Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

**Research Report**
TM-93-02

# Conflict Resolving Negotiation for COoperative Schedule Management Agents (COSMA)

**Pierre Sablayrolles, Achim Schupeta**

**May 1993**

**Deutsches Forschungszentrum für Künstliche Intelligenz GmbH**

Postfach 20 80
D-67608 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-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.

Friedrich J. Wendl
Director

# Conflict Resolving Negotiation for COoperative Schedule Management Agents (COSMA)

**Pierre Sablayrolles, Achim Schupeta**

DFKI-TM-93-02

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9104).

# Conflict Resolving Negotiation for COoperative Schedule Management Agents (COSMA)

Pierre SABLAYROLLES, Achim SCHUPETA

DFKI (German Research Center for Artificial Intelligence)
Research group AKA-MOD
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11
GERMANY

email: {sablay, schupeta}@dfki.uni-sb.de

## Abstract

A five component negotiation model which structures the interaction of agents along different levels of increasing complexity is investigated: The lowest level introduces the possible *message types*. The *protocol level* links those messages together to create communicational contexts. The *decision layer* needs some criteria to rank and compare different scheduling proposals. The usage of those criteria implements the *negotiation strategy* of the strategic level. As an outline we sketch how a *cooperation level* emerges within the agent society and enforces cooperative behavior of the agents.

The model is presented at the problem of appointment scheduling, because it is an inherently distributed problem which necessarily involves communication, negotiation, and cooperation mechanisms to resolve possible conflicts. COSMA agents (COoperative Schedule Management Agents) are designed to act as personal assistants to maintain their user's calendar. They are provided with competence to negotiate about the scheduling of appointments with other users. After describing the message types, the negotiation protocol is presented. The decision criteria are derived from a time model based on the association of time intervals and preferences. The negotiation strategies are guided by a local time file, the priority of a meeting and the results of earlier negotiations, which is encoded in a simple partner model maintained by the agents. An example finally shows the overall behavior of the agent society and will reveal and answer several questions about the technical details.

# Contents

# 1 Introduction

Since the development of large software and hardware systems has shown a trend towards distributed systems the metaphor of a **society of cooperating autonomous agents** has become more and more attractive in AI. The basic idea of this approach is to design agents with local plans, goals, knowledge, perception and a conceptual world model and to let them interact in a cooperative manner (cf. e.g. [BG88]). From the engineering perspective the advantages of this approach are similar to those of modular- and object oriented programming. For example we achieve better maintenance and extendibility of systems, robustness, flexibility, splitting of complexity into managable portions, encapsulation of knowledge and its usage, inheritance of general capabilities and so on.

From the scientific perspective the multi-agent approach tries to explore the role of "interactional intelligence" i.e. the aspects of intelligent behavior that emerge from the interaction of individual agents, in contrast to the classical AI approach to design intelligence by means of the representation and usage of knowledge.

The design of cooperating autonomous agents comprises several problems: Which **interaction** may occur between the agents? What style of **communication** is necessary? Which kinds of **conflicts** may arise and how can **negotiation** be designed to resolve those conflicts? Which **planning** paradigm is suited best to coordinate the agent's local plans towards a collectively executed group plan? What **knowledge about the other agerts** (their plans, knowledge and capabilities) must an agent maintain and how is it used within a cooperative problem solving behavior? Which conceptual "**agent architecture**" is necessary to integrate those various capabilities of communication, cooperation, planning, reasoning and partner modelling into a consistent structure?

A lot of Problems! Our goal is to illuminate some aspects of communication, negotiation and conflict resolution within the COSMA project. For the question of an integrative agent architecture we refer to [MP93] where the agent architecture InteRRaP is described, which promises to handle planned rational agent interactions as well as reactive behavior. The concepts used in COSMA are designed to fit naturally into the InteRRaP architecture. In fact InteRRaP is a result of investigation into three multi-agent systems with different application domains:

(1) the implementation of a society of cooperating vehicles in a loading-dock [MP93],
(2) the MARS system, a simulation of cooperating transportation companies [KMM93], and
(3) COSMA, a distributed appointment scheduling manager described in this report.

## 1.1 Appointment Scheduling

Consider the following situation:

*John, an employee of a software company, wants to meet his colleagues Tom and Dan to discuss with them a few details about the interface of a module he is going to implement. The task should be done within two weeks, so he inspects his diary to find that there are only some dates on Thursday next week. He phones Tom and asks for a meeting next week except Thursday. Tom agrees and explains that he is absent from Monday to Wednesday, but Friday would be ok for him. John calls Dan to hear that next week is ok for him except Monday and Friday. Furthermore Thursday is bad, but possible. John takes a closer look at his diary to inspect Thursday again and finds that there is a free slot of half an hour from ten to half past ten. Half an hour would not be sufficient. But there is also a date in the afternoon at two o'clock which is less important and can probably be shifted. He asks Dan if a one hours meeting at two o'clock would be possible. Assuming that Dan agrees, John still has to confirm this date to Tom who might have given his previous statement on the basis of a rough inspection of his calendar or might have put another meeting there in the meantime, assuming than the meeting with Dan and John will be on Friday.*

This short scene describes an instance of the **appointment scheduling problem**. The problem of making appointments with other people is an everyday problem to most of us. Usually an initiator suggests a set of possible time intervals for a meeting to a set of invitees. They in turn have to check their calendars for an interval that fits best and they have to inform the initiator about their result. In most cases, especially if more than two invitees are involved and the set of alternatives is small, it is not possible to generate a mutually accepted appointment date. Hence, the initiator has to resolve the conflict by applying heuristics. Often this is not enough, i.e. it is not only the computation of some numerical data. The initiator has to convince some of the invitees to reschedule their calendar dates to free a certain interval. It is necessary that the participants negotiate to come to a mutually accepted conclusion.

The historic solution to the problem would be a central mediator or a global module which has access to all calendars of the participants. The mediator computes the first possible meeting date or reports a failure if the constrains are too strict. The problem with the centralized solution has many facets:

- The global module is clearly a bottleneck.
- Most calendars are not used with the necessary accurateness.
- A calendar contains private material.
- Marked intervals say nothing about the real participation of the owner.

Hence, the centralized solution is slow, it has acceptance problems by the users, and it does not reflect the natural problem solving process.

Our intention when creating the COSMA system was to investigate the negotiation capabilities of autonomous agents. Therefore we designed COSMA as a personnel assistant of the user that maintains its dairy and has the means and competence to negotiate about the time of meetings. We restict COSMAs functionality to agreement about the time of a meeting to have a well limited negotiation domain.

### 1.2 Multi-Agent Features of Appointment Scheduling

Appointment scheduling is an excellent example how interaction and cooperation of automonous agents leads to mutual committed multi-agent-plans and therefore is a good domain for research in distributed AI. The process of finding a solution involves communication and negotiation about scheduling proposals, mechanisms of cooperation to resolve possible conflicts and a simple model of the other partners. The problem is inherently distributed because each agent maintains its local private calendar and even tries to hide all informations which are not relevant for a specific appointment under discussion. Therefore any centralized approach to this problem would rely on the unnatural assumption that all users are willing to give unlimited access to their private data. Though appointment scheduling is a real world problem, it has furthermore the advantage to be relatively simple to implement because any interaction that occurs is communicational.

Our main interest lies in the development of a negotiation model which supports communication on different levels from simple message sending to different negotiation strategies which lead to different cooperation mechanisms on the social level. Such a model should be as general as possible to use the same design method for other applications. On the other hand we will point out which parts of the model must be specifically tailored for the application. For example in the appointment scheduling problem we had to provide a time model to support the decision layer with a basis for the ranking of different scheduling proposals.

