PUC

Monografias em Ciência da Computação
nº 24/91

ACCORD
A Framework for Dialogue Representation using Commitment

Hugo Fuks

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453

RIO DE JANEIRO - BRASIL

PUC RIO - DEPARTAMENTO DE INFORMÁTICA

Monografias em Ciência da Computação, Nº 24/91
Editor: Carlos J. P. Lucena

Dezembro, 1991

ACCORD
A Framework for Dialogue Representation using Commitment *

Hugo Fuks

* This work has been sponsored by the Secretaria de Ciência e Tecnologia da Presidência da República Federativa do Brasil.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC Rio - Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453 - Rio de Janeiro, RJ
Brasil

Tel.:(021)529-9386 Telex:31078 Fax:(021)511-5645
E-mail:rosane@inf.puc-rio.br

Resumo

ACCORD é uma estrutura para sistemas de representação de diálogos. Seu uso é proposto para a representação de alguns dos aspectos de negociação característicos do trabalho cooperativo. Esta estrutura é dividida em um Cálculo de Compromissos e um Componente de Diálogo Ação. A modelagem do conhecimento de cada participante sobre o diálogo e sobre a sua interação com os outros participantes, é representada por meio dos compromissos assumidos durante a conversação. Cada participante tem uma carteira de compromissos onde seus compromissos são registrados. O Cálculo de Compromissos foi desenvolvido para lidar com as consequências de ter compromissos. O Componente Diálogo Ação dita a etiqueta diálogo, e a maneira pela qual a conversação afeta e atualiza as carteiras de compromissos. As noções de cliches, scripts e de padrões de raciocínio cooperativo são esboçadas visando a formação de estereótipos de conversação .

Palavras-chave

Compromisso; Diálogo; Cooperação e Representação de Conhecimento.

Abstract

ACCORD, a framework for dialogue representation systems using commitment is proposed as a means for representing some of the negotiation aspects characteristic of cooperative work. A two-tiered framework comprising a Commitment Calculus and a Dialogue Action Component is developed. The modelling of each participant's range of information about the dialogue, 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 Commitment Calculus is developed for dealing with the consequences of having commitments. The Dialogue Action Component dictates the dialogue's etiquette, and the way that the conversation affects and updates the commitment stores. The notions of cliches, scripts and patterns of cooperative reasoning for the provision of conversation stereotypes are outlined.

Keywords

Commitment; Dialogue; Cooperation and Knowledge Representation.

ACCORD
A Framework for Dialogue Representation using Commitment

Hugo Fuks
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 dialogue representation systems using commitment is proposed as a means for representing some of the negotiation aspects characteristic of cooperative work. A two-tiered framework comprising a Commitment Calculus and a Dialogue Action Component is developed. The modelling of each participant's range of information about the dialogue, 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 Commitment Calculus is developed for dealing with the consequences of having commitments. The Dialogue Action Component dictates the dialogue's etiquette, and the way that the conversation affects and updates the commitment stores. The notions of cliches, scripts and patterns of cooperative reasoning for the provision of conversation stereotypes are outlined.

Keywords

Commitment; Dialogue; Cooperation and Knowledge Representation.

1. Introduction

Negotiation permeates many aspects of our everyday life. Normally, the accomplishment of a task involving more than one party is by means of agreements, commitments and compromises.

We see the interactional aspects of language as a sound common basis for the structuring and coordination of cooperative relationship among groups of people and groups of computer systems. People talking to people is by far the best way of establishing common ground, working together, negotiating contracts and solving conflicts. We believe that computerised conversational working environments are going to be extended and improved in such a way that soon it will be normal practice to see people *talking* to computer systems, computer systems *talking* to people and computer systems *talking* to computer systems. This is neither to be seen as the use of speech nor the use of natural language, but we are interested in the usage of conversation primitives to guide this process.

But we would like to go even further, extending the conversational framework for the outer circle of communication among groups of people and groups of computer systems to a dialogic framework for the inner circle of reasoning, fundamental to the performance of importance of logics for the programming of computer systems nowadays. Then, we should remind the reader that logics and dialectics, literally originated from the works of the very same Aristotle. For reasons that fall outside the scope of this work, dialectics was associated with rhetoric, which in its turn was associated with bad argumentation practices. We understand that today we are prepared enough, in terms of theoretical results and in machinery, to reaccess dialectics and to use it as a sound framework for automated reasoning and communication.

1.1. Forms of Reasoning

Aristotle divides reasoning - or logics - into two basic forms: analytical and dialectical. We also identify a third form of reasoning - oratorical - as we present below:

i) Analytical reasoning: It is the form of reasoning that conforms to the laws of formal logic. The notion of *truth* prevails here. Results can be demonstrated, i. e. calculated in accordance with rules that have been laid down in advance.

ii) Dialectical reasoning: It is founded on opinion and concerned with contingent realities. It is not *formally* valid but it is only reasonable or probable. It is founded on agreement. It conforms to the laws of interaction and minimum or immediate consistency preservation. The parties involved make concessions and commitments to achieve a *satisfactory* mid-term position.

iii) Oratorical reasoning: In this sort of reasoning the parties present their cases, leaving the decision to a third party. It presupposes the existence of a *tertius* for harmonising conflicting views.

One could consider examining each of these three forms of reasoning on the basis of the qualitative aspects of the number of parties involved. Analytical reasoning is related to one party, the dialectical form is related to two parties and the oratorical form is related to three parties.

Logics, 'considered in its most meaningful nucleus - which is the theory of deduction' (Ladriere 89) - and argumentation relate to statements - propositions - in an opposite, but not conflicting way. While deduction guarantees the passing of *truth* from a collection of statements - premises - to those statements - conclusions - that follow from it, argumentation traverses back along this *path* providing justification for the conclusions by making the premises available.

Metaphorically speaking, this *path* could be seen as a river of *implications* where deduction flows downstream while argumentation zigzags upstream. Negotiation seems to zigzag more than flow. It also comes closer to argumentation for scarcely making use of the notion of *truth*. During the negotiation process one is busier in giving support to one's views of the subject than concluding things about it, especially if one understands negotiation as the process of compromising and reaching agreement.

We propose the use of dialectical reasoning as a course of action for negotiation. Argumentation epitomizes it, being the use of debate and methodical reasoning for solving conflicts. We believe that a dialogic framework is suitable for capturing the relevant features of argumentation, especially the interaction mechanisms embedded in it.

A framework for dialogue representation systems is proposed as a means for representing some of the negotiation aspects characteristic of cooperative work. An action logic presentation is chosen for its ability to talk about actions, that is paramount for making the interactional aspects of the negotiation process explicit. This way we are able to talk about straightforward things like questions and answers, which is not easy to do with other logic presentations.

