Laufer, C.C. & Fuks, H. (1995) ACCORD: Conversation Clichés for Cooperation, Proceedings of The International Workshop on the Design of
Cooperative Systems, Juan-les-Pins, França, Janeiro 1995, pp. 351-369

ACCORD
Representing Conversation Clichés for Cooperation

Hugo Fuks
Carlos Laufer
Daniel Schwabe

hugo@inf.puc-rio.br
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente 225
22453 RJ Brasil

Abstract

ACCORD, a framework for dialog representation systems using commitment is proposed as a means for representing some of the negotiation aspects characteristic of cooperative work. The modelling of each participant's range of information about the dialog, and its interaction with the others, is represented by commitments - placed in a commitment store - generated during the conversation process. Clichés - conversation stereotypes - are state transition machines that control the sequencing of dialog events in a conversation between two participants. The notion of contract - mapped to Winograd's *Conversation for Action* diagram - is presented as an example of the framework's ability for representing conversation clichés for cooperative work.

Keywords

Clichés, Commitment, Contract, Cooperative Work, Dialog, Conversation for Action

1. Introduction

Cooperation requires conversation. Conversations, particularly bureaucratic ones, tend to be structured and repetitive. Clichés are state transition machines that control the sequencing of dialog events between two participants in a conversation. A cliché restrains the unfolding of a conversation. Hence, it is possible to use specific clichés for reaching specific results like the opening of a current account or buying some article over the phone using a credit card.

This paper proposes the use of conversation clichés for structuring cooperative work. These clichés are built using the dialog primitives of ACCORD. In this framework,a dialog is a sequence of dialog events; a dialog event is a tuple consisting of a locution, and the speaker and the hearer of that locution. A locution is a locution modifier applied to one or more sentences.

In section 2 of this work ACCORD is described. The notion of cliché is introduced in section 3. In section 4, the notion of contract - mapped to Winograd's *Conversation for Action* diagram (Winograd 88) - is presented as an example of the framework's ability for representing conversation clichés with a view to cooperative work. Finally, the conclusion and future extensions of this work are presented in section 5.

2. ACCORD

ACCORD, a framework for dialog representation systems using commitment is proposed as a means for representing some of the negotiation aspects characteristic of cooperative work (Fuks 91a, 91b). It is a two-tiered framework comprising a Commitment Calculus (CC) and a Dialog Action Component (DAC).

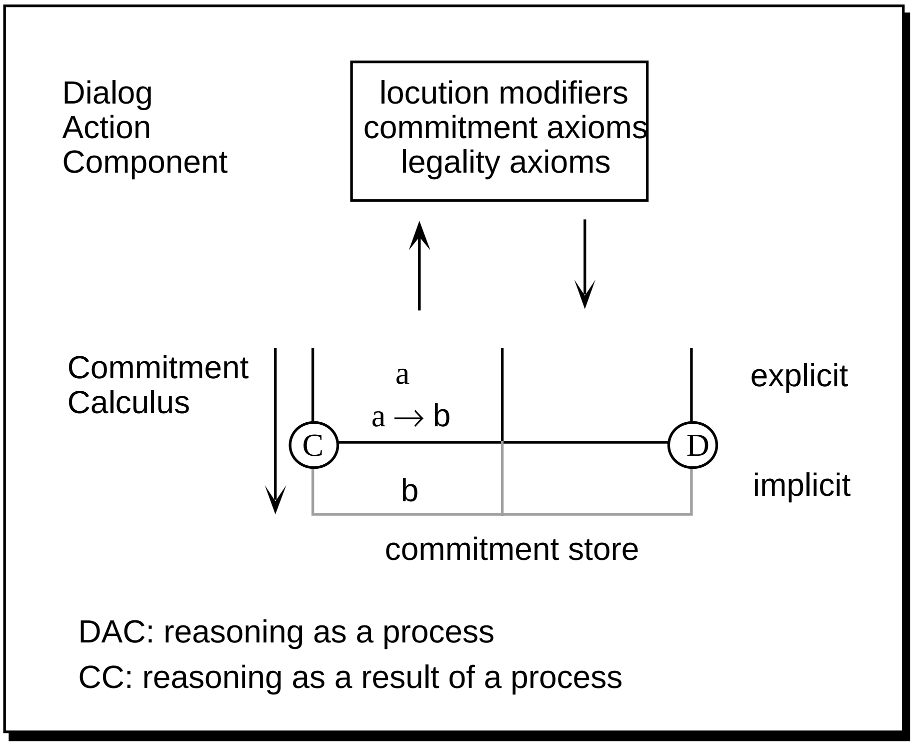


Figure 1 ACCORD’s organization.

The modelling of each participant's range of information about the dialog, and its interaction with the others, is represented by commitments generated during the conversation process. Each participant has a commitment store where its commitments are placed. The CC is responsible for dealing with the consequences of having commitments inside the commitment stores. Its rules envisage consistency preservation inside the commitment stores. The semantics of this calculus is based on the notion of commitment membership, i.e. the truth and falsity of formulas is defined by their membership status in the commitment stores.

DAC dictates the dialog’s etiquette, and the way that the conversation affects and updates the commitment stores. It has axioms of the form {Pre∅[Participant_AtoParticipant_B, Locution] Post}, where Participant_A (currently the speaker) and Participant_B (currently the hearer) are the participants in the dialog and Locutions are the application of a finite set of locution modifiers to statements. Pre and Post are the pre- and post-conditions of the locutions performed by the participant. In ACCORD, it is responsible for the insertion/deletion of commitments in the commitment stores. At the current stage of this work our dialogs are limited to two participants, justifying the sense of direction given to our locutions: AtoB (from Participant_A to Participant_B). However, in the future we would like to extend our framework to *N*-party.

Below, we introduce the part of the framework which is relevant for this article.

2.1. Locutions

Locutions consist of a statement and a modifier, represented as **locution modifier**(Statement). Statements are constructed in a propositional language which includes negation, conditional, disjunction and conjunction of statements. Locution modifiers are as follows:

Assertions, to be read as “It is the case that *Statement*”, notationally **asserts**(*Statement*).

Questions, to be read as “Is it the case that *Statement*?”, notationally **questions**(*Statement*).

Withdrawals, to be read as “I am not sure that *Statement*”, or “No commitment to *Statement*”, notationally **withdraws**(*Statement*).

Challenges, to be read as “Why is it to be supposed that *Statement*” or “How is it known that *Statement*”, notationally **why**(*Statement*).