## 2. Related Work

### 2.1 Automatic Distributed Calendar and Appointment System

The Automatic Distributed Calendar and Appointment System was developed as a prototype scheduler to gain experience in the specification of programming of a non-trivial distributed application. The architecture provides a message handling system as a kernel on the system support level which is connected to the calendar agents of the different users. The user accesses his calendar agent through a dialog manager which presents a graphical view of the calendar together with facilities of maintenance. If the user likes to schedule a meeting, he has to fill out a form to specify the appointment profile. The dialog manager passes the information to the calendar agent, which calculates a set of proposals to be send to the calendar agents of the specified participants. Once a proposal for a meeting arrived a calendar agent checks its calendar and if the given interval does not collide with any already scheduled appointments, then it sends a confirmation to the initiator. If there is a collision an assessment value is computed, which is a hint to the attractivity of the proposal. If there is consensus about the appointment, it is fixed by the initiator and the users are informed by their dialog managers. If there is no consensus, then other proposals have to be made if available. If not, the initiator has to apply heuristics to resolve the conflics.

The conflict resolution strategies were supported by four structural features:

(1) The user profile reflects the preferences of the user for making appointments during certain intervals of the day.

(2) The appointment priority reflects the importance of the meeting.

(3) The participant priority reflects how essential a certain person is with respect to the meeting at hand.

(4) The staff priority reflects the hierarchic structure of the organisation.

The conflic resolution has three stages: First, a new set of proposals are computed on the basis of the reported assessment values. Second, already confirmed appointments are rescheduled if the participant priority of a conflicting invitee is higher than the fixed appointment priority. This is only possible if the staff priority of the user is higher than the staff priority of the initiator of the appointment to be rescheduled. Third, participants with low participant priority are removed.

The system was implemented in CSSA (Computing System for Societies of Agents) and it is reported in [MS89].

## 2.2 Distributed Meeting Scheduling (DMS)

Negotiation between meeting scheduling agents were discribed by Sen and Durfee [SD92a], [SD92b]. They basically experimented with the Contract Net Protocol as a negotiation mechanism between the cooperating agents. The focus of their approach was in evaluation of different strategies for announcing, bidding and commiting appointments. They use only simple heuristics like announcing only one time interval versus three alternatives, yes_no answers versus more information bearing answers, and blocking versus non_blocking of time intervals under negotiation. However, under those restricted assumptions it was possible to define a theoretical framework for analyzing different combinations of strategies. In using arguments from probabilistic theory and statistics they were able to prove for instance that if the agents are homogeneous, i.e. they all use exactly the same strategies, there is a saving of communication overhaed and a reduction of the number of negotiation cycles with respect to heterogenious strategies. Moreover they showed formally, that the number of iterations, i.e. negotiation cycles, increase with the density of the appointments in the calendars and with the number of participants. They implemented an experimental testbed and proved their formally predicted behavior in a series of experiments. They not only define a formal framework for the distributed meeting scheduling problem but they also gave a formal definition of the DMS process model they used. The DMS process model is based on finite automata where the different primitives of the Contract Net Protocol are coded into announce states, receive states, generative states, decision states, verification states and success (final) states. We will extend this idea, though not that formally, in the COSMA model.

## 2.3 CAPII - Calendar APprentice

CAPII [Boc93] is another system of negotiating calendar agents develloped at Carnegie Mellon University. The new ideas are to use e-mail facilities for the communication between the agents and to adopt dialog based learning techniques to improve the agents behavior. The protocol for the communication is based on an extended Contract Net Protocol and it is implemented as a finite automata. The CAP agent uses an information format

(DATE, TIME, DURATION, LOCATION, ATTENDEES, TOPICS)

together with predefined natural sentence fragments to announce meetings. The special key words (propose, bid, confirm,....) and information given by the mail system, e.g. reply of your mail # from * with subject $, are used to follow the negotiation protocol. Users without a CAP agent at hand can also take part in the negotiation process. They can easily follow the negotiation process, because the message contents are natural language like. They can be active, too, if they follow some guidelines, e.g. answering an announce by starting with YES, NOT-THEN, MAYBE or NO. However, it was intented that the human user may answer as he likes, without knowing any of the key words. The learning mechanism then should provide the agents with the ability to identify synonyms or special styles of the human users. The ideas of CAP are currently applied to a Room APprentice (RAP) for the automatic handling of room reservations.

## 2.4 Group Scheduling

The distributed calendar idea was used by Lux [Lux92] to study his cooperation model as part of the general body-head-mouth agent model. The special idea is to define different kinds of agents by using different kinds of calendar systems for the application dependend body of the agent model. The heads of the agents use the same application independend cooperation primitives to be composed to cooperation methods, e.g. master-slave behavior or the Contract Net Protocol. Cooperation primitives consist of message types (PROPOSE, ACCEPT, REFINE, REJECT, MODIFY, TELL, REQUEST, ORDER), cooperation objects (appointments in this application) and reply constraints (expected_replies, deadline, etc). The initiator of an appointment proposal collects all answers of the invitees and tries to find a minimal conflicting time interval. If there are dramatic conflicts and if he is in a mandatory position, he has the possibility to order a rescheduling of other appointments. If the relationship is a peer-to-peer one then the user is asked to give a new time interval for the appointment. The system is implemented together with a graphical appointment-manager user interface where various appointment relevant options and parameters can be specified.

### 2.5 MADMAN - Multi-Agent Diary Manager

In the MADMAN scenario [EE92] a set of SADMANs (Single Agent Diary Managers) are responsible for the scheduling of meetings for their users. The most important idea is the use of graphically represented user profiles that show the users' preferences for scheduling meetings during the working time intervals. After a proposal the initiator collects the invitees profiles for the proposed period and computes a group activity profile for that interval. If a common period with the necessary lengh and a value above a certain threshold can be computed, then the meeting will be fixed by the initiator. If not alternative time proposals have to be suggested by the initiator.

## 3 The Negotiation Ingredients

The negotiation process in COSMA consists of five main ingredients. The **message** format has to be specified together with the message types that possibly occur during negotiation. Using these messages, the **protocol** declares which possible sequences of messages are allowed and senseful. We say that the communicational context determines via the protocol which messages are possible next. The protocol still leaves open the decisions of an actual negotiation (e.g. accept or reject a proposal). Therefore beside the messages and protocol we need **valuation** mechanisms which provides functions for ranking and comparing different scheduling proposals. The usage of those rankings at certain decision points defined by the protocol is what we call the **negotiation** strategy. The effect of a negotiation strategy concerning the whole agent society defines the abstract **cooperation** behavior**.** Figure 1 shows the interplay of these components that will be described in the following sections in more detail.

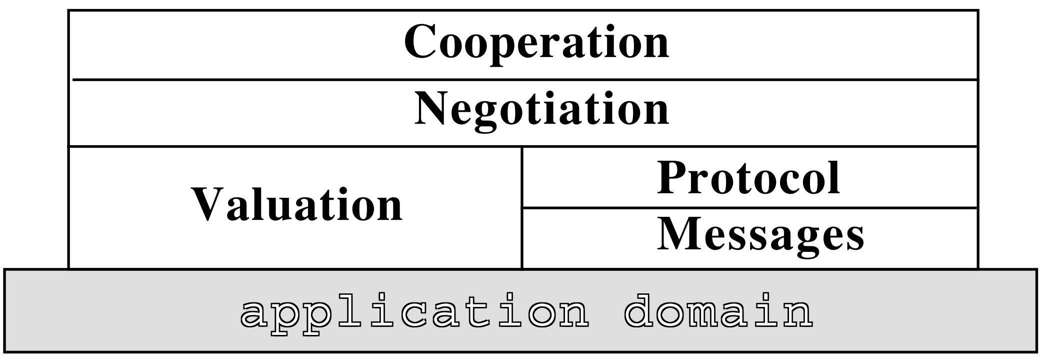


*Figure 1: Interdependencies of COSMA's negotiation ingredients*

## 3.1 Message types

According to the central idea of speech act theory (cf. e.g. [App85]) we view communicational actions as planned actions. Planning and communication are tightly entangled: Plans contain communication acts and communiucational acts influence the agent's plans. For COSMA all planned actions are communicational acts and therefore we don't have any problems with the integration of planning and communication here.