Concerning the modelling of each party's range of information about the dialogue, we are more interested in developing a *calculus* for dealing with the interaction process between different parties' actions - a pragmatic approach, rather than simulating their thinking process - a psychological approach. While the latter asks for a highly specialized theory of belief, the former seeks a much simpler set of rules for coordination and conflict resolution among different parties' interests. 'We work together by making commitments so we can successfully anticipate the actions of others and coordinate them with our own' (Winograd 88). The notion of commitment seems to embody the visible aspects involved in the negotiation process. Finally, (Hamblin 87) bonds the notions of commitment and belief: "In short, the concept of belief is an idealization of that of indicative commitment."

2. A Dialogic Framework

In the previous section we proposed the use of a dialogic framework as a platform for negotiation to solve conflicts among parties. This section briefly surveys the field and introduces part of the chosen formalism through examples originating from software specification.

2.1. Survey

The underlying vehicle for defining our framework is a formal account of dialogue. Such accounts have their roots in a number of different traditions:

the *game theoretic semantics tradition* in which "dialogue games" are used to define the meanings of components of a formal language, for example (Lorenz 82);

the *foundations of logic tradition* in which an understanding of the communicative context of argument is examined to understand the development of different logical traditions, notably (Hamblin 71);

the *human computer interaction tradition* in which representations of dialogues are developed for design and evaluation of user interfaces, for example (Schneiderman 82) and (Green 83);

the *rhetorical or argumentative tradition* in which a model of dialogue provides the normative base for deciding what constitutes rhetorical "competence", for example (Allwood 86);

the *natural language processing tradition* in which computationally tractable models are sought to provide a basis for automatically interpreting and generating dialogues, for example (Carbonel 82);

the *distributed artificial intelligence tradition* in which computational models of multi-agent "negotiation" are constructed to integrate diverse knowledge sources, notably (Erman & Lesser 75), (Smith 80), (Smith & Davis 81), (Kornfeld & Hewitt 81) and (Lenat 75).

We have sought to combine the formal apparatus - dialogue formalism - of the foundations of logic tradition with the approach - cooperation and negotiation - of the distributed artificial intelligence tradition. We have explicitly rejected the competitive approach typical of the game theoretic semantics tradition and are not directly concerned with the discourse level issues that dominate both the natural language processing and the rhetorical tradition. The descriptive tools provided by the human computer interaction tradition lack the required expressiveness for the less highly constrained dialogues on which we have focussed.

2.2. Hamblin's formalism

Hamblin's formalism for the analysis of dialogue is based on the notions of legality - as a means of providing the dialogue with etiquette - and commitment. Each party of the dialogue has a commitment store where its own state of affairs is being registered and updated through a conjunction of commitments.

Mackenzie (79, 80, 81, 84) extended this formalism with systems DT, DC, DD and DC+. We have adopted his dialogue system DC as the basis for the design of a dialogic framework - with a strong cooperative flavour - for negotiation. In the examples that follow, DC is introduced using our own notation and with modifications that are spelled out. The basic concepts nevertheless, were preserved.

2.3. Introducing the formalism

Human organizations, being based on team activity, strongly rely upon negotiation. We introduce the dialogue formalism through short conversations, which are common in the office context. The term dialogue, as generally used, refers to a conversation or spoken interaction between two parties.

Next we apply the dialogic framework to enable a few short dialogues to take place between two parties. Each has differing perspectives on, and knowledge about, the domain, as well as a variety of skills, roles and so on. In some cases the perspectives may be based on underlying contradictions. Below we briefly introduce the context within which our examples are developed.

A party is a *logical* participant in the dialogue. We can loosely define a party as an agent responsible for maintaining a particular perspective. A physical participant in a dialogue may *act the part of* or *represent* many logical parties. For example, a librarian might be responsible for both acquisitions policy and disposals.

Each party has a commitment store which holds its commitments within the dialogue. A commitment is the public engagement to a statement that restricts freedom of action. A commitment to a statement is, in effect, holding yourself out as liable for the consequences of that statement. As a concept commitment needs to be carefully distinguished from epistemic notions such as belief, which are essentially private and which we exclude from our model both on philosophical and technical grounds.

The contents of the commitment store of each party changes as the dialogue progresses. A party can read from any store. The only way it can alter the commitments of any party, including its own, is through participation in the dialogue. A specification is the pool of commitments that result from such a dialogue.

Each party has a working area which is its private *sketch-pad* or database. It contains the internal or working statements which do not have the full status of public engagement. No other party can directly access the working area without permission nor is there any obligation for the party to maintain its consistency. (Notationally: $WA_a \Rightarrow$ Party_A's working area and $WA_b \Rightarrow$ Party_B's working area.)

The dialogue scheme is presented in terms of three important constructs:

(i) 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. 'Since our locution modifiers must be applied to statements, no problem of the iteration or interaction of locution modifiers arise here' (Mackenzie 1985). 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).

Mackenzie does not have a specific locution modifier named justification. What Mackenzie calls 'an assertion given as a ground answer for a challenge' and what we call 'justification' above are one and the same thing. We believe that by isolating this particular instance of 'assertion' - and calling it 'justification' - it is easier to represent the nature of the verbal exchanges that occur in a cooperative environment.

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

Mackenzie (85) does not grant denial the status of a locution modifier, because denial only occurs after questions. For reasons that are similar to those given to justification above, we are treating it as a locution modifier. Mackenzie does not provide us with a commitment rule for it, and this will be necessary for the example below.

Resolution Demands, to be read as "Resolve this set of statements against this specific statement", notationally **resolve**(Set_of_Statements/Statement).

Resolution demand is a special locution modifier that is applied to a set of statements and not to a single one. Mackenzie (85) presents it in two different forms: The first one R'T where 'T' is a set of statements and a second one R'T/s where 's' is a statement that seen together with 'T', indicates the existence of some sort of inconsistency inside the commitment store. We have stuck to the latter, because in a cooperative environment fleshing out the contradictions helps the parties to reach a consensus.

(ii) Dialogue Events

Represented by a triple of the form <Stage, Party, Locution>. Stage marks the progress of the dialogue: stage, stage+1 and so on. Party indicates the current speaker. A dialogue is a sequence of such events.

(iii) Commitments

Represented C_{Party}(Stage, Statements*). Statement* is either Statement and/or **why**(Statement). This remark is important because of commitment rule Challenge below. Each party has a commitment store which holds its commitments within the dialogue. (Notationally: $C_s \Rightarrow$ Speaker's commitment store and $C_h \Rightarrow$ Hearer's commitment store.)