Justifications, to be read as “*Statement* is a justification for the Challenge”, notationally **justifies**(*Statement*).

Denials, to be read as “I deny that it is the case that *Statement*”, notationally **denies**(*Statement*).

Resolution Demands, to be read as “Resolve this set of *Statements* against this specific *Statement*”, notationally **resolve**(*Set_of_Statements/Statement*).

2.2. Commitments

Having a commitment establishes a family of ways for a participant to behave, in a manner consistent with the information available and with the role that that participant performs in its community. A commitment is an engagement or involvement that restricts freedom of action. If one is committed to something, one should stick to it. For a better understanding of our notion of commitment, the reader is referred to (Fuks, Ryan & Sadler 89, Fuks 91a, 91b). There are other notions of commitment in the computing literature (Fikes 82, Cohen & Levesque 87, Koo 88, Woo 90, Bond 90). Our notion of commitment could be seen as a proper subset of Bond’s concept of commitment. Although we agree on the behavioral aspects of commitment, we have different views about retracting commitments. Moreover, we provide a calculus for dealing with commitments.

Commitments are placed in commitment stores, which are represented in the following way: *Participant(*C*(<Set of sentences>),* D*(<Set of sentences>)).* For the sake of clarity, each participant will not only have a commitment store for its positive and negative commitments - C part of the commitment store - but also a D part of the commitment store as a place for the statements which the participant is not committed to.

For example, (P_1(C(s1,s2)) P_2(C(s1), D(s3)) indicates that sentence ‘s1’ is in part C of the commitment stores of participants P_1 and P_2, that sentence ‘s2’ is in part C of P_1’s commitment store and finally that sentence ‘s3’ is in part D of P_2’s commitment store. We could read from that, that P_1 is committed to both ‘s1’ and ‘s2’, while P_2 is committed to ‘s1’ and is not committed to ‘s3’.

2.3. Commitment & Legality Axioms

Commitment axioms define the changes in the commitment stores and the *health* of the post-conditions caused by the uttering of each specific locution. Legality axioms define the proper sequencing of locution modifiers - the symbol “|” is being used to denote alternatives. In this work we are concerned only with the axioms involving the three locution modifiers of our example - **asserts**, **questions** and **withdraws**. For a thorough look at ACCORD’s locution modifiers, the reader is referred to (Fuks 91a).

2.3.1. Assertion

The assertion of a statement commits the speaker and the hearer to the statement (and to its immediate consequences, i.e. consequences inferred from the application of any rule of the commitment calculus once) described by that statement.

Commitment Axioms

$$\textit{healthy} \varnothing [P_1\text{to}P_2, \textbf{asserts}(s)]\ P_1(C(s)) \quad P_2(C(s))$$

After P_1 uttered **asserts**(s) to P_2, then given that Pre is a healthy state prior to the assertion, both participants are committed to s.

A participant is committed to anything stated by another participant. In other words, a participant can not deny having *heard* what the other said.

$$P_1(C(s)) \varnothing [P_1\text{to}P_2, \textbf{asserts}(s)]\ \textit{error}$$

After P_1 uttered **asserts**(s) to P_2, then given that P_1 is committed to s prior to the assertion, Post is an error state, and both commitment stores remain the same.

This rule essentially states that one is not allowed to re-assert one's commitment. Being repetitive is not a *healthy* way to advance reasoning.

Legality Axioms (s □ s')

$$\textit{healthy} \varnothing [P_1\text{to}P_2, \textbf{asserts}(s)][P_2\text{to}P_1, ((\textbf{asserts}(s')|\textbf{questions}(s)|\textbf{withdraws}(s))]\ \textit{healthy}$$

$$\textit{healthy} \varnothing [P_1\text{to}P_2, ((\textbf{asserts}(s)|\textbf{questions}(s)|\textbf{withdraws}(s))][P_2\text{to}P_1, \textbf{asserts}(s')]\ \textit{healthy}$$

The placing of a commitment inside the other participant's store is a way to force him to react to it. Silence has the force of agreement so, if he does not react to it he becomes committed to it and to its immediate consequences. To get rid of a commitment (in its commitment store) P_2 has to withdraw it. When given as an answer to a question, an assertion is the same as confirmation or a *yes.*

2.3.2. Question

To question a statement is to doubt (the very existence) of that statement. It does not commit either speaker or hearer to the questioned statement.

Commitment Axioms

$$\textit{healthy} \varnothing [P_1\text{to}P_2, \textbf{questions}(s)]\ \textit{healthy}$$

After P_1 uttered **questions**(s) to P_2, then given that Pre is a healthy state prior to the question, Post is a healthy state. Both commitment stores remain the same.

The uttering of a question does not affect the commitment stores.

$$P_1(C(s)) \varnothing [P_1\text{to}P_2\ \textbf{questions}(s)]\ \text{error}$$

After P_1 uttered **questions**(s) to $P_{2,}$ then given that P_1 is committed to s prior to the question, Post is an error state.

One cannot question one's commitments. A participant cannot doubt the existence of a commitment that is already in his commitment store, he can only **withdraw** it or **challenge** it.

Legality Axioms (s □ s')

$$\textit{healthy} \varnothing [P_1\text{to}P_2, \textbf{questions}(s)][P_2\text{to}P_1, ((\textbf{asserts}(s)|\textbf{withdraws}(s))]\ \textit{healthy}$$

$$\textit{healthy} \varnothing [P_1\text{to}P_2, ((\textbf{asserts}(s)|\textbf{withdraws}(s))][P_2\text{to}P_1, \textbf{questions}(s')]\ \textit{healthy}$$

After asking a question, it is up to the other participant to confirm it (**asserts**), deny it (**denies**) or distance himself from it (**withdraws**). The *question-answer* mechanism is very useful for advancing the reasoning process. By means of it, conclusions are made available and common ground is established that way.

2.3.3. Withdrawal

To withdraw a statement is to dissociate oneself from that statement. Nevertheless it is not just a rejection of a statement, because now the speaker is aware of it. To mark this difference, the statement is placed in the D part of the commitment store. If the speaker was committed to that statement previously to its withdrawal, after the withdrawal it is not committed to it any more.

Commitment Axioms

$$P_1(C(s)) \varnothing [P_1\text{to}P_2, \textbf{withdraws}(s)]\ P_1(D(s))$$

After P_1 uttered **withdraws**(s) to P_2, then given that P_1 is committed to s prior to the withdrawal, P_1 is not committed to s.