The following list presents all message types that are possible in COSMA-negotiations[1] and scetches their meaning:

| | |
|---|---|
| ARRANGE: | To initialize a negotiation an agent sends this message with a specification of the priority, the participants and a suggested time-interval to all participants. |
| REFINE: | As an answer to a proposal this message specifies a subinterval of the proposed time-interval. |
| MODIFY: | As an answer to a proposal this message rejects the proposed time interval and suggests an alternative interval. |
| CHANGE: | This message cancels the date of an already arranged appointment and makes a proposal to fix it into another interval. It reinitializes an already finished negotiation. |
| ACCEPT: | Agreement to a proposed time-interval. |
| CONFIRM: | This message finishes the negotiation and fixes a specific time for a meeting when an agreement of all participants is reached. |
| REJECT: | Irrevocable refusal of an appointment. This message finishes the negotiation when it is impossible to reach an agreement. |
| CANCEL: | This message cancels an already arranged appointment. There is no alternative date for the meeting specified because the sender refuses to be a participant in that meeting at all. (The Initiator of the meeting has to decide if he is going to arrange a modified meeting with a modified list of participants.) |
| NEGOTIATE: | This message is dedicated to conflict resolution. When the participants have refined a proposed time-interval in a mutually incompatible way, the initiator tries to negotiate about the different possible subintervals to change the others mind (i.e. ranking of possible scheduling decisions). This message provides information exchange for this negotiation. |
| COND-ACCEPT: | This message is also special for conflict resolution. An agent may conditionally agree to a time proposal depending on the ranking that other agents gave to that proposal. But if he has no knowledge about their ranking, he can reply with a decision threshold in the |

[1]Additionally there are some technical message types between the COSMA and the graphical user interface which are not relevant for the description of COSMA's negotiation capabilities.

cond_accept message. The initiator uses that threshold to evaluate the decision when the knowledge about the rankings is complete.

The last two message types are very specific for our way of conflict resolution by negotiation and their usage will become much clearer in the following chapters.

The structure of all these message types is defined homogeneously. It specifies

- a message-identification (a unique number),
- the sender of the message,
- the recipient list,
- the message-type,
- an appointment-identification (a unique number),
- the list of participants, and
- a proposed time.

An example of that structure is shown in section 4. The "proposed time" is a data structure that specifies the start- and end-point of an interval together with a preference structure, (i.e. preferences are associated with subintervals), the duration and the priority of the meeting. Additional information slots can be provided depending on the specific message type.

## 3.2 The Negotiation Protocol

The protocol specifies the possible communicational interactions between the agents. We think of a protocol as a set of possible actions within a given communicational context. E.g. an answer is usually given in the context of a question.

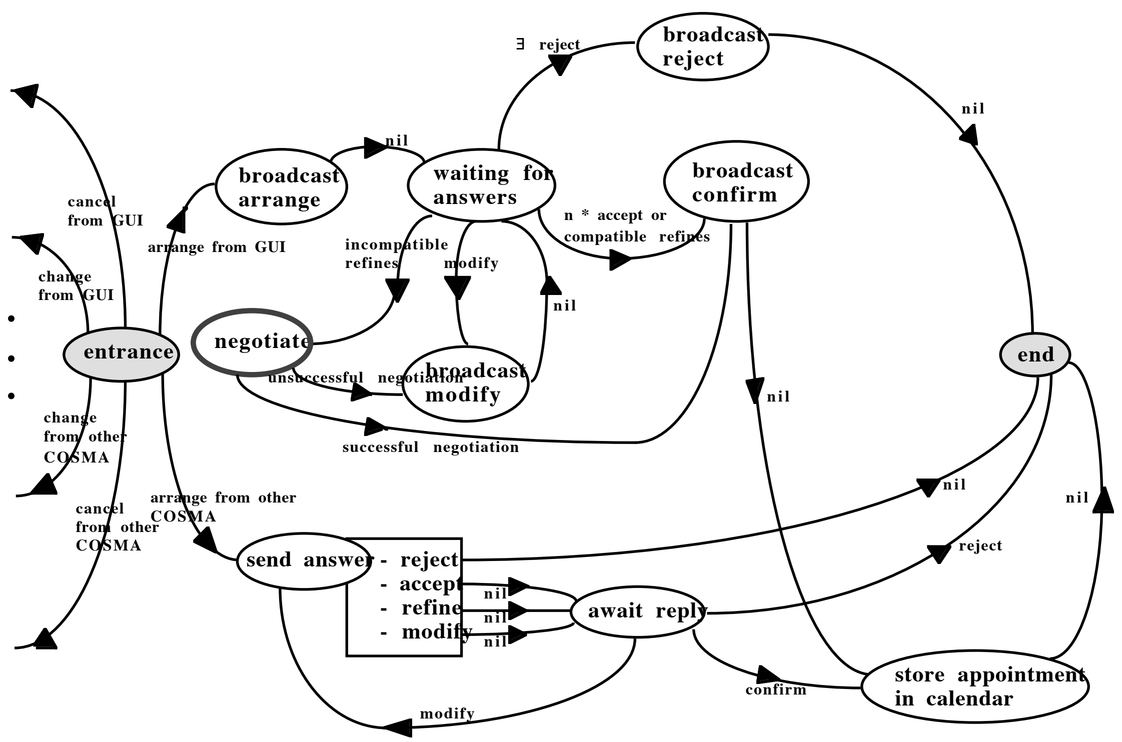


*Figure 2: Representation of the protocol as graph*

We represent the COSMA protocol as a state automaton, where the states represent the (communicational) actions of an agent and the transitions represent the received messages of other agents. Therefore the protocol is represented from the local view of an agent. The reception of an unexpected message will lead to the activation of special error recovery mechanisms not shown in the protocol graph. Figure 2 shows the important part of the protocol.

The basic idea is simple: A negotiation starts with the arrange-message of the initiator, which specifies the priority of the meeting, the list of participants and a first time proposal.

The COSMA agent receives this message in the initial state "entrance" from the graphical user interface (GUI) and broadcasts it to all participants. It awaits the answers, which could fall into four categories:

- At least one of the participants rejects. In this case the agent has to reject the meeting to all other participants, which terminates the negotiation. (We leave it to the user to decide, if he wants to rearrange that meeting with a modified list of participants.)
- Some participants reply with the suggestion to modify the time proposal. In this case the COSMA agent has to broadcast a modified time proposal to all participants, which it has to compute based on local preferences and the modify-proposals.
- All participants accept the proposal or refine the proposed time in such a way, that the different refinements overlap in one acceptable time-interval. The COSMA agent has to fix a specific time point for the begining of the meeting within that interval and has to broadcast this with a confirm-message to the participants.
- All participants agree in principle to the proposed time interval, but refine this in a mutually incompatible way, i.e. there is no sufficient overlap for the different refinements. This case could principally be handled like the second one (propose some other interval) but our protocol is designed to negotiate about the different subintervals of the proposal to reveal the criteria on which the agent's replies are based.

The state "negotiate" in the above graph represents a subprotocol which again could be represented as a graph. In this subnegotiation the initiator splits the interval under discussion into possible subintervals. It orders these subintervals according to its local preferences and sequentially works on this list of proposals. For each entry it asks the other agents to evaluate their ranking of this proposal based on the now partially available rankings of other participants for that interval. This is done by using the message type negotiate. The answers to this message are of type reject or accept or cond-accept. The details of this subnegotiation will become clear in the discussion of the ranking criteria and in the example of chapter 4.