These constructs are used in the rules of the scheme which are divided into three subsets:

(i) Dialogue rules

Establish the *etiquette* or rules governing the legitimate shape of the interaction; they provide a way of maintaining a *legal* dialogue. For example:

<u>Dialogue rule Question</u>:

No legal dialogue of length stage+1 contains an event

<stage-1,hearer,**questions**(Statement)>

unless it also contains an event

<stage,speaker,**asserts**(Statement)> or

<stage,speaker,**withdraws**(Statement)> or

<stage,speaker,**denies**(Statement)>.

After the questioning of a statement (**questions**(Statement)), the next event must be either the assertion (confirmation) of that statement, its withdrawal or its denial (**asserts**(Statement), **withdraws**(Statement) or **denies**(Statement)).

(ii) Commitment rules

Set out how locutions affect the commitment store of each party. For example:

Commitment rule Withdrawal:

After <stage,speaker,withdraws(Statement)>

C_s(stage+1,Statements*) = C_s(stage,Statements*) – {Statement}

C_h(stage+1,Statements*) = C_h(stage,Statements*)

After a withdrawal the statement is removed from the speaker's commitment store. The hearer's store remains unchanged.

(iii) Argument forms

Define, syntactically, the form of reasoning permissible within the dialogue and common to its participants. Our presentation of DC primarily involves *modus ponens*, though addition of other schemas to fit various logical tastes is possible. The argument form mechanism for *modus ponens* is embedded in the rule below:

Commitment rule Justification:

After <stage,speaker, justifies(AnotherStatement)>

where the preceding dialogue event was <stage–1,hearer,why(Statement)>

C_s(stage+1,Statements*) = C_s(stage,Statements*) ∪ {AnotherStatement, AnotherStatement→Statement}

C_h(stage+1,Statements*) = C_h(stage,Statements*) ∪ {AnotherStatement, AnotherStatement→Statement}

After a justification (AnotherStatement) which occurs as a ground answer to a challenge (why(Statement)) both views are committed to the ground answer (AnotherStatement) and to the conditional (AnotherStatement→Statement).

As can be seen, not only was {AnotherStatement} added to both stores, as would be expected, but also {AnotherStatement→Statement}. If we take {AnotherStatement, AnotherStatement→Statement} and apply the *modus ponens* rule to it, we deduce {Statement} - exactly what was originally challenged.

The following examples show how the framework can be applied to represent and to register the type of short conversations that are common in the office context.

2.3.1. Example A

This example illustrates the way people get committed to situations - a meeting in this case - as a result of interacting with other people. Initially the commitment stores are empty. The speaker (Party_B) asserts the statement, in this case *meeting at 10 am tomorrow*, and so by:

Commitment rule Assertion:

After <stage,speaker,asserts(Statement)>

C_s(stage+1,Statements*) = C_s(stage,Statements*) ∪ {Statement}

C_h(stage+1,Statements*) = C_h(stage,Statements*) ∪ {Statement}

After the assertion of a statement the speaker and the hearer are both obliged to place that statement in their commitment store.

The rule Assertion defines an important feature of this dialogue scheme. A party is committed to anything stated by another party. The placing of the statement inside the hearer's commitment store is a way to force him to react to it. If he agrees with it, he becomes committed to it and to its immediate consequences. The notion of immediacy can be partly exemplified by the derivation of 's' from 'T' using modus ponens only once. If he does not agrees with it and wants to remove it from his commitment store, this can only be done by a subsequent withdrawal or challenge.

The resulting commitment stores are C_a(1,(meeting at 10 am tommorrow)) and C_b(1,(meeting at 10 am tommorrow)). Both Party_A and Party_B are committed to and must answer for any immediate consequences of this commitment and other commitments added in a similar manner.

2.3.2. Example B

The working areas are:

WA_a:

project_manager→member_of_project
system_analyst→member_of_project
programmer→member_of_project
¬member_of_project→member_of_support
¬member_of_support→member_of_project
¬project_manager and ¬system_analyst→programmer
¬member_of_support and ¬member_of_project→burocrat
¬burocrat and ¬member_of_support→technical_meeting

WA_b:

program→code
specification→program
member_of_project→specification
project_manager→report
system_analyst→report
report→specification

Let us now consider a slightly more complicated example illustrating the preparation of a technical meeting. We start at a point some way into a set of dialogues with C_a(n,(project_manager→code)) - Party_A's commitment store in the previous stage - containing a commitment to *project_manager→code* [*project_manager is responsible for delivering code*]. Party_A, the speaker and initiator of this part of the dialogue, asks something along the lines of *Why is it to be supposed that the project_manager is responsible for delivering code?*

It should be noted that in this setting asking why is a demand for evidence, not for an explanation. So by:

<u>Commitment rule Challenge</u>:

After <stage,speaker,why(Statement)>

C_s(stage+1,Statements*) = C_s(stage,Statements*) ∪ {**why**(Statement)}

C_h(stage+1,Statements*) = C_h(stage,Statements*) ∪ {Statement}

After a challenge the hearer adds the challenged statement to its own commitment store and the speaker removes the statement from its commitment store, replacing it by the challenge itself. This is part of the mechanism that is necessary to avoid the problem of circularity. (*Why is the book on loan?*, *Because it is out of the library!*, *Why is it out of the library?*, *Because it is on loan!* and so on).

The resulting commitent stores are:

C_a(n+1,(why(project_manager→code)));

C_b(n+1,(project_manager→code)).

Party_B replies to maintain dialogue legality as indicated by:

Dialogue rule Turn-Taking:

No legal dialogue contains an event <stage,given_party, locution>

if it also contains an event <stage-1,given_party, locution> or

if locution is not properly constructed.

Each party contributes a locution at a time. Each locution must be well formed; that is an assertion, question, withdrawal, etc.

Next , *program* is given as a justification for the challenge. It is taken from the statement contained in the working area that *program→code*. The resulting commitment stores are derived according to the rule below:

Commitment rule JustIMP: (Justification for the Challenge of an Implication)

After <stage,speaker, **justIMP**(AnotherStatement)>

where the preceding dialogue event was

<stage-1,hearer,**why**(Statement1→Statement2)>

C_s(stage+1,Statements*) = C_s(stage,Statements*) ∪ {AnotherStatement, AnotherStatement→Statement2}

C_h(stage+1,Statements*) = C_h(stage,Statements*) ∪ {AnotherStatement, AnotherStatement→Statement2}

After an assertion (AnotherStatement) which occurs as a reply to a challenge (why(Statement1→Statement2)) both parties are committed to the reply (AnotherStatement) and to the conditional (AnotherStatement→Statement2).

In effect, the rule above treats 'Statement 1' as an assumption, and handles the reply to challenge, as a justification for 'Statement2'. Mackenzie (85) does not have this locution modifier. Our inspiration comes from the way Gabbay (85) handles conditionals.