The situation pictured above typifies the way that a participant gets rid of a commitment that was probably - but not exclusively - placed there as the result of an assertion uttered by the other participant .

$$\textit{healthy} \varnothing [P_1\text{to}P_2, \textbf{withdraws}(s)]\ P_1(D(s))$$

After P_1 uttered **withdraws**(s) to P_2, then given that Pre is a healthy state prior to the withdrawal, P_1 is not committed to s.

The situation above reflects the awareness of the statement that comes from a comment like *I am not so sure that.* It is weaker than a commitment to the negation of the same statement.

$$P_1(D(s)) \varnothing [P_1\text{to}P_2, \textbf{withdraws}(s)]\ \textit{error}$$

After P_1 uttered **withdraws**(s) to $P_{2,}$ then given that P_1 is not committed to s prior to the withdrawal, Post is an error state.

One cannot withdraw what one has withdrawn before. We do not provide a mechanism for *forgetting* in our framework. This axiom stresses the *awareness* aspect given to our notion of commitment.

Legality Axioms

$$\textit{healthy} \varnothing [P_1\text{to}P_2,\textbf{withdraws}(s)][P_2\text{to}P_1,((\textbf{asserts}(s)|\textbf{questions}(s)|\textbf{withdraws}(s))]\textit{healthy}$$

$$\textit{healthy} \varnothing [P_1\text{to}P_2,((\textbf{asserts}(s)|\textbf{questions}(s)|\textbf{withdraws}(s))][P_2\text{to}P_1,\textbf{withdraws}(s)]\textit{healthy}$$

2.4. Dialog Representation

Dialogs have the following format:

$$\text{Pre} \varnothing [P_i\text{to}P_j, \text{Locution}]^n\ \text{Post}$$

where i and j {1,2}, i □ j; n ε 1 and Pre- and Post- are the conditions respectively holding before and after the dialog has taken place. Pre- and Post- are written in the language of commitments plus the constants *healthy* and *error*. While the former indicates that the state of the dialog is valid, the latter indicates that the state is not a valid one.

3. Conversation Cliché

We could define a *Conversation Cliché* for a specific situation, as the collection of the relevant conversations for that situation. Let us consider that only two participants are involved in an (verbal or written) exchange that we call dialog. Conversation is defined as the sequence of dialog events. For example, we could define the *Cliché for the Opening of a Current Account.*

This cliché is comprised by all the dialogs necessary for opening a current account. As part of the conversation with the bank clerk, the costumer might enquire about saving accounts. This particular enquiry would not be part of this specific cliché, because it is not relevant for opening a current account.

Clichés are modeled as state transition machines. They can be sub-divided into clichés or dialogs. They start from a initial state, traversing through intermediate states, untill reaching a final state. The transition from one state to another state can only be achieved by the triggering of a dialog event. Each path traversed in a cliché is associated to a valid dialog.

Representing conversation clichés for cooperation using ACCORD has the benefit of having, at the end of each dialog, the commitments undertaken by each participant during the dialog, recorded in its commitment store. The consequences of having taken part in the dialog, are for each participant, the same consequences of having commitments in its commitment store. The other way around, the content of the commitment stores mirror the *content* of the dialog.

Below, we use the notion of contract - a widespread form of social binding envisaging cooperation - as an example for the application of conversation clichés.

4. Example: Contract - *Conversation for Action*

According to the *Encyclopædia Brittanica* (Vol. 6, pp 424-431, 1965), ‘The simplest definition of a contract is that it is a promise enforceable by law. ... A usual requisite of a contactual promise is that it must be the product of a bargain; *i.e.*, the person to whom the promise is made (the promisee) must give to the person making the promise (the promisor) something in exchange for the promise. The thing given in exchange (called consideration) may be an act, like the payment of money... the making of a contract requires mutual assent of two or more persons. This means that ordinarily there must be an offer by one person (the offeror) and an acceptance by another (the offeree). ... An adequate description of a contract must include the legal consequences as well as the facts necessary to its creation. The traditional term used to describe these consequences is obligation. Obligation here means legal duty. One who makes a contractual promise to do an act comes under the legal duty to do the act.’

We are not interested in capturing all the aspects of a contract as defined above. Concepts like consideration and obligation were intentionally left out of this work, although we proposed elsewhere (Fuks, Ryan & Sadler 89) the existence of a link between having commitments and being obliged to perform actions for realizing this commitments. Here we are interested in showing that ACCORD is capable of representing and recording the promise and offer exchange, together with the commitments originated from it.

