elizabeth A. Hinkelman and Stephen P. Spackman. Communicating with Multiple Agents. In: *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, Kyoto, Japan, 1994.

[Krieger and Schäfer 1994] Hans-Ulrich Krieger and Ulrich Schäfer. $\mathcal{TDL}$—A Type Description Language for Constraint-Based Grammars. In: *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, Kyoto, Japan, 1994.

[Lux *et al.* 1992] Andreas Lux, Frank Bomarius, and Donald Steiner. A Model for Supporting Human Computer Cooperation. In: *AAAI Workshop on Cooperation among Heterogeneous Intelligent Systems*, San Jose, Ca., 1992.

[Lux 1992] Andreas Lux. A Multi-Agent Approach Towards Group Scheduling. Research Report RR-92-41, DFKI (Kaiserslautern, Germany), 1992.

[Mattern and Sturm 1989] F. Mattern and P. Sturm. An Automated Distributed Calendar and Appointment System. Technical Report SFB124-24/89, University of Kaiserslautern, Department of Computer Science (Kaiserslautern, Germany), 1989.

[Nerbonne *et al.* 1993a] John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, and Stephan Oepen. Software for Applied Semantics. In: Chu-Ren Huang, Claire Hsun hui Chang, Keh jiann Chen, and Cheng-Hui Liu (editors), *Proceedings of Pacific Asia Conference on Formal and Computational Linguistics*, pages 35–56, Taipei, 1993. Academica Sinica. Also available as DFKI Research Report RR-92-55.

[Nerbonne *et al.* 1993b] John Nerbonne, Stephan Oepen, Abdel Kader Diagne, Karsten Konrad, and Ingo Neis. $\mathcal{NLL}$—Tools for Meaning Representation. In: [Busemann and Harbusch 1993], pages 43–50.

[Netter 1993] Klaus Netter. Architecture and Coverage of the DISCO Grammar. In: [Busemann and Harbusch 1993].

[Neumann 1993] Günter Neumann. Design Principles of the DISCO System. In: *Proceedings of the Twente Workshop on Language Technology (TWLT5)*, University of Twente, 1993.

[Pollard and Sag 1987] Carl J. Pollard and Ivan A. Sag. *Information-Based Syntax and Semantics. Vol. 1: Fundamentals.* Chicago (Univ. of Chicago Press), 1987. Center for the Studies of Language and Information, CSLI Lecture Note No. 13.

[Uszkoreit *et al.* 1994] Hans Uszkoreit, Rolf Backofen, Stephan Busemann, Abdel Kader Diagne, Elizabeth A. Hinkelman, Walter Kasper, Bernd Kiefer, Hans-Ulrich Krieger, Klaus Netter, Günter Neumann, Stephan Oepen, and Stephen P. Spackman. DISCO–An HPSG-based NLP System and its Application for Appointment Scheduling. In: *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, Kyoto, Japan, 1994.

[Wahlster 1993] Wolfgang Wahlster. VERBMOBIL–Translation of Face-to-Face Dialogs. In: Otthein Herzog, Thomas Christaller, and Dieter Schütt (editors), *Grundlagen und Anwendungen der Künstlichen Intelligenz. 17. Fachtagung für Künstliche Intelligenz*, pages 393–402, Berlin, Heidelberg, New York, 1993. Springer. Informatik Aktuell.