The resulting commitment stores are:

C_a(n+2,(why(project_manager→code) ∧ program ∧ program→code));

C_b(n+2,(project_manager→code ∧ program ∧ program→code)).

The continuation of this dialogue generates the argumentation steps - making the premises available - required to show why a project_manager is responsible for delivering the code.

Party_A can challenge either *program→code* or *program*. It chooses to chalenge *program* because what is there to be chalenged is *project_manager→program [Why is the project_manager responsible for the program?]*. Party_A is trying to establish the chain of reasoning that justifies *program→code* . This illustrates Party_A's strategy, which is a topic that we are not going to deal within this article.

Party_B checks its working area and justifies(specification).

The resulting commitment stores after these pair of dialogue events are:

C_a(n+4,(why(project_manager→code) ∧ program→code ∧ why(program) ∧ specification ∧ specification→program));

C_b(n+4,(project_manager→code ∧ program ∧ program→code ∧ specification ∧ specification→program)).

Party_A can challenge either *specification→program* or *specification* It chooses to challenge *specification* because what is there to be challenged is *project_manager→specification [Why is the project_manager responsible for the specification?]*.

Party_B consults his working area and justifies(report).

The resulting commitment stores after these pair of dialogue events are:

C_a(n+6,(why(project_manager→code) ∧ program→code ∧ why(program) ∧ specification→program ∧ why(specification) ∧ report ∧ report→specification));

C_b(n+6,(project_manager→code ∧ program ∧ program→code ∧ specification ∧ specification→program ∧ report ∧ report→specification)).

Party_A can challenge either *report→specification* or *report* . It chooses to challenge *report* because what is there to be challenged is *project_manager→report [Why is the project_manager responsible for the report]*.

Party_B checks its working area and justifies(project_manager).

The resulting commitment stores after these pair of dialogue events are:

C_a(n+8,(**why**(project_manager→code) ∧ program→code ∧ **why**(program) ∧ specification→program ∧ **why**(specification) ∧ report→specification ∧ **why**(report) ∧ project_manager ∧ project_manager→report));

C_b(n+8,(project_manager→code ∧ program ∧ program→code ∧ specification ∧ specification→program ∧ report ∧ report→specification ∧ project_manager ∧ project_manager→report)).

This example shows how this framework provides us with the means for representing the challenge-justify mechanism - making the premisses available - that is common in cooperative work.

2.3.3. Example C

The working areas are:

WA_a:

project_manager→member_of_project
system_analyst→member_of_project
programmer→member_of_project
¬member_of_project→member_of_support
¬member_of_support→member_of_project
¬project_manager and ¬system_analyst→programmer
¬member_of_support and ¬member_of_project→burocrat
¬burocrat and ¬member_of_support→technical_meeting
report→specification
¬report
system_analyst→report
system_analyst

WA_b:

package→code
program→code
specification→program
report→specification
¬specification→package
member_of_project→specification
project_manager→report
system_analyst→specification

The final example we will consider illustrates a number of features including progressive verification of one party with respect to another. By looking at WA_b above it should be easy to spot the inconsistency which has been introduced (*¬report*) [*It is not the case that there is a report*], as a result of which notice the inconsistency that may possible arise between Party_B (which is working on the basis of [*system_analyst→specification*]) and Party_A.

Party_B challenges *system_analyst→specification.*

Party_A consults his working area and **justifies**(report) by matching with the implication *report→specification.*

The resulting commitment stores after these pair of dialogue events are:

C_a(n+2,(system_analyst→specification ∧ report ∧ report→specification);

C_b(n+2,(why(system_analyst→specification) ∧ report ∧ report→specification).

Party_B now challenges *report* and Party_A withdraws it being unable to deny it due to Dialogue rule Challenge which, substantially abbreviated, states:

Dialogue rule Challenge:

The reply to a challenged statement must be the withdrawal of the statement or it must be the resolution demand of an immediate consequence conditional of the statement whose consequent is the challenged statement and whose antecedent is a conjunction of statements to which the challenger is committed or it must be a statement to whose challenge the challenger is not committed.

The resulting commitment stores after these pair of dialogue events are:

C_a(n+4,(system_analyst→specification ∧ report→specification);

C_b(n+4,(why(system_analyst→specification) ∧ report→specification ∧ why(report)).

Next Party_B will force the construction of a chain of reasoning that will help both parties to verify if they both have the same understanding of the meeting's preparation. Some straightforward question-answering follows with the results determined by Commitment rule Assertion above and:

Commitment rule Question:

After <stage,speaker,questions(Statement)>

C_s(stage+1,Statements*) = C_s(stage,Statements*)

C_h(stage+1,Statements*) = C_h(stage,Statements*)

Questions do not affect commitment stores.

Party_B **questions**(system_analyst→report) and Party_A answers yes by replying **asserts**(system_analyst→report).

The resulting commitment stores after these pair of dialogue events are:

C_a(n+6,(system_analyst→specification ∧ report→specification ∧ system_analyst→report);

C_b(n+6,(why(system_analyst→specification) ∧ report→specification ∧ why(report) ∧ system_analyst→report).

Party_B **questions**(system_analyst) and Party_A answers yes by replying **asserts**(system_analyst).

The resulting commitment stores after these pair of dialogue events are:

C_a(n+8,(system_analyst→specification ∧ report→specification ∧ system_analyst→report ∧ system_analyst);

C_b(n+8,(why(system_analyst→specification) ∧ report→specification ∧ why(report) ∧ system_analyst→report ∧ system_analyst).

Party_B questions *report*, which he is free to do because *report* is not in C_b. Party_A denies it, as a result of an inconsistency in its working area.

Commitment rule Denial:

After <stage,speaker,denies(Statement)>

$C_s(stage+1, Statements^*) = C_s(stage, Statements^*) \cup \{\neg Statement\}$

$C_h(stage+1, Statements^*) = C_h(stage, Statements^*) \cup \{\neg Statement\}$

If a denial of a statement has been made then the speaker and the hearer are obliged to place the negation of the statement in their commitment store.

Mackenzie (85) does not provide a commitment rule for denial, because he does not grant denial the status of a locution modifier as we said before.

The resulting commitment stores after these pair of dialogue events are:

C_a(n+10,(system_analyst→specification ∧ report→specification ∧ system_analyst→report ∧ system_analyst ∧ ¬report);

C_b(n+10,(**why**(system_analyst→specification) ∧ report→specification ∧ **why**(report) ∧ system_analyst→report ∧ system_analyst ∧ ¬report).