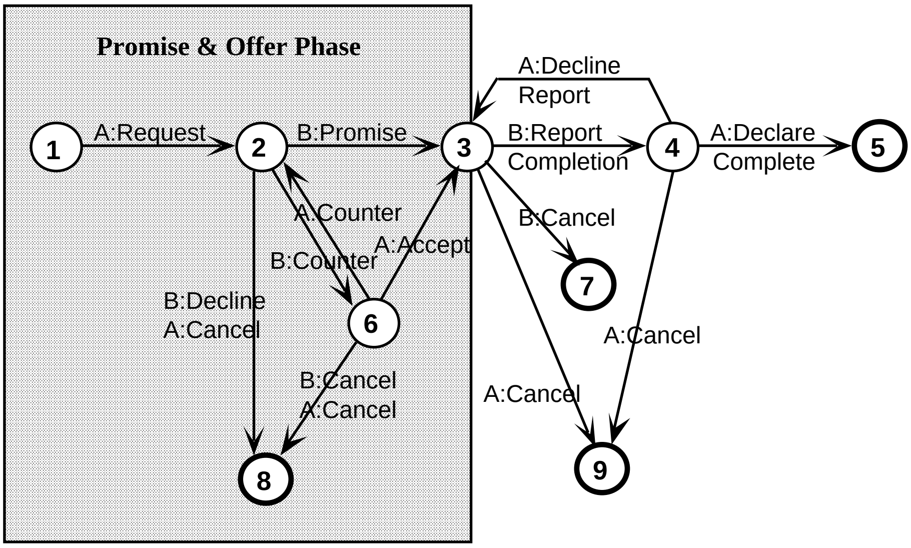


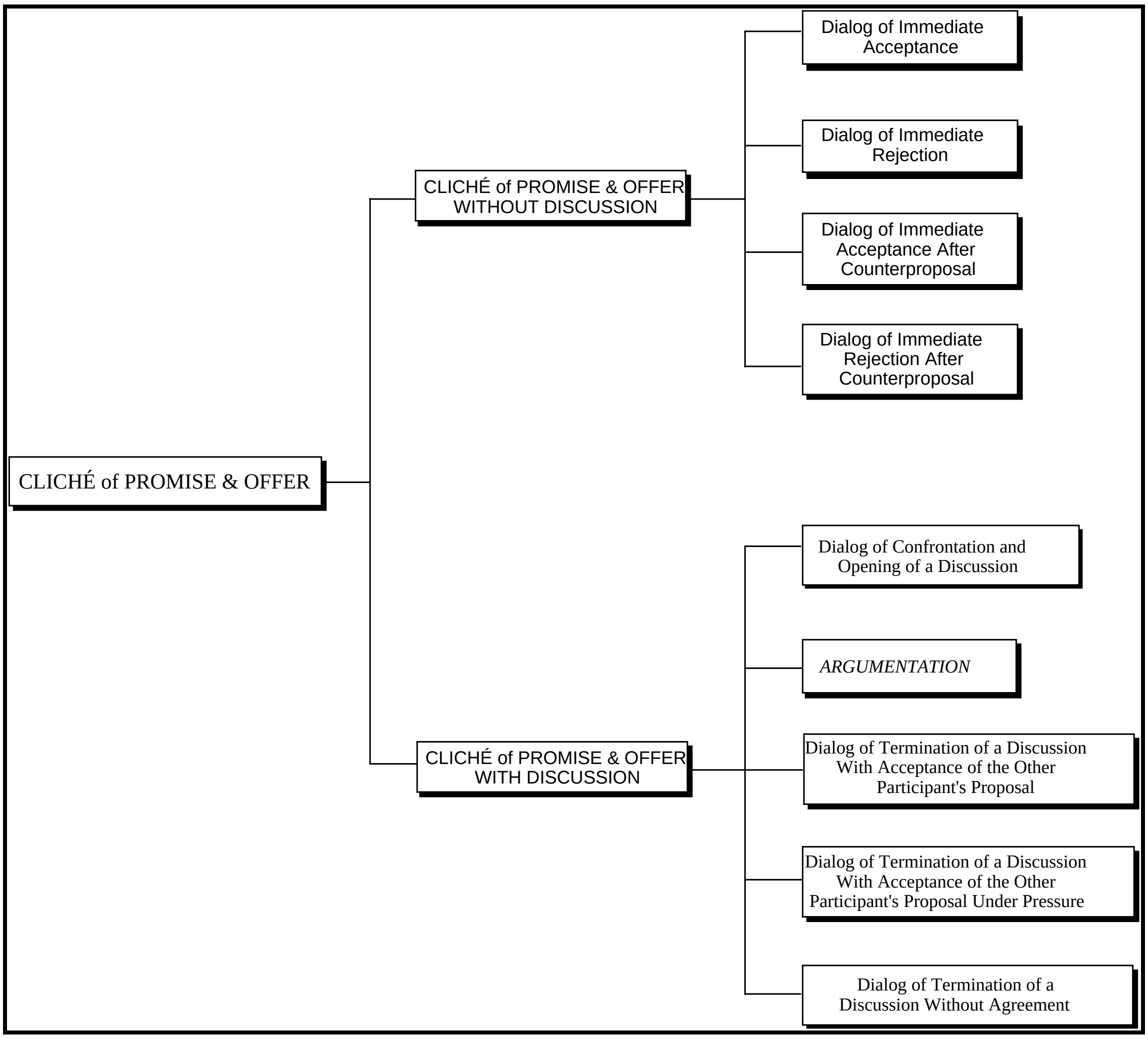
Figure 2 Conversation for Action diagram.

The *Conversation for Action* diagram presented above was taken from (Winograd 88, pp. 628). Observing the diagram is possible to identify three phases which are similar to those of a contract. The first one is the Promise & Offer Phase, where the desired action is specified. The second one is the Performance Phase. Finally, there is a Completion Phase. For our example we will stick to the first phase.

In the following sections we define a *Cliché of Promise & Offer* which correponds to the Promise & Offer Phase of the diagram. This cliché is divided into two other clichés namely, *Cliché of Promise & Offer Without Discussion* and *Cliché of Promise & Offer With Discussion*.

For the analysis below we replace ‘A’ (from the diagram) with ‘offerer’ and ‘B’ with ‘promiser’. We also consider the existence of only two proposals at stake, one per participant. We refer to the offerer’s proposal as ‘a’ and to the promiser’s counterproposal as ‘b’. It also considers that at the end of each dialog, either one proposal is accepted by both participants or no proposal is accepted.

Firstly, we present a taxonomy for the *Cliché of Promise & Offer*:


CLICHÉ of PROMISE & OFFER
CLICHÉ of PROMISE & OFFER WITHOUT DISCUSSION
Dialog of Immediate Acceptance
Dialog of Immediate Rejection
Dialog of Immediate Acceptance After Counterproposal
Dialog of Immediate Rejection After Counterproposal
CLICHÉ of PROMISE & OFFER WITH DISCUSSION
Dialog of Confrontation and Opening of a Discussion
ARGUMENTATION
Dialog of Termination of a Discussion With Acceptance of the Other Participant's Proposal
Dialog of Termination of a Discussion With Acceptance of the Other Participant's Proposal Under Pressure
Dialog of Termination of a Discussion Without Agreement

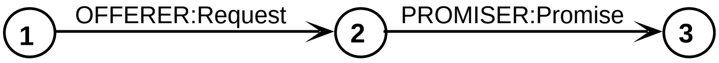
4.1. Cliché of Promise & Offer Without Discussion

This cliché is defined by the dialogs that take place between offerer and promiser, whenever there is no discussion about the terms of the contract. It is independent of the subject of the contract.

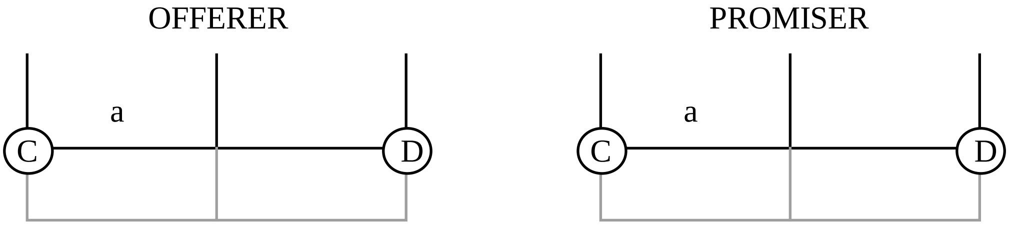
The *Cliché of Promise & Offer Without Discussion* comprises four dialogs, all of them starting with an offerer’s request. According to Searle (79), *to request* is a directive i.e., speech acts where the speaker wants the hearer to perform some action. In terms of ACCORD's locution modifiers, **questions** perfectly suit the task, for it does not alter the commitment stores and *forces* the promiser to give an answer that commits itself.