Note that the protocol captures the notion of a context in a negotiation. It specifies senseful sequences of messages. It also contains an implicit model of the other agents by the assumption that they follow the same protocol in a negotiation. From this point of view it is also a coordinator of interaction. The nesting of subprotocols within protocols is a promising feature to structure protocols according to different abstraction levels. In the following section we will describe the criteria that are provided to take those decisions the protocol leaves open.

## 3.3 Valuation Criteria

The valuation level is based on a time model which associates preferences to different time intervals. The two ranking criteria provided are a **utility** that is assigned to a specific scheduling decision and a degree of **concession**. The utility is a numerical abstraction of the willingness to accept the scheduling decision. It is also used to compare different alternatives, e.g. in the decision to displace another meeting or not. The degree of concession was introduced to tune the negotiation with an additional parameter that modifies the negotiation decisions by the introduction of knowledge about previous negotiations. We will first describe our time model and the computation of the utility and then the computation of the degree of concession in the following.

### 3.3.1 Priorities, Preferences and Utility

We represent timepoints in COSMA as quadruples specifying the year, the day, the hour and the minute of a time:

Definition: T is the set of all time-points

$$T=\{(Y,D,H,M) \mid Y \in Nat, D \in \{1,..,365\}, H \in \{0,..,23\}, M \in \{0,..,59\}\}$$

Intervals are build as usual:

Definition: I is the set of all intervals

$$I = \{(Sp, Ep) \mid Sp \in T, Ep \in T, Sp < Ep\}$$

,where '<' denotes the canonical precedence of timepoints. We associate a preference to intervals to build preferenced intervals:

Definition: PI is the set of preferenced intervals

$$PI = \{(i, p) \mid i \in I, p \in [0, 1]\}$$

If the users assigns some preference p to a time interval i we intend the following meaning with this: There is some activity planned for that interval to which the user assigns the relative importance p, where p = 1 denotes maximal importance and p = 0 denotes a totally unimportant activity. All other values express degrees of importance in between. We combine sequences of preferenced intervals into time files that will reflect a users time availibility over a longer range of time:

Definition: A time file is a sequence of contiguous preferenced intervals

$$TF = \{(p_{i1},..., p_{in}) \mid p_{ik} \in PI, 1 \le k < n: Ep_k = Sp_{k+1} \}$$

Let M be the set of meetings, then the function prio: M -> [0,1] gives the priority of a meeting and dur: M-> Nat gives the duration of the meeting in minutes. Thereby the priority of a meeting expresses the importance of the meeting and can be directly compared with the preferences of times as explained above. This is in fact the idea to compute the utility as ranking measurement for a scheduling decision. The function pref: $T \times T \times TF \rightarrow R^+$ computes the value of the integral of preference[2] from one timepoint to another for a given time file. With this preliminaries we could define the utility of the decision to schedule meeting m at timepoint t for a given time file to be:

$$ut: M \times TF \times T \rightarrow [0,1]$$

$$ut(m, tf, t) := \frac{prio(m) * \Box dur(m) \Box - \Box pref(t, \Box t \Box + \Box dur(m), \Box tf)}{\Box dur(m)}$$

The priority of the meeting is specified by the user, that initially creates the meeting within a dialog with its COSMA. The time file is maintained by COSMA in refering to four different sources:

- Common knowledge about preferences of time like e.g. most people do not work at night or will have a very low preference for working during the lunch time.
- User specific preferences express the users personnel preferences to allocate meetings and represent all time consuming activities, that have nothing to do with meetings.
- Calendar dates represent the allocations of time for already scheduled meetings. Usually the preferences of these allocations are the priorities of the corresponding meetings. Additionally by increasing the priority of a scheduled meeting we could express the inconvenience to reschedule it when a time slot for another meeting with higher priority is searched.
- The action memory stores intervals that are involved in an open negotiation. This is useful to control the interaction between two or more negotiations running in parallel.

Figure 3 shows how these sources are combined by taking the maximum or the average to compute the time file.

### 3.3.2 Degree of Concession

The concession depends on the utilities of past negotiations and is used to guide the subnegotiation about a specific interval. The agents have to maintain a simple partner model to be aware of what has happened in past negotiations. An agent that has gained relatively much utility out of the meetings schould make more concessions in future negotiations, i.e. should agree to proposals even if the utility-ranking is not very good. This mechanism should balance the utility fair to the agents over a history of negotiations. The degree of concession is used in the negotiation and updated for the usage in future negotiations. An agent has to maintain for each partner X it ever met in a negotiation two

[2]You could view a time file as a stepwise constant function.

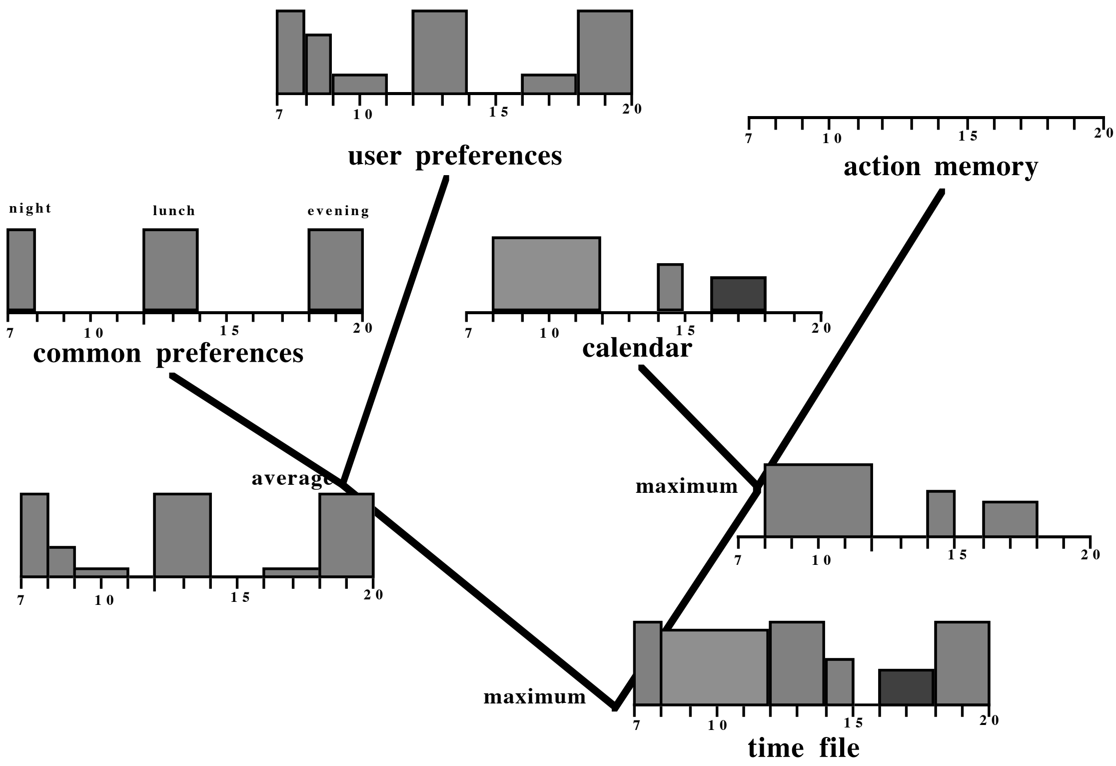


*Figure 3: Sources for the computation of the time file*

values: X's utility $U_x$ and its own utility $U_{o,x}$ in negotiations with X. After each negotiation where X is also a participant, these values are updated by averaging:

$$U_x \leftarrow \frac{U_x + ut_x}{2} \quad ; \quad U_{o,x} \leftarrow \frac{U_{o,x} + ut_o}{2}$$

Thereby $ut_x$ stands for the utility of X and $ut_o$ for the own utility in the current negotiation. With this simple mechanism of averaging we do not have to store the whole history of negotiations and besides gain the effect that more recent utilities are weighted higher than elder ones. Other weightings as 1/2 for the old and the actual value could be introduced easily.