Party_B immediately demands that Party_a, having denied an immediate consequence of its commitments **resolve**(system_analyst ∧ system_analyst→report /¬report), which is now inconsistent (using modus ponens we have *{system_analyst, system_analyst→report}/report* which is inconsistent with *¬report*):

Commitment rule Resolution Demand:

After <stage,speaker,**resolve**(Set_of_Statements/Statement)>

$C_s(stage+1, Statements^*) = C_s(stage, Statements^*)$

$C_h(stage+1, Statements^*) = C_h(stage, Statements^*)$

Resolution demands do not affect the commitment stores.

Party_A withdraws its previous denial restoring consistency by adjusting its commitments according to Commitment rule Withdrawal given in our overview of the dialogue scheme.

Party_B follows suit by also withdrawing the inconsistency, which if not removed would now leave it liable to a resolution demand from Party_A, and so the dialogue concludes witha shared description and discovery of the *misunderstanding* hidden in WA_a.

The resulting commitment stores after these three final dialogue events are:

C_a(n+13,(system_analyst→specification ∧ report→specification ∧ system_analyst→report ∧ system_analyst);

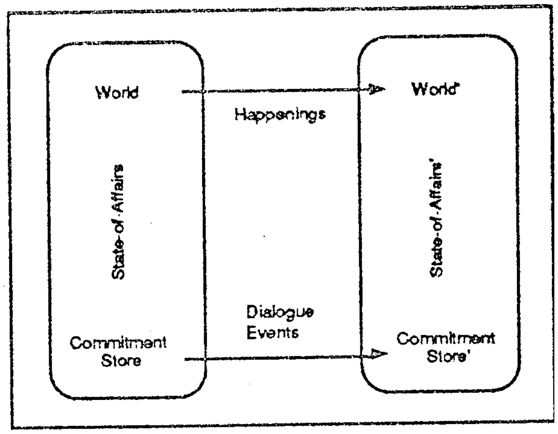
C_b(n+13,(**why**(system_analyst→specification) ∧ report→specification ∧ **why**(report) ∧ system_analyst→report ∧ system_analyst).

This example shows how this framework provides us with the means for representing the elucidation-retracting mechanism, which is a helpful way for sorting out inconsistencies - conflict resolution - that are bound to happen in cooperative working environment.

We have previously used this framework to capture in a very idealised way, the conventional setting of requirements specification in which clients and systems analysts sit around a table - the clients explaining the requirements, waving documents in the air and occasionally arguing among themselves while the developers ask guiding questions, seek clarification, point out inconsistencies and raise unanticipated consequences. In (Finkelstein & Fuks 89) the examples were based on a small case study concerning description of an automated travel ticketing system. In that case study various statements about travel and travel discounts are distributed between working areas of two parties. In (Finkelstein & Fuks 90) the examples were based in the specification of software to support the preparation and assembly of user manuals for a range of production tools.

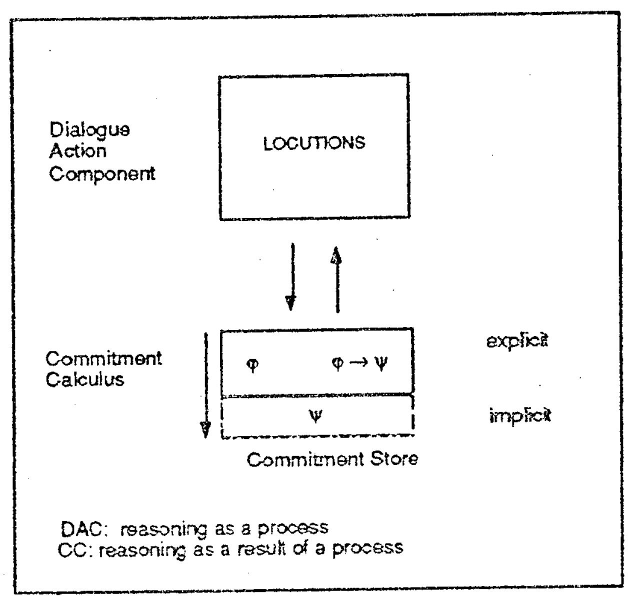
2.4. A Framework for Dialogue Representation Systems

Hamblin (87) describes the state transformations occurring in the environment (from a party's point of view), being caused either by *deeds* - where party is the active agent - or *happenings* - where party is a passive observer of the changes that are taking place. We substitute Dialogue Events for Deeds and divide the state of affairs into World - a global database - and Commitment Stores - a local database.



We propose a framework for dialogue representation systems - ACCORD, which is only concerned with the dialogic part of the model. It consists of a Commitment Calculus (CC) and a Dialogue Action Component (DAC). While the former defines the relationship between commitments inside individual commitment stores, the latter concerns itself with the proper ordering of locutions and their effects on the commitment stores.

The figure below illustrates the relationship between CC and DAC. While DAC is busy with the dialogue process, thus managing the insertion/deletion of commitments in the commitment stores, CC reasons with the result of this process. CC never generates commitments, although it might seem to be doing this while inferring the consequences - implicit commitments - of the commitments - explicit commitments - already in the commitment stores.



3. Commitment

We have been using the notion of commitment in this article without plunging deep into its nature. Hamblin's and Mackenzie's formalisms did not provide us with a structured way to deal with the commitments inside the commitment stores. More recently, the notion of commitment has been used in computing (Winograd 88), (Koo & Wiederhold 88) and (Bond 90) for modelling interaction between agents. However, no associated calculus was provided to infer the consequences of having these commitments.

In this section the topic of commitment is discussed The few attempts to define a commitment logic have always characterised it as an offshoot of deontic logic (von Wright 51), (Rescher 58), (Anderson 59), (Castaneda 59) and (Hintikka 71) In these schemes the notions of permission, obligation and commitment are normally prefixed by some sort of adjective, for example: moral, absolute, conditional and derived. It is necessary to understand the way these concepts are knitted together before we can achieve our objective of an *independent* commitment logic.

3.1. Deontic Commitment

Much effort towards a formal understanding of commitment has been devoted to deontic logics, logics of permission and obligation. However, there are important differences between these two notions. While obligations apparently ask for immediate responses (actions), there is a sense of delayed effect embedded in commitments (behaviour). For example, you are obliged to pay taxes, and you have a commitment to being a good citizen. In other words, one does not have a commitment to perform actions, but to the achievement of states of affairs. If one has a commitment to a certain state of affairs, then either one observes the ocurrence of that state of affairs or retains one's commitment to it. Eventually that commitment will have to be satisfied. Commitments are of a passive nature while obligations are of an active kind. People should behave coherently with their commitments and with the information available to them.

The *essence* of commitment for us, is that if one has a commitment to a certain state of affairs, one is answerable for the achievement of this state of affairs. Unfortunately our notion of commitment says nothing about the party's ability to bring it about.