4.1.1. Dialog of Immediate Acceptance

This dialog takes place when the promiser immediately accepts the offerer's proposal. After the initial **questions**(a) by the offerer, in order to commit itself to 'a', the promiser **asserts**(a), introducing that way, the offerer's proposal 'a' into the C part of both commitment stores.

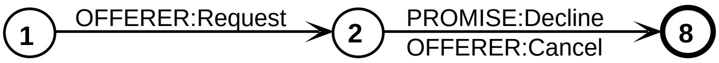


healthy ∅ [OFFERERtoPROMISER, **questions**(a)]
[PROMISERtoOFFERER, **asserts**(a)]
OFFERER(C(a))   PROMISER(C(a))

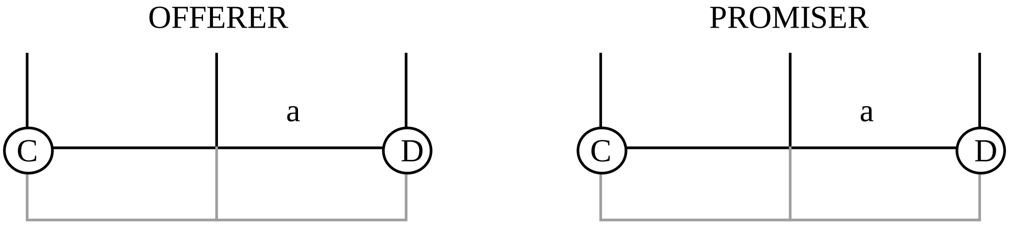


4.1.2. Dialog of Immediate Rejection

This dialog takes place when the promiser immediately rejects the offerer's proposal. After the initial **questions**(a) by the offerer, in order to show its no commitment to 'a', the promiser **withdraws**(a), introducing that way, the offerer's proposal 'a' into the D part of its commitment store. To finish off the dialog, the offerer also **withdraws**(a) - its own proposal, introducing it into the D part of its commitment store.

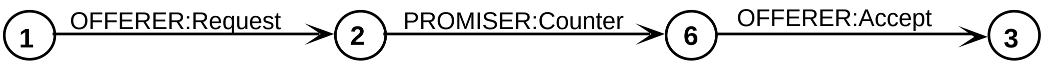


healthy ∅ [OFFERERtoPROMISER, **questions**(a)]
[PROMISERtoOFFERER, **withdraws**(a)]
[OFFERERtoPROMISER, **withdraws**(a)]
OFFERER(D(a))   PROMISER(D(a))



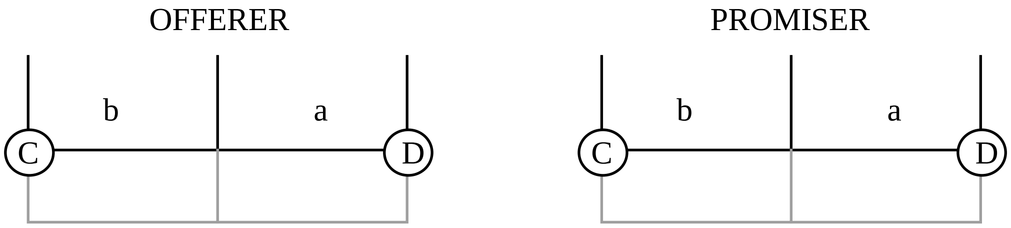
4.1.3. Dialog of Immediate Acceptance After Counterproposal

This dialog takes place when the promiser poses a counterproposal to the offerer's request, and the offerer immediately accepts it.



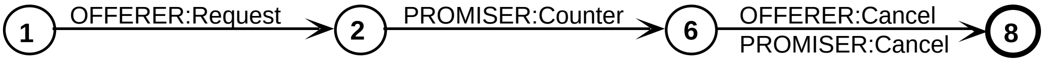
After the initial **questions**(a) by the offerer, promiser replies with an **asserts**(b), introducing that way, its counterproposal ‘b’ into the C part of both commitment stores. Offerer shows its acceptance to the promiser’s counterproposal by withdrawing its own proposal ‘a’ - placing it into the D part of its commitment store. To finish off the dialog, the promiser also **withdraws**(a) - the offerer’s proposal, introducing it into the D part of its commitment store.

healthy ∅ [OFFERERtoPROMISER, **questions**(a)]
[PROMISERtoOFFERER, **asserts**(b)]
[OFFERERtoPROMISER, **withdraws**(a)]
[PROMISERtoOFFERER, **withdraws**(a)]
OFFERER(C(b), D (a))   PROMISER(C(b), D(a))



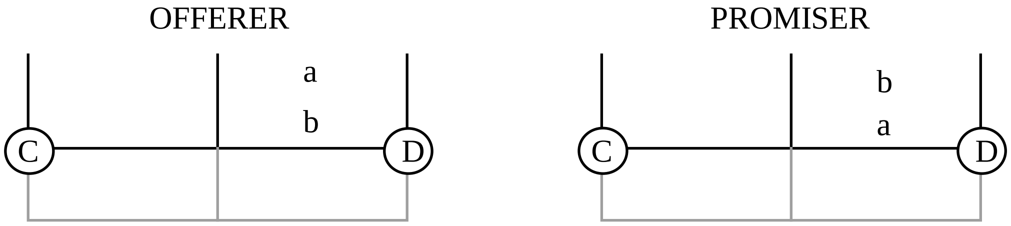
4.1.4. Dialog of Immediate Rejection After Counterproposal

This dialog takes place when the promiser poses a counterproposal to the offerer’s request, and the offerer immediately rejects it.