For a decision within the actual negotiation the degree of concession is computed as the difference from the average utility and the agent's own utility. The following example assumes that there are three other agents, X, Y and Z in the list of participants. The degree of concession C is then computed in the following way:

$$C = U_{av} - U_o \quad \text{where}$$

$$U_{av} = \frac{U_o + U_x + U_y + U_z}{4} \quad \text{and} \quad U_o = \frac{U_{o,x} + U_{o,y} + U_{o,z}}{3}$$

The usage of C and the effect of this usage will be described in the following section:

## 3.4 Decision Critera

The negotiation level is build directly above the protocol and the valuation criteria. The protocol determines the communicational context for a decision. Valuation provides the ranking criteria for a decision. The negotiaition level now defines a negotiation strategy in the way the ranking criteria are used for a decision. In general we assume that the agent tries to maximize the utility of all its meetings. In the context of a proposal in the top-level protocol an agent decides purely on the existence of a subinterval with positive utility. For the subnegotiation we have run experiments with three different strategies:

| | |
|---|---|
| Absolute decision: | The agent agrees to a scheduling proposal if its utility is positive. |
| Relative decision: | The agent agrees to a scheduling proposal if its utility is greater than the average of the utility of all other participants of the meeting. |
| History involving decision: | The agent agrees to a proposal if its utility reduced by the degree of concession is greater than the average of the utility of all other participants of the meeting. |

Especially the history involving strategy has the effect that dates are packed more densely. On the other hand there is more communicational overhead with this strategy because it is more likely that already fixed appointments must be rescheduled. The relative decision is of course very uncooperative and if run strictly by all agents will never result in a conflict resolution within the subnegotiation, because it is obviously not possible for all agents to gain more utility than the average.

## 3.5 Cooperation Mechanisms

The cooperation level is a kind of meta-level because it evolves from the local interactive behavior of the agents. To predict the global behavior of a distributed system with knowledge about the local behaviour of the agents in the system is one of the most urgent open problems in DAI.

The cooperation in COSMA evolves from the interaction according to the described negotiation model and results in the distributed local calendars. These are globally consistent in that sense that all participants of successful negotiated meetings will have an entry for that meeting in their calendar at the same time interval. Furthermore with the degree of concession and its usage, we intended to model a mechanism of equally distributed utility that the agents gain from meetings. This resembles a typical human behavior. When one person makes concessions in a negotiation such that a compromise can be found, then the next time it should be someone other's turn to be placable. In our simulations we observe this effect. But we also noticed that the degree of concession has a strong convergence towards zero. This degenerates the history involving strategy towards the relative decision strategy, which is very uncooperative. So for a further development of the negotiation strategy it would be useful to find a mechanism to disturb the balance of concessions from time to time a little bit.

# 4 Example

In the following section we will trace an example of a COSMA-negotiation which involves three participants **A**, **B** and **C**. It uses the history involving negotiation strategy and will give an impression how the COSMA agents interact. After the initial arrange messages are send, we will see how two incompatible refine messages lead to a specific negotiation about the subintervals of the proposal. These inspections of subintervals fail three times, but the trial for the fourth subinterval is successful. The negotiation is finished with a confirm message.

Initially user **A** specifies a meeting with priority 0.7, with a duration of 1 hour and with participants **A**, **B** and **C** to be scheduled at monday next week. The graphical user interface (GUI) of **A**'s COSMA gives these data to the planning-module of COSMA.

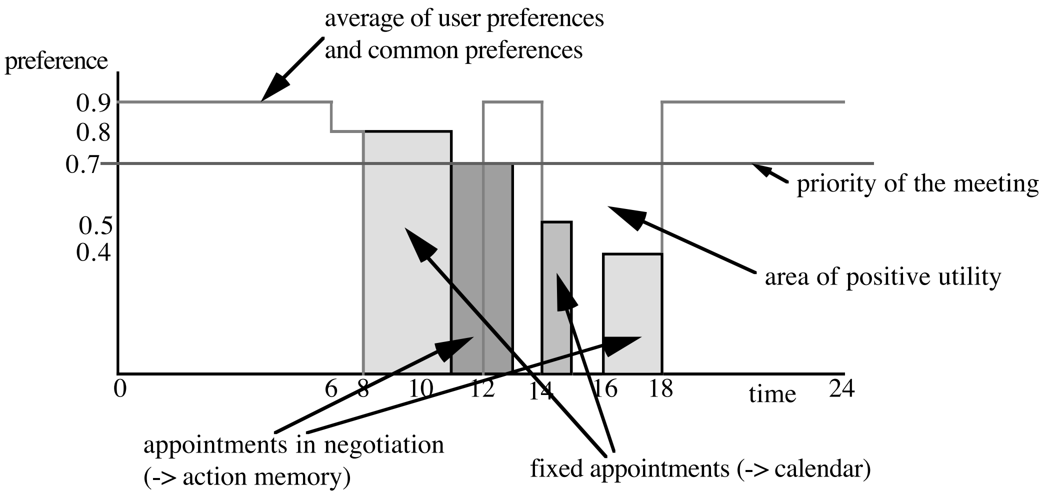


*Figure 4: **A**'s time file for monday*

The planning-module computes the time file for monday using the user's preferences, the common preferences, the calendar data and data from the action-memory to obtain the time file (cf. figure 3). Let us assume it looks like shown in figure 4.

The subinterval from 14.00 to 18.00 o'clock is possible for **A** because the priority of the meeting exceeds the time preferences in the time file. **A** now sends an arrange message to **B** and **C** which proposes this interval for the meeting. If we assume **B**'s and **C**'s time files to look like shown in figure 5 and 6, then **B** as well as **C** will reply with a refine message. **B** suggests the subinterval from 14.00 to 15.00 while **C** would like to schedule the meeting between 15.00 and 17.00.

Up to this stage of negotiation the agents still hide as much as possible of their private calendar data. Therefore in the refine message e.g. **C** only specifies the interval from