Next we review the notion of commitment trying to provide a calculus for dealing with it. This time we approach the notion of commitment having in mind that if one is committed to something, one is aware of this thing - if one has a commitment to do something then one is aware of the object of one's commitment. At least three important aspects of commitment will be imported from CL (Fuks, Ryan & Sadler 89) into the new calculus: its three-valuedness (having, not having and lacking commitments), allowing commitments to be weakened, and not allowing commitments to be strenghened.

3.2. Revisiting Commitment

Having a commitment to bring about a state of affairs (s-o-a), presses one to behave in a manner consistent with the achievement of that s-o-a. It is important to see that one is neither alone nor aware of everything that is taking place on the planet. Thus, we could go further saying that having a commitment establishes a family of ways for a party to behave, in a manner consistent with the information available and with the role that that party performs in its community.

There is a relationship between being in the situation of having a commitment and the behaviour pattern that follows from it. In that sense, commitment reminds us of belief. One behaves according to one's beliefs. But belief, especially in the sense of faith, is not an operational notion. One can hide one's beliefs or truly need not be capable of expressing them. Commitments should not only be expressible but also available for public examination. It is easier to check one's commitments against one's behaviour this way. Moreover, a wider spectrum of agents are capable of undertaking commitments, human beings, corporations and governments while beliefs are better used when applied to people.

From the above, it is implicit that there is no commitment without being conscious of the objects of one's commitment. It also implies that there is a difference between not having a commitment and its absence. While the former involves cognizance and rejection of some data, not much can be said about the latter. This three-valued aspect of commitment influences the way that we design the commitment stores in the next section.

Another important characteristic of commitments - imported from the work developed on deontic commitment - that we wish to capture in the calculus, is that while it is fair to weaken commitments it is forbidden to strengthen them.

Given that one is committed to achieve some desired s-o-a, one adjusts one's behaviour pattern to suit one's commitment. In other words, one's commitments impose constraints on one's behaviour. One way of describing one's behaviour is by looking at one's structure of permissions and obligations - do's and don't's. A commitment is an engagement or involvement that restricts freedom of action. If one is committed to something, one should stick to it.

One is tempted to join the idea of having a commitment with that of having a goal, because in either case, success will be equated with the realisation of that envisaged s-o-a (goal or commitment). But from a computing perspective, goal is the query that one makes for a database; or relaxing it a bit, it is just the final state that is to be achieved. Moreover, associating commitment with goal, we lose the awareness aspect of the former (it is not clear if after achieving a goal whether one should keep conscius of the object of one's goal). Having said that, we feel free to connect commitment with goal, whenever it sheds some light on our understanding.

Finally, advancing the use of commitment, we are not neglecting other concepts like intention, belief and knowledge. They all play an important part in the process of reasoning. But at the current stage of this work, we see commitment as a pragmatic *amalgam* for the interaction aspects of all of them. We hope in future work we will be able to interconnect all these concepts by means of specific axioms and/or rules.

3.3. State-of-Affairs

About general s-o-a Pollock (84) says: "In many ways, states of affairs resemble propositions. In particular, they are truth bearers of a sort. States of affairs are not literally true or false, but obtaining and not obtaining are truth-like properties". He makes a distinction between transient s-o-a (*being a patriot*) and nontransient s-o-a (*being a patriot during WWII*).

One can become committed to either sort of s-o-a (transient or nontransient). The latter fits better within our notion of commitment where one retains one's commitment even after realising it. Is one going to keep one's commitments forever? No, provided that there is a mechanism allowing for their retraction. The framework that we develop in the following sections will cater for this.

One can become committed to actions, for actions promote the achievement of s-o-a's - from now on we use either form. Having a commitment to paying my debts, boils down to having a commitment to transferring money from my account to my lender's account. The obtaining of this s-o-a could be asserted after checking that my lender's account was increased by the same amount of money that was subtracted from mine, securing that it corresponds to the debt's amount. Commitment to conditionals is acceptable for a conditional is a constraint on a s-o-a.

3.4. Conversation, Commitment and Speech Acts

It is not only through utterances of the form *I am committed to* or *I have a commitment to* that one finds oneself having commitments. For example, when your auntie rings you saying that she is coming for dinner on Saturday evening, you had better say a clear no otherwise you are committing yourself to that fact. Moreover, our parties are far from being omnipotent. They are fallible and unable to deal with all the matters that comprise their everyday life. But still they accept and reject statements about the world all the time. Accepting and rejecting amount to engaging and disengaging. Listening to the weatherman saying that it is going to snow today, gives you precious data for deciding which means of transportation to use. Also when you read the memo that says that the corporation - your employer - is expanding its activities, even if that expansion has nothing to do with the role that you perform in the company, you are still engaged in the expansion although you are not going to play an active part in it.

The notion of commitment also plays an important part in speech act theory, for example (Searle & Vanderveken 87), (van Eemeren & Grootendorst 84) and (Searle 79). Our understanding of commitment does not contradict the one advanced by speech act theory. We are only dealing with a part of it where statements are left unanalysed.

4. Commitment Calculus

A Commitment Calculus for dealing with the consequences of commitments inside the commitment stores is developed in this section. Commitment is used to represent the parties' range of information in the dialogue. Having a commitment establishes a family of ways for a party to behave in a manner consistent with the information available and with that party's role in its community.

Each party has a commitment store, where the unfolding of the dialogue is recorded from its point of view. Commitment stores are open to general observation, enabling the parties *to anticipate the actions of the others and coordinate them with their own* (Winograd 88).

But what are the consequences of having commitments? How can one determine if one's or the other's commitment store is inconsistent? Although becoming inconsistent does not mean the end of times, it may put one in a vulnerable position.

These and other questions were the motivation for designing such a calculus. We are not interested in giving a definitive meaning to the notion of commitment, but solely providing a framework within which to operate it. Our treatment of commitment and therefore its calculus is based on the discussion developed in the previous section.

For the sake of clarity, each party will not only have a commitment store for its positive and negative commitments - 'C' commitment store (called C) - but also a 'D' commitment store (called D) as a place for the statements which the party is not committed to. We decided to design the stores in this way, because it is very revealing about the three-valued nature of our notion of commitment. The choice for the letter 'D' is because it is the next letter in our alphabet, and not to make the word *dis*commitment, which fits better for the idea of lack of commitment.

4.1. Language

Definition: *Language* L'

Let L' be the propositional language:

$p, q, .. \in$ L' (atomic propositions)

If $\varphi, \psi \in$ L' then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in$ L'.

Notation: Throughout thes rest of this section we are going to use

φ, ψ to range over propositions in L' and

Γ, Δ to range over sets of propositions in L'.