After the initial **questions**(a) by the offerer, the promiser replies with an **asserts**(b), introducing that way, its counterproposal ‘b’ into the C part of both commitment stores. The offerer shows its rejection to the promiser’s counterproposal by withdrawing the promiser’s counterproposal ‘b’ - placing it into the D part of its commitment store. To finish off the dialog, the promiser also **withdraws**(b) - its own proposal, and both of them withdraw their own proposals, thus introducing them into the D part of both commitment stores.

healthy ∅ [OFFERERtoPROMISER, **questions**(a)]
[PROMISERtoOFFERER, **asserts**(b)]
[OFFERERtoPROMISER, **withdraws**(b)]
[PROMISERtoOFFERER, **withdraws**(a)]
[OFFERERtoPROMISER, **withdraws**(a)]
[PROMISERtoOFFERER, **withdraws**(b)]
OFFERER(D(a,b))   PROMISER(D(b,a))



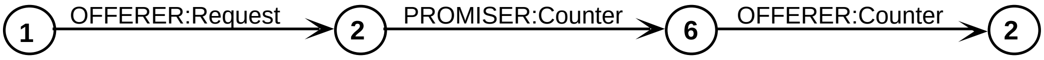
4.2. Cliché of Promise & Offer With Discussion

This cliché is defined by the dialogs that take place between offerer and promiser, whenever there is a discussion about the terms of the contract.

This cliché is comprised by the dialogs that cover the four stages of a debate - confrontation, opening, argumentation and conclusion (van Eemeren & Grotendorst 84). The first two stages are mapped to a c*onfrontation and opening of discussion dialog*. The concluding stage is mapped to its three possible outcomes: termination of a discussion with the acceptance of the offerer’s proposal, termination of a discussion with the acceptance of the promiser’s counterproposal and termination of a discussion without agreement. We intentionally left the argumentation stage out of this work, because its definition requires a study in detail of the common aspects of debates, which is the subject of a future extension of this research.

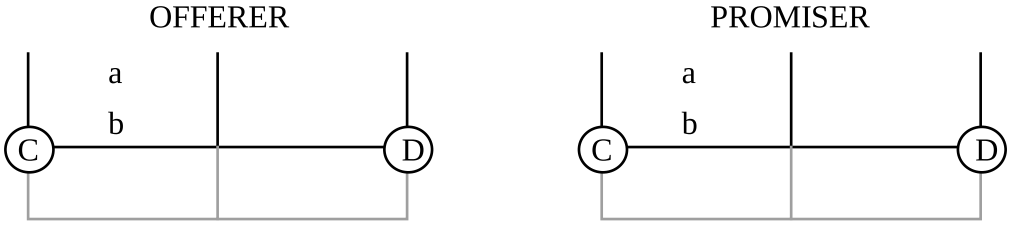
4.2.1. Dialog of Confrontation and Opening of a Discussion

This dialog takes place when the promiser poses a counterproposal to the offerer’s request, and they decide to discuss the two proposals at stake.



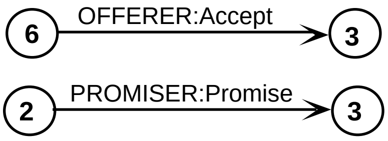
After the initial **questions**(a) by the offerer, the promiser replies with an **asserts**(b), introducing that way, its counterproposal ‘b’ into the C part of both commitment stores. The offerer shows its decision to engage in a discussion by asserting its own original proposal ‘a’ - placing it into the C part of both commitment stores. Thus, at this point of the conversation, proposal and counterproposal are in the C part of both commitment stores.

healthy ∅ [OFFERERtoPROMISER, **questions**(a)]
[PROMISERtoOFFERER, **asserts**(b)]
[OFFERERtoPROMISER, **asserts**(a)]
OFFERER(C(a,b))   PROMISER(C(a,b))



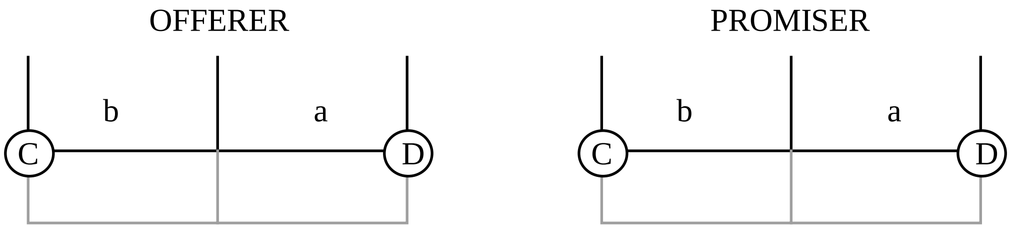
4.2.2. Dialog of Termination of a Discussion With Acceptance of the Other Participant’s Proposal

This dialog takes place when one of the participants accepts the other’s proposal while the discussion is going on.

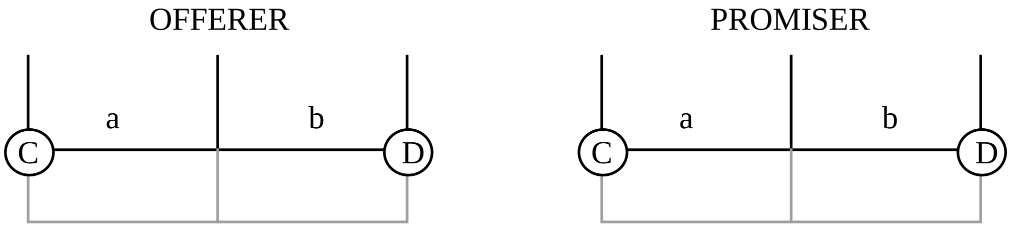


At some point during the discussion, one of the participants voluntarily **withdraws** its own proposal. The other participant also **withdraws** that same proposal.