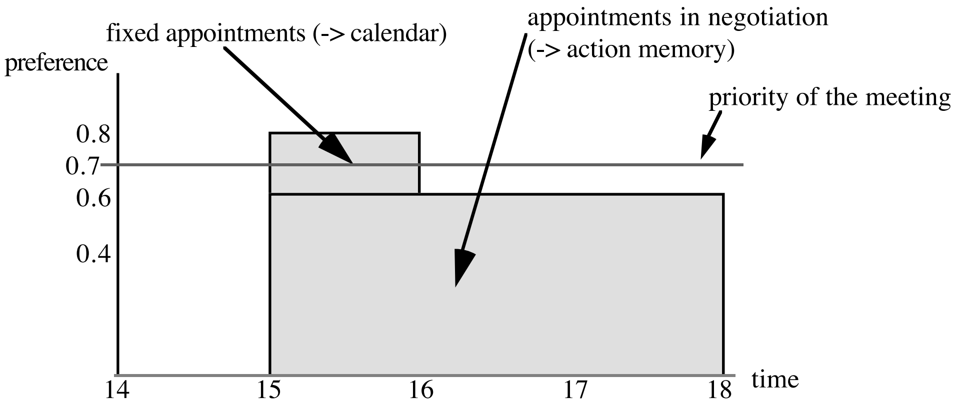


*Figure 5: **B**'s time file for monday afternoon*

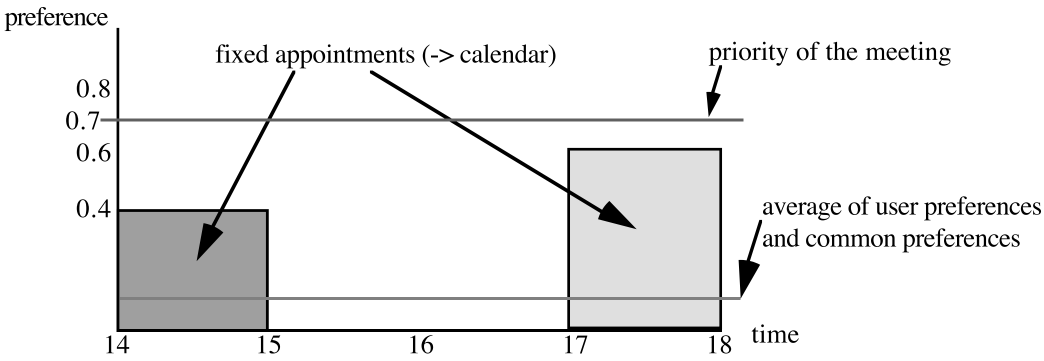


*Figure 6: **C**'s time file for monday afternoon*

15.00 to 17.00 as possible and claims the subintervals before and after this as impossible though there would be a positive utility for **C** even at these subintervals. To give an impression of the messages we list the refine message of **C**:

```
((MESSAGE-ID . 224) (IN-REPLY-TO . 92) (FROM . "C@DFKI.UNI-SB.DE")
 (TO . "A@DFKI.UNI-SB.DE") (ACTION . REFINE)
 (ID-APP . "1037 A@DFKI.UNI-SB.DE")
 (PARTICIPANTS "A@DFKI.UNI-SB.DE" "B@DFKI.UNI-SB.DE" "C@DFKI.UNI-
SB.DE")
 (PROPOSED-TPF
  (START-POINT (YEAR . 1993) (DAY . 88) (HOUR . 14) (MINUTE . 0))
  (END-POINT (YEAR . 1993) (DAY . 88) (HOUR . 18) (MINUTE . 0))
  (PRIORITY . 0.7)
  (DURATION (HOUR . 1) (MINUTE . 0)))
  (PRIORITY-FILE
   ((START-POINT (YEAR . 1993) (DAY . 88) (HOUR . 14) (MINUTE . 0))
    (END-POINT (YEAR . 1993) (DAY . 88) (HOUR . 15) (MINUTE . 0))
    (PRIORITY . 1))
   ((START-POINT (YEAR . 1993) (DAY . 88) (HOUR . 15) (MINUTE . 0))
    (END-POINT (YEAR . 1993) (DAY . 88) (HOUR . 17) (MINUTE . 0))
    (PRIORITY . 0.1))
   ((START-POINT (YEAR . 1993) (DAY . 88) (HOUR . 17) (MINUTE . 0))
    (END-POINT (YEAR . 1993) (DAY . 88) (HOUR . 18) (MINUTE . 0))
    (PRIORITY . 1))))
```

The slot PROPOSED-TPF contains the start- and endpoint of the interval under discussion, the priority, the list of participants, the duration of the meeting and the PRIORITY-FILE which specifies the preferences of the sender for the different subintervals. In this case **C** does not give the full information about its time file, but wants to express that only the subinterval from 15.00 to 17.00 is acceptable.

From the protocol-point of view, **A** now enters the state "negotiate" (cf. figure 3). In this context, **A** orders all possible subintervals according to its own preferences, which results in the following list:

(15.00 - 16.00; 16.00 - 17.00; 17.00 - 18.00; 14.00 - 15.00)

16.00 - 17.00 is preferred to 17.00 - 18.00 because it is member of at least one of the refining-proposals. **A** starts to work on this list sequentially: First it sends a negotiate message about 15.00 - 16.00 to **B**. It contains **A**'s and **C**'s preferences for that file which are 0 and 0.1 respectively. Note that **A** has obtained **C**'s preference in the refine message shown above. According to the formula given in section 3.3.1, **B**'s utility for that proposal is -0.1. Using the absolute decision criterion (i.e. accept only positive utility-proposals) **B** would have to reject this proposal immediately. In our case using the history involving decision, we must take a closer look to **B**'s partner model:

Let us assume the following actual partner model for **B** from which **B** computes the degree of concession:

previous utilities:

| | of partner | of B |
|---|---|---|
| A | 0.5 | 0.3 |
| C | 0.7 | 0.3 |
| D | 0.4 | 0.6 |
| E | 0.3 | ... |
| ... | ... | |

According to the formulas from section 3.3.2, **B** obtains 0.2 as degree of concession. **B** computes the average utility ( (-0.1 + 0.7 + 0.6) / 3 = 0.4) and compares it with its utility reduced by the degree of concession:

-0.1 - 0.2 > 0.4 ? -> no!

This condition is not true, therefore **B** sends a reject message to **A**. **A** takes the next subinterval from its lists and sends a negotiate about 16.00 - 17.00 to **B**. Nearly the same as before happens, only that this time **B** has to test the condition:

0.1 - 0.2 > 0.333 ? -> no!

Again **B** rejects this proposal. Note that this time there would be a difference in using the absolute decision criterion: **B** would accept this proposal due to the positive utility and the interval 16.00 - 17.00 would be confirmed for the meeting. In our situation nevertheless **A** now tries to negotiate about the interval 17.00 - 18.00. This time **A** has to negotiate with both, **C** and **B** and cannot provide knowledge about the other's preferences in that messages. Therefore **B** and **C** have problems in computing the average utility and cannot make a definitive decision. **B** has to decide upon the condition:

$$0.1 - 0.2 > \text{average ?}$$

This inequality can only be true for a negative value of average. We assume a negative average of utility for a meeting not acceptable for the agent society and therefore **B** can decide to reject without knowing the exact value of the average-utility in this special case. **C** has to decide upon the condition:

$$0.1 + 0.2 > \text{Average ?}$$

(We assume that the degree of concession for **C** in this example is - 0.2). **C** replies with a cond_accept message which contains the value of 0.3 and has the meaning: "I would accept this proposal if the value of the average of all participant's utility does not exceed 0.3." The message of course further contains **C**'s utility (0.1 in this example) such that the agent **A** is able to evaluate the condition if all other participants accept or conditionally accept the proposal. In our case **B** already rejected this proposal so that **A** finally has to try the last entry in its list which is 14.00 - 15.00.

This time **A** already knows **B**'s acceptance (from the refine message in the beginning) and has to ask only **C** with a negotiate message. **C** has to evaluate the condition:

$$0.3 + 0.2 > 0.4$$

which is true. **C** sends an accept message and **A** finishes the negotiation by sending a confirm to both **C** and **B**. This also contains the utilities of the participants such that they are able to update their partner model accordingly. For the example case of **B** we obtain the following update:

previous utilities:

| | of partner | of B |
|---|---|---|
| A | (0.5 + 0.2) /2 | (0.3 + 0.7) /2 |
| C | (0.7 + 0.3) /2 | (0.3 + 0.7) /2 |
| D | 0.4 | 0.6 |
| E | 0.3 | ... |
| ... | ... | |

This negotiation agreement further means that there are two other activities that has to be rescheduled, namely the activity during 14.00 - 15.00 with priority 0.4 from **C**'s calendar and the activity during the same time with priority 0.5 from **A**'s calendar. **C** and **A** have to send change-messages to the corresponding participants. This will initiate or to be more exact reinitiate the negotiations about these meetings. Because of the dynamic changes of information in this distributed system we do not assume that these negotiations can rely on any informations from the first negotiations about these meetings. The change (and also cancel) of a meeting also does not retract the effect that the previous successful negotiation about that meeting had have on the partner model. Figure 7 finally sketches the exchange of messages of the described example.