Definition: *Language* CL

Let CL be the commitment propositional language:

If $\varphi \in L'$ then φ, $C\varphi$, $D\psi \in CL$.

Definition: $C\Gamma = \{C\varphi \mid \varphi \in \Gamma\}$

Definition: $D\Gamma = \{D\varphi \mid \varphi \in \Gamma\}$

Definition: $\neg\Gamma = \{\neg\varphi \mid \varphi \in \Gamma\}$

Notation: Throughout thes rest of this section we are going to use

φ, ψ to range over propositions in CL ($C\varphi$,... $D\psi$) and

Γ, Δ to range over sets of propositions in CL ($C\Gamma$,... $D\Delta$).

4.2. Membership Semantics

In this section we develop a membership semantics for the commitment calculus. (Gardenfors 88) develops a membership semantics - one in which truth in an epistemic state is determined by membership of a set - for theory revision which is not relevant at the current stage of this work.

In our semantics states-of-affairs are described by propositions.The truth and falsity of formulas is defined by their membership status in the commitment stores. For each formula φ we check if it belongs to C or D. 'φ' is undefined if it does not belong to either commitment store.

$C\wedge$		$(\varphi \wedge \psi) \in C$	iff	$\varphi \in C$ and $\psi \in C$
$C\vee$	if			
		$\varphi \in C$	then	$(\varphi \vee \psi) \in C$
		$\psi \in C$	then	$(\varphi \vee \psi) \in C$
	if	$(\varphi \vee \psi) \in C$	and	
		$\neg\psi \in C$	then	$\varphi \in C$
		$\neg\varphi \in C$	then	$\psi \in C$
		$\psi \in D$	then	$\neg\varphi \in D$
		$\varphi \in D$	then	$\neg\psi \in D$
$C\neg\wedge$	if			
		$\neg\varphi \in C$	then	$\neg(\varphi \wedge \psi) \in C$
		$\neg\psi \in C$	then	$\neg(\varphi \wedge \psi) \in C$
	if	$\neg(\varphi \wedge \psi) \in C$	and	
		$\varphi \in C$	then	$\neg\psi \in C$
		$\psi \in C$	then	$\neg\varphi \in C$
$C\neg\vee$		$\neg(\varphi \vee \psi) \in C$	iff	$\neg\varphi \in C$ and $\neg\psi \in C$

$D_\wedge$	if			
		$\varphi \in D$	then	$(\varphi \wedge \psi) \in D$
		$\psi \in D$	then	$(\varphi \wedge \psi) \in D$
	if	$(\varphi \wedge \psi) \in D$	and	
		$\varphi \in C$	then	$\psi \in D$
		$\psi \in C$	then	$\varphi \in D$
		$\neg\varphi \in D$	then	$\psi \in D$
		$\neg\psi \in D$	then	$\varphi \in D$
$D_\vee$		$(\varphi \vee \psi) \in D$	iff	$\varphi \in D$ and $\psi \in D$
$D_{\neg\wedge}$		$\neg(\varphi \wedge \psi) \in D$	iff	$\neg\varphi \in D$ and $\neg\psi \in D$
$D_{\neg\vee}$	if			
		$\neg\varphi \in D$	then	$\neg(\varphi \vee \psi) \in D$
		$\neg\psi \in D$	then	$\neg(\varphi \vee \psi) \in D$
	if	$\neg(\varphi \vee \psi) \in D$	and	
		$\neg\varphi \in C$	then	$\neg\psi \in D$
		$\neg\psi \in C$	then	$\neg\varphi \in D$
$C_\rightarrow$	if	$(\varphi \rightarrow \psi) \in C$	and	
		$\varphi \in C$	then	$\psi \in C$
		$\neg\psi \in C$	then	$\neg\varphi \in C$
$D_\rightarrow$	if			
		$\varphi \in C$ and $\psi \in D$	then	$(\varphi \rightarrow \psi) \in D$
		$\neg\varphi \in D$ and $\neg\psi \in C$	then	$(\varphi \rightarrow \psi) \in D$
$D_\neg$	if	$\varphi \in C$	then	$\neg\varphi \in D$
$C_\neg$	if	$\neg\varphi \in C$	then	$\varphi \in D$

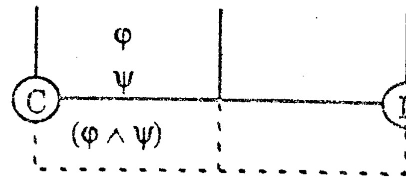
4.3. Rules

From this semantics we can read off a set of rules. These rules envisage consistency preservation inside the commitment stores. Below we present the rules in a different order from that given in the semantics. Here we are concerned with showing when two sets of rules behave like duals. In the cases that duality is not attained, we point out the cause. Most of the rules are accompanied by additional text, to explain and guide the reader throughout this section. We also present some of the rules that we do not have in the calculus, giving the reasons for not having them.

C∧

$$\frac{C\varphi,\ C\psi}{C(\varphi \wedge \psi)}$$

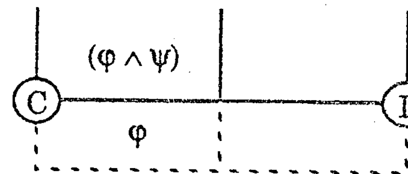
C∧I



Rule C∧I says that if one is committed to φ and committed to ψ, then one is committed to φ and ψ.

$$\frac{C(\varphi \wedge \psi)}{C\varphi}$$

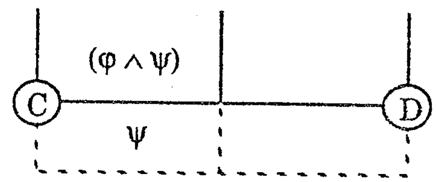
C∧E1



If I'm committed to being *a good citizen* and *a parent*, then I'm committed to being *a good citizen*.

$$\frac{C(\varphi \wedge \psi)}{C\psi}$$

C∧E2

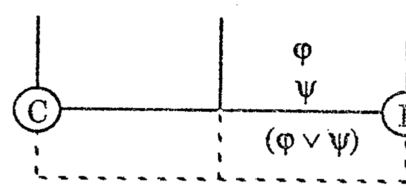


If I'm committed to being *a good citizen* and *a parent*, then I'm committed to being *a parent*.