OFFERER(C(a,b))   PROMISER(C(a,b)) ∅
[OFFERERtoPROMISER, **withdraws**(a)]
[PROMISERtoOFFERER, **withdraws**(a)]
OFFERER(C(b), D(a))   PROMISER(C(b), D(a))

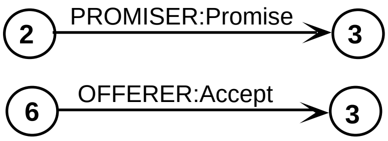


OFFERER(C(a,b))   PROMISER(C(a,b)) ∅
[PROMISERtoOFFERER, **withdraws**(b)]
[OFFERERtoPROMISER, **withdraws**(b)]
OFFERER(C(a), D(b))   PROMISER(C(a), D(b))



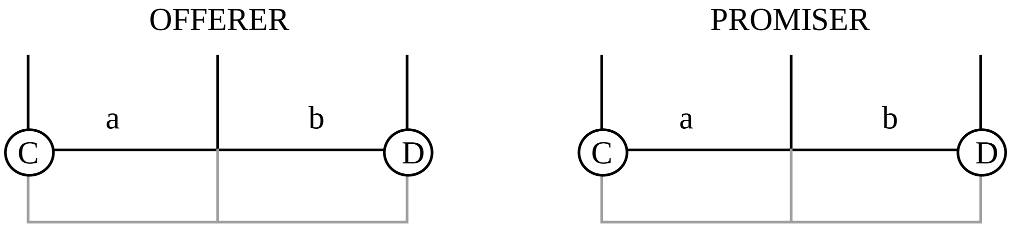
4.2.3. Dialog of Termination of a Discussion With Acceptance of the Other Participant’s Proposal Under Pressure

This dialog takes place when one of the participants retracts the other participant’s proposal while the discussion is going on, forcing it that way to take a position. The other participant also retracts its own proposal, showing that way its acceptance of the former’s proposal.

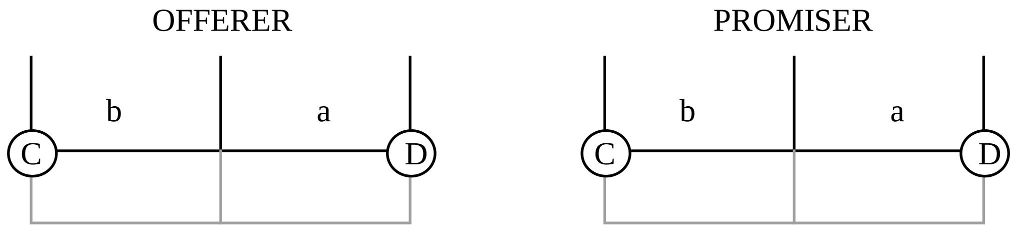


At some point during the discussion, one of the participants **withdraws** the other participant’s proposal. The other participant *gives in* and **withdraws** its own proposal.

OFFERER(C(a,b))   PROMISER(C(a,b)) ∅
[OFFERERtoPROMISER, **withdraws**(b)]
[PROMISERtoOFFERER, **withdraws**(b)]
OFFERER(C(a), D(b))   PROMISER(C(a), D(b))

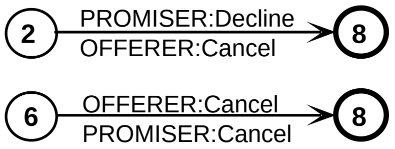


OFFERER(C(a,b))   PROMISER(C(a,b)) ∅
[PROMISERtoOFFERER, **withdraws**(a)]
[OFFERERtoPROMISER, **withdraws**(a)]
OFFERER(C(b), D(a))   PROMISER(C(b), D(a))



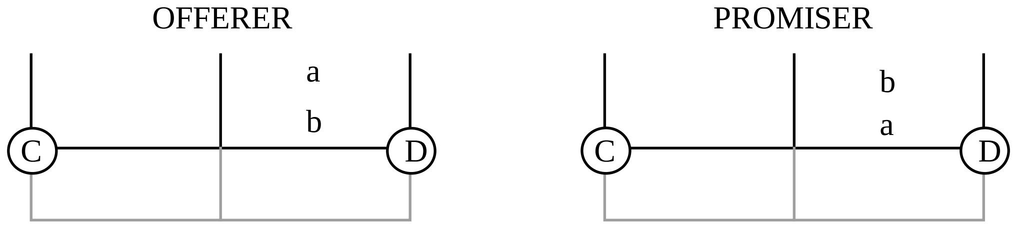
4.2.4. Dialog of Termination of a Discussion Without Agreement

This dialog takes place when one of the participants retracts the other participant’s proposal while the discussion is going on, forcing it that way to take a position. The other participant retracts the former’s proposal, showing that way that there is no agreement about that matter.



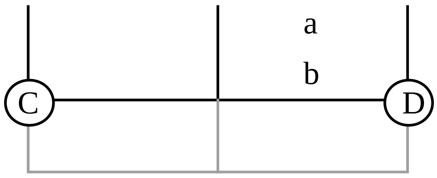
At some point during the discussion, one of the participants **withdraws** the other participant’s proposal. The other participant does not *give in* and **withdraws** the former’s proposal. To finish off the dialog, both participants withdraw their own proposals.

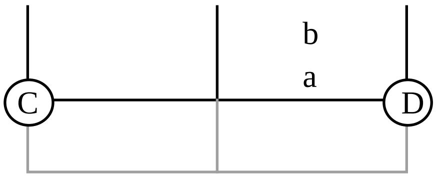
OFFERER(C(a,b))   PROMISER(C(a,b)) ∅
[OFFERERtoPROMISER, **withdraws**(b)]
[PROMISERtoOFFERER, **withdraws**(a)]
[OFFERERtoPROMISER, **withdraws**(a)]
[PROMISERtoOFFERER, **withdraws**(b)]
OFFERER(D(a,b))   PROMISER(D(b,a))



OFFERER(C(a,b))   PROMISER(C(a,b)) ∅
[PROMISERtoOFFERER, **withdraws**(a)]
[OFFERERtoPROMISER, **withdraws**(b)]
[PROMISERtoOFFERER, **withdraws**(b)]
[OFFERERtoPROMISER, **withdraws**(a)]
OFFERER(D(a,b))   PROMISER(D(b,a))

OFFERER   PROMISER





4.3. Final Considerations

In the table below, we show the final content of the commitment stores of the eight dialogs from the previous example.

DIALOGS	COMMITMENT STORES			
	OFFERER		PROMISER	
	C	D	C	D
immediate acceptance	a		a	
immediate rejection		a		a
immediate acceptance after counterproposal	b	a	b	a
immediate rejection after counterproposal		a b		b a
confrontation and opening of a discussion	a b		a b	
termination of a discussion with acceptance of the other participant’s proposal**	b (a)	a (b)	b (a)	a (b)
termination of a discussion with acceptance of the other participant’s proposal under pressure**	a (b)	b (a)	a (b)	b (a)
termination of a discussion without agreement		a b		b a

** The option in brackets shows the other possible configuration of commitment stores for this dialog.

Some configurations are unique, making it possible to identify in a straightforwardly way the dialog that has generated them, as it is the case of the *dialog of immediate acceptance*. On the other hand, some configurations are the same for different dialogs, which is the case for the *dialog of immediate rejection after counterproposal* and *dialog of termination of a discussion without agreement*, being impossible to distinguish which dialog has generated them - for this reason, ACCORD also keeps an event record. From that we can learn that one could end up with the same commitments - or *not* commitments in this particular case - following differents paths of a cliché.