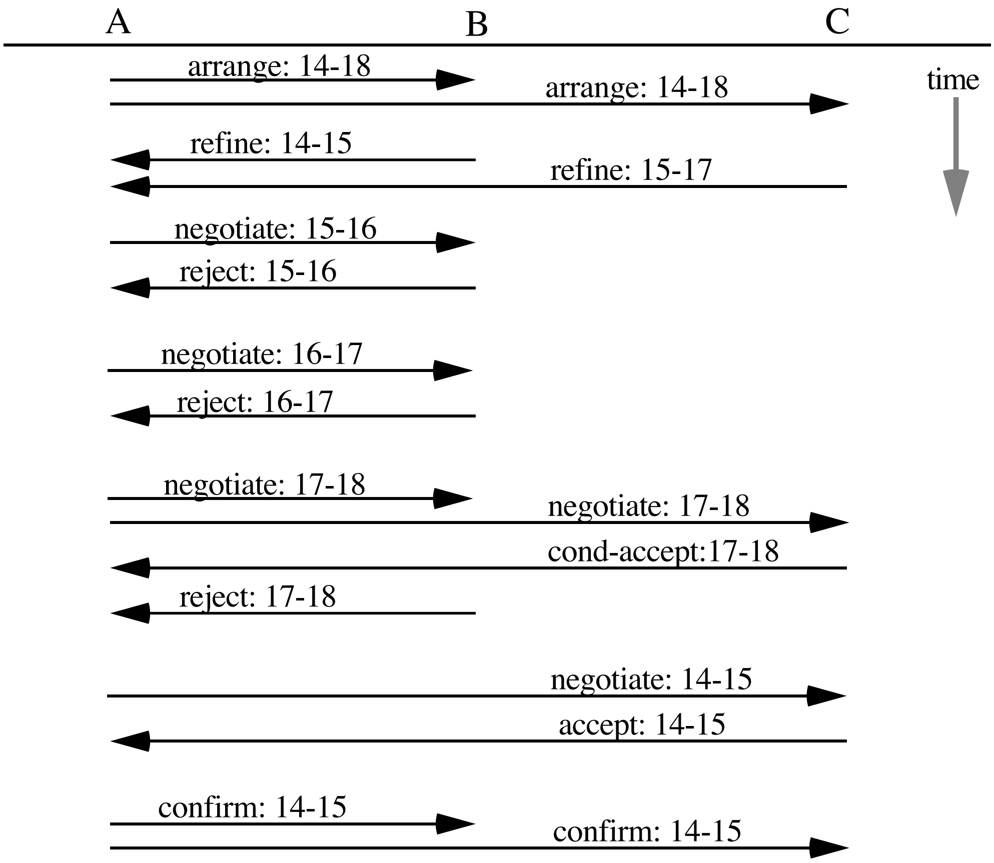


*Figure 7: Exchange of messages in our example*

# 5 Conclusion

We have motivated the appointment scheduling problem to be a good candidate for the study of communicational interaction between autonomous agents. After a short review of other approaches to solve the appointment scheduling problem in a distributed manner, we described the COSMA system. The design of the negotiation about dates of meetings involved several aspects: Starting from the selection of an appropriate set of messages and the specification of meaningful sequences of messages via a protocol, then the development of a time model that supports ranking and comparison of proposals to guide the decisions to be made when running a specific negotiation strategy. The effect of the local behavior of the agents within a negotiation was a cooperative allocation of appointments within the calendars of the participants. Finally we exemplified a specific negotiation example by tracing it in detail.

All experiments we tried succeeded in finally finding an acceptable time slot except when one of the participants rejects or cancels the meeting. This is no surprise, because our protocol is designed in a way to produce theoretically an infite sequence of proposals that place the meeting more and more in the far future. There is still no deadline for a meeting. Also we did not give criteria that quantitatively describe the quality of one scheduling solution compared to another. Our intention in this study was not to design a distributed algorithm which optimally solves the scheduling appointment problem, but to investigate into the negotiation and the special case of conflict resolution. The frame devoloped to structure the interdependencies of the different aspects of the negotiation might be useful for the design of systems in other application domains, while it is clear that the specific contents of the negotiation components as they have just been described are of limited value.

To preserve as much as possible of this, for future work we think of a generalization to more general scheduling problems. Crucial for that goal seems to be the introduction of deadlines and of some notion of resources (e.g. the rooms where the meetings take place.)

ACKNOWLEDGEMENTS:

We like to thank Jürgen Müller for guidance, various discussions and helpful support in exploring related work. Also the cooperation with the DISCO project group in an early stage of this work has been very fruitful.

This Project is sponsored by the German Ministry for Research and Technology under grant FKZ ITW-9104.

# 6 References

App85 D.E. Appelt: Planning English Sentences, Cambridge University Press, New York, 1985

BG88 A. Bond, L. Gasser: Readings in Distributed AI, Morgan Kaufmann, Los Angeles, 1988

Boc93 S. Bocionek: CAP II & RAP: Lernfähige, verhandelnde Agenten für die Büroautomatisierung, in "Beiträge zum Gründungsworkshop der Fachgruppe Verteilte Künstliche Intelligenz, Saarbrücken, 29.-30. April 93", DFKI Document D-93-06

EE92 N. Eisinger, N. Elshiewy: MADMAN- Multi-Agent Diary Manager, ECRC-92-7i internal report of the ECRC, München, March 92

KMM93 N. Kuhn, H.J. Müller, and J.P. Müller: Simulating cooperative transportation companies, Proceedings of the European Simulation Multiconference (ESM-93), Lyon, France, June 1993. Society for Computer Simulation.

Lux92 A. Lux: A Multi-Agent Approach towards Group Scheduling, RR-92-41, Research Report of the DFKI, August 92

MP93 J.P. Müller and M. Pischel: The Agent Architecture InteRRaP and its Application in the Loading Dock Domain, Technical Report Research Report RR-93-26, German Artificial Intelligence Research Center (DFKI), Saarbrücken, 1993

MS89 F. Mattern, P. Sturm: An Automatic Distributed Calendar and Appointment System, Proc. of the EUROMICRO Conference 1989, North-Holland

SD92a S. Sen, E. H. Durfee: Automated Meeting Scheduling among Heterogeneous Agents, in Workshop Notes: Cooperation Among Heterogeneous Intelligent Systems, AAAI-92, San Jose Convention Center, California, July 12 - 16, 92

SD92b S. Sen, E. H. Durfee: A Formal Analysis of Communication and Commitment in Distributed Meeting Scheduling, 11th Internat. Workshop on DAI, Homestead, Glen Arbor, MI, 25 - 29 Feb. 92

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

## DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

## DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or via anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

### DFKI Research Reports

**RR-93-10**
*Martin Buchheit, Francesco M. Donini, Andrea Schaerf:* Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

**RR-93-11**
*Bernhard Nebel, Hans-Jürgen Bürckert:*
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

**RR-93-12**
*Pierre Sablayrolles:* A Two-Level Semantics for French Expressions of Motion
51 pages

**RR-93-13**
*Franz Baader, Karl Schlechta:*
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

**RR-93-14**
*Joachim Niehren, Andreas Podelski, Ralf Treinen:* Equational and Membership Constraints for Infinite Trees
33 pages

**RR-93-15**
*Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster:* PLUS - Plan-based User Support Final Project Report
33 pages

**RR-93-16**
*Gert Smolka, Martin Henz, Jörg Würtz:* Object-Oriented Concurrent Constraint Programming in Oz
17 pages

**RR-93-17**
*Rolf Backofen:*
Regular Path Expressions in Feature Logic
37 pages

**RR-93-18**
*Klaus Schild:* Terminological Cycles and the Propositional $\mu$-Calculus
32 pages

**RR-93-20**
*Franz Baader, Bernhard Hollunder:*
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

**RR-93-22**
*Manfred Meyer, Jörg Müller:*
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

**RR-93-23**
*Andreas Dengel, Ottmar Lutzy:*
Comparative Study of Connectionist Simulators
20 pages

**RR-93-24**
*Rainer Hoch, Andreas Dengel:*
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

**RR-93-25**
*Klaus Fischer, Norbert Kuhn:* A DAI Approach to Modeling the Transportation Domain
93 pages

**RR-93-26**
*Jörg P. Müller, Markus Pischel:* The Agent Architecture InteRRaP: Concept and Application
99 pages

**RR-93-27**
*Hans-Ulrich Krieger:*
Derivation Without Lexical Rules
33 pages

**RR-93-28**
*Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker:* Feature-Based Allomorphy
8 pages

**RR-93-29**
*Armin Laux:* Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

**RR-93-30**
*Stephen P. Spackman, Elizabeth A. Hinkelman:*
Corporate Agents
14 pages

**RR-93-31**
*Elizabeth A. Hinkelman, Stephen P. Spackman:*
Abductive Speech Act Recognition, Corporate Agents and the COSMA System
34 pages