5. Conclusion and Future Work

Bureaucratic conversations are structured and repetitive. Clichés are state transition machines that control the sequencing of dialog events between two participants in a conversation. Representing conversation clichés using ACCORD, has the benefit of having at the end of each dialog, the commitments undertaken by each participant during the dialog, recorded in its commitment store.

Starting from the notion of contract, which is paramount for cooperation, we developed a *Cliché of Promise & Offer*. It was based on the first phase of Winograd's *Conversation for Action* diagram. For this task, only three of ACCORD's locution modifiers were used - **questions**, **asserts** e **withdraws**. A taxonomy for the *Cliché of Promise & Offer* was presented and some observations were made about the example.

An immediate extension of this work is the development of the *ARGUMENTATION* stage of the *Cliché of Promise & Offer With Discussion*. It will require a meticulous research about the common characteristics of debates. For example, it could have its offer and counter-offer steps enacted by using the argumentative structures presented by (van Eemeren & Grotendorst 84). The conversation built around a valid argument would progressively construct a dialog structure, which should map to a well formed Toulmin structure (Toulmin 64) or other similar structure. ACCORD's other locution modifiers - **why**, **justifies**, **denies** and **resolve** - will be used in this extension.

ACCORD seems to suit applications like software configuration management (Finkelstein & Fuks 90) where specifications - versions - are treated as commitments that are established, updated and discharged by members of the software development group. Currently we are extending the framework for the development of groupware for cooperative software design (Lucena, Leite, Schwabe & Fuks 91), especially from the team activity point of view (Hahn, Jarke & Rose 91).

We follow a path similar to the one taken by (Rittel & Kunz 70, Lowe 85 and Schuler & Smith 90) where design decisions and rationales are logged using argumentation structures. In our case, the content of the commitment stores at the end of each dialog, provides for the log containing the argument's basis. Taking the *dialog of termination of a discussion without agreement* as an example, from the placement of the proposal and the counterproposal in the D part of both commitment stores, one could infer that both proposals were put at stake, to be finally rejected.

Finally, we are implementing ACCORD's basic machine in Prolog (Ridolfi, Fuks & Schwabe 92). At the current stage of this work, we are dealing with a small set of locution modifiers. The addition of new locution modifiers might prove necessary for coping with yet unknown situations. We hope that our experiments with software design will give us some insight as to whether the addition of new locution modifiers is necessary.

Acknowledgements

The authors would like to thank Carlos J. P. Lucena for his comments on a previous version of this paper. This work was partly supported by CNPq - Brazilian National Council for Research and Development.

References

Bond, A. H.: A Computational Model for Organizations of Cooperating Intelligent Agents: Proceedings Conference on Office Information Systems, SIGOIS Bulletin, Vol. 11, nº 2 & 3, pp 21-30, 1990.

Cohen, R. P. & Levesque, J. H.: *Persistence, Intention and Commitment*; in Formal Theories of Communication, (eds.) P. R. Cohen & C. R. Perrault, Report nº CSLI-87-69, pp 171-203, 1987.

Finkelstein, A. & Fuks H.: Conversation Analysis and Specification; in Computers and Conversation, ed: P. Luff, N. Gilbert & D. Frohlich, pp 173-186, Academic Press, 1990.

Fuks, H., Ryan M. & Sadler, M.: Outline of a Commitment Logic for Legal Reasoning; Proc 3rd International Conference on Logics, Informatics and Law; Florence, Nov.1989.

Fuks, H.: *Negotiation using Commitment and Dialogue*, PhD. Thesis, Department of Computing, Imperial College, University of London, Feb. 1991(a).

Fuks, H.: ACCORD - A Framework for Dialogue Representation using Commitment; Research Report nº 24/91, Dept. de Informática, PUC-RJ, Dezembro 1991(b).

Hahn, U., Jarke, M. & Rose, T.: Teamwork Support in a Knowledge-Based Information Systems Environment; IEEE Transactions on Software Engineering, Vol.17, nº.5, pp 467-482, May 1991.

King, M.: Designing for Cooperation: Cooperating in Design; Communications of the ACM, Vol. 34, nº 12, pp 65-73, December 1991.

Koo, C. & Wiederhold, G.: A commitment-based communication model for distributed office environments; Proceedings Conference on Office Information Systems, SIGOIS Bulletin, Vol 9, nº 2 & 3, pp 291-298, 1988.

Lowe, D.G.: Co-operative structuring of information: the representation of reasoning and debate; Int. Journal of Man-Machine Studies, Academic Press, pp 97-111, 1985.

Lucena, C. J. P., Leite, J., Schwabe, D. & Fuks, H.: A Research Agenda for Software Design; Monografias em Ciência da Computação Research Report nº 29/91, Departamento de Informática, PUC-RJ, Março 1991.

Ridolfi, L. F. G. G. M., Fuks, H. & Schwabe, D.: ACCORD - Máquina Básica; Research Report nº 2/92, Dept. de Informática, PUC-RJ, March 1992, in portuguese.

Rittel, W and Kunz, W: “Issues as Elements of Information Systems”; Working Paper 131, Center for Planning and Development Research, University of California, Berkeley, 1970.

Schuler, W. & Smith, J.B.: Author's Argumentation Assistant (AAA): A Hypertext-Based Authoring Tool for Argumentative Texts; Proc European Conference on Hypertext, France, pp 137-151, ed: A. Risk, N. Streitz & J. Andre; The Cambridge Series on Electronic Publishing, November 1990.

Searle, J.R.: Expression and Meaning - Studies in the Theory of Speech Acts; Cambridge University Press, 1979.

Toulmin, S.E.: The Uses of Arguments; Cambridge University Press, 1958.

van Eemeren, F.H. & Grootendorst, R.: Speech Acts in Argumentative Discussions; Foris Publications, 1984.

Winograd, T.: *A Language/Action Perspective on the Design of Cooperative Work*; In Computer-Supported Cooperative Work: A Book of Readings; ed: I. Greif, Morgan Kaufmann Publishers, 1988.

Woo, C. C.: Sact - A Tool for Automating Semi-Structured Organizational Communication; Conf. on Office Information Systems, ACM Press, 1990.