**RR-93-32**
*David R. Traum, Elizabeth A. Hinkelman:*
Conversation Acts in Task-Oriented Spoken Dialogue
28 pages

**RR-93-33**
*Bernhard Nebel, Jana Koehler:*
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

**RR-93-34**
*Wolfgang Wahlster:*
Verbmobil Translation of Face-To-Face Dialogs
10 pages

**RR-93-35**
*Harold Boley, François Bry, Ulrich Geske (Eds.):* Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
**Note:** This document is available only for a nominal charge of 25 DM (or 15 US-$).

**RR-93-36**
*Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt:* Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

**RR-93-38**
*Stephan Baumann:* Document Recognition of Printed Scores and Transformation into MIDI
24 pages

**RR-93-40**
*Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf:*
Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

**RR-93-41**
*Winfried H. Graf:* LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

**RR-93-42**
*Hubert Comon, Ralf Treinen:*
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

**RR-93-43**
*M. Bauer, G. Paul:* Logic-based Plan Recognition for Intelligent Help Systems
15 pages

**RR-93-44**
*Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt:* Subsumption between Queries to Object-Oriented Databases
36 pages

**RR-93-45**
*Rainer Hoch:* On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

**RR-93-46**
*Philipp Hanschke:* A Declarative Integration of Terminological, Constraint-based, Data-driven, and Goal-directed Reasoning
81 pages

**RR-93-48**
*Franz Baader, Martin Buchheit, Bernhard Hollunder:* Cardinality Restrictions on Concepts
20 pages

**RR-94-01**
*Elisabeth André, Thomas Rist:*
Multimedia Presentations:
The Support of Passive and Active Viewing
15 pages

**RR-94-02**
*Elisabeth André, Thomas Rist:*
Von Textgeneratoren zu Intellimedia-Präsentationssystemen
22 Seiten

**RR-94-03**
*Gert Smolka:*
A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards
34 pages

**RR-94-05**
*Franz Schmalhofer, J.Stuart Aitken, Lyle E. Bourne jr.:*
Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse
81 pages

**RR-94-06**
*Dietmar Dengler:*
An Adaptive Deductive Planning System
17 pages

**RR-94-07**
*Harold Boley:* Finite Domains and Exclusions as First-Class Citizens
25 pages

**RR-94-08**
*Otto Kühn, Björn Höfling:* Conserving Corporate Knowledge for Crankshaft Design
17 pages

**RR-94-10**
*Knut Hinkelmann, Helge Hintze:*
Computing Cost Estimates for Proof Strategies
22 pages

**RR-94-11**
*Knut Hinkelmann:* A Consequence Finding Approach for Feature Recognition in CAPP
18 pages

**RR-94-12**
*Hubert Comon, Ralf Treinen:*
Ordering Constraints on Trees
34 pages

**RR-94-13**
*Jana Koehler:* Planning from Second Principles —A Logic-based Approach
49 pages

**RR-94-14**
*Harold Boley, Ulrich Buhrmann, Christof Kremer:*
Towards a Sharable Knowledge Base on Recyclable Plastics
14 pages

**RR-94-15**
*Winfried H. Graf, Stefan Neurohr:* Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces
20 pages

**RR-94-16**
*Gert Smolka:* A Foundation for Higher-order Concurrent Constraint Programming
26 pages

**RR-94-17**
*Georg Struth:*
Philosophical Logics—A Survey and a Bibliography
56 pages

## DFKI Technical Memos

**TM-92-04**
*Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:*
On the Representation of Temporal Knowledge
61 pages

**TM-92-05**
*Franz Schmalhofer, Christoph Globig, Jörg Thoben:*
The refitting of plans by a human expert
10 pages

**TM-92-06**
*Otto Kühn, Franz Schmalhofer:* Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

**TM-92-08**
*Anne Kilger:* Realization of Tree Adjoining Grammars with Unification
27 pages

**TM-93-01**
*Otto Kühn, Andreas Birk:* Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

**TM-93-02**
*Pierre Sablayrolles, Achim Schupeta:*
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

**TM-93-03**
*Harold Boley, Ulrich Buhrmann, Christof Kremer:*
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

**TM-93-04**
*Hans-Günther Hein:*
Propagation Techniques in WAM-based Architectures — The FIDO-III Approach
105 pages

**TM-93-05**
*Michael Sintek:* Indexing PROLOG Procedures into DAGs by Heuristic Classification
64 pages

**TM-94-01**
*Rainer Bleisinger, Klaus-Peter Gores:*
Text Skimming as a Part in Paper Document Understanding
14 pages

**TM-94-02**
*Rainer Bleisinger, Berthold Kröll:*
Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations
11 pages

## DFKI Documents

**D-93-11**
*Knut Hinkelmann, Armin Laux (Eds.):*
DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

**D-93-12**
*Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein:*
RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

**D-93-14**
*Manfred Meyer (Ed.):* Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
**Note:** This document is available only for a nominal charge of 25 DM (or 15 US-$).

**D-93-15**
*Robert Laux:*
Untersuchung maschineller Lernverfahren und heuristischer Methoden im Hinblick auf deren Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

**D-93-16**
*Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Gabriele Schmidt:* Design & KI
74 Seiten

**D-93-20**
*Bernhard Herbig:* Eine homogene Implementierungsebene für einen hybriden Wissensrepräsentationsformalismus
97 Seiten

**D-93-21**
*Dennis Drollinger:*
Intelligentes Backtracking in Inferenzsystemen am Beispiel Terminologischer Logiken
53 Seiten

**D-93-22**
*Andreas Abecker:*
Implementierung graphischer Benutzungsoberflächen mit Tcl/Tk und Common Lisp
44 Seiten

**D-93-24**
*Brigitte Krenn, Martin Volk:*
DiTo-Datenbank: Datendokumentation zu Funktionsverbgefügen und Relativsätzen
66 Seiten

**D-93-25**
*Hans-Jürgen Bürckert, Werner Nutt (Eds.):*
Modeling Epistemic Propositions
118 pages
**Note:** This document is available only for a nominal charge of 25 DM (or 15 US-$).

**D-93-26**
*Frank Peters:* Unterstützung des Experten bei der Formalisierung von Textwissen
INFOCOM:
Eine interaktive Formalisierungskomponente
58 Seiten

**D-93-27**
*Rolf Backofen, Hans-Ulrich Krieger, Stephen P. Spackman, Hans Uszkoreit (Eds.):*
Report of theEAGLES Workshop on Implemented Formalisms at DFKI, Saarbrücken
110 pages

**D-94-01**
*Josua Boon (Ed.):*
DFKI-Publications: The First Four Years
1990 - 1993
75 pages

**D-94-02**
*Markus Steffens:* Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff
90 pages

**D-94-03**
*Franz Schmalhofer:* Maschinelles Lernen:
Eine kognitionswissenschaftliche Betrachtung
54 pages

**D-94-04**
*Franz Schmalhofer, Ludger van Elst:*
Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung und kooperative Wissensevolution
22 pages

**D-94-06**
*Ulrich Buhrmann:*
Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien
117 pages

**D-94-08**
*Harald Feibel:* IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung
58 Seiten

**D-94-07**
*Claudia Wenzel, Rainer Hoch:*
Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten
25 Seiten