D∨

$$\frac{D\varphi,\ D\psi}{D(\varphi \vee \psi)}$$

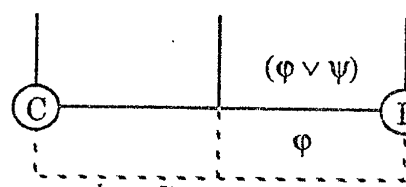
D∨I



Rule D∨I says that if one is not committed to φ and not committed to ψ, then one is not committed to either φ or ψ.

$$\frac{D(\varphi \vee \psi)}{D\varphi}$$

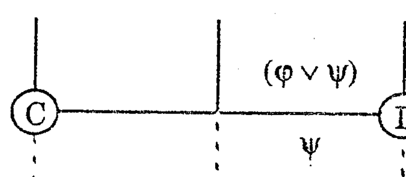
D∨E1



If I'm not committed to being either *a good citizen* or *a parent*, then I'm not committed to being *a good citizen*.

$$\frac{D(\varphi \vee \psi)}{D\psi}$$

D∨E2



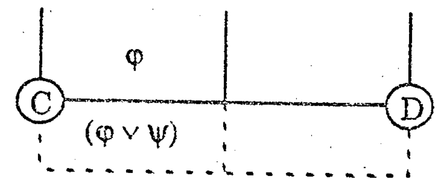
If I'm not committed to being either *a good citizen* or *a parent*, then I'm not committed to being *a parent*.

The two sets of rules C∧ and D∨ above are duals. This implies that the rules for dealing with conjunctions of commitments are similar to the rules for dealing with disjunctions of D-commitments.

C∨

$$\frac{C\varphi}{C(\varphi \vee \psi)}$$

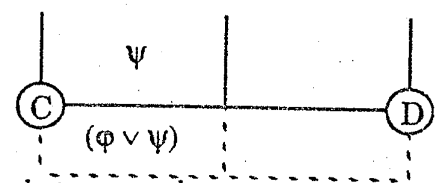
C∨I1



Rule C∨I1 says that if one is committed to φ, then one is committed to φ or ψ.

$$\frac{C\psi}{C(\varphi \vee \psi)}$$

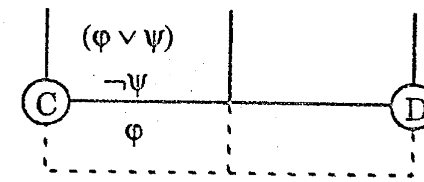
C∨I2



If I'm committed to being *a parent*, then I'm committed to being *a good citizen* or *a parent*.

$$\frac{C(\varphi \vee \psi), C\neg\psi}{C\varphi}$$

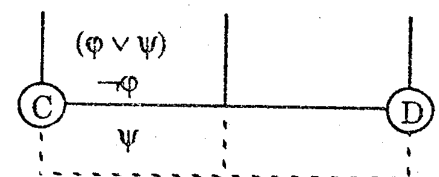
C∨E1



Rule C∨E1 says that if one is committed to φ or ψ and committed to not ψ, then one is committed to φ.

$$\frac{C(\varphi \vee \psi), C\neg\varphi}{C\psi}$$

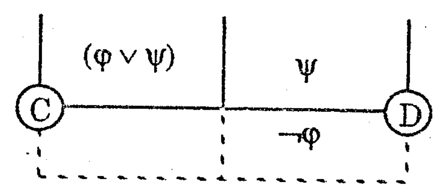
C∨E2



If I'm committed to being *a good citizen* or *a parent*, and I'm committed to not being *a good citizen*, then I'm committed to being *a parent*.

$$\frac{C(\varphi \vee \psi), D\psi}{D\neg\varphi}$$

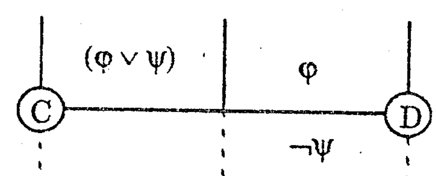
C∨E3



Rule C∨E3 says that if one is committed to φ or ψ and not committed to ψ, then one is not committed to not φ.

$$\frac{C(\varphi \vee \psi), D\varphi}{D\neg\psi}$$

C∨E4

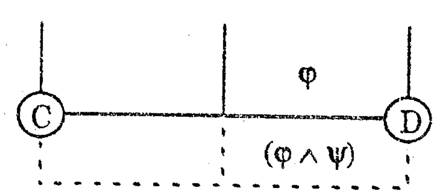


If I'm committed to being *a good citizen* or *a parent*, and I'm not committed to being *a good citizen*, then I'm not committed to not being *a parent*.

D∧

$$\frac{D\varphi}{D(\varphi \wedge \psi)}$$

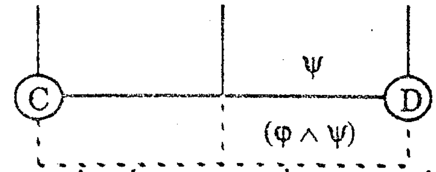
D∧I1



Rule D∧I1 says that if one is not committed to φ, then one is not committed to φ and ψ.

$$\frac{D\psi}{D(\varphi \wedge \psi)}$$

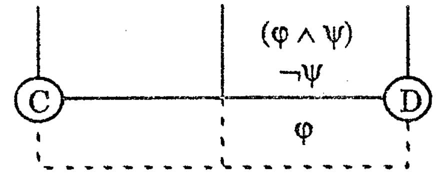
D∧I2



If I'm not committed to being *a parent*, then I'm not committed to being *a good citizen* and *a parent*.

$$\frac{D(\varphi \wedge \psi)\,,\,D\neg\psi}{D\varphi}$$

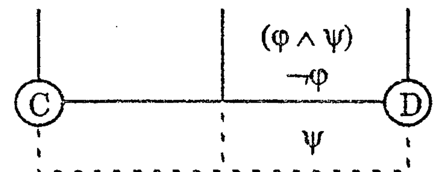
D∧E1



Rule D∧E1 says that if one is not committed to φ and ψ and not committed to not ψ, then one is not committed to φ.

$$\frac{D(\varphi \wedge \psi)\,,\,D\neg\varphi}{D\psi}$$

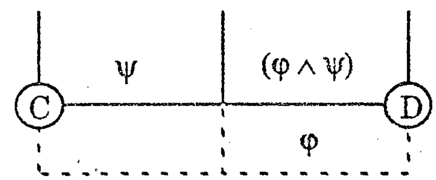
D∧E2



If I'm not committed to being *a good citizen* and *a parent*, and I'm not committed to not being a *good citizen*, then I'm not committed to being *a parent*.

$$\frac{D(\varphi \wedge \psi)\,,\,C\psi}{D\varphi}$$

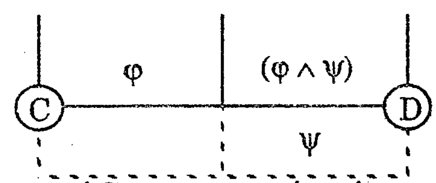
D∧E3



Rule D∧E3 says that if one is not committed to φ and ψ and committed to ψ, then one is not committed to φ.

$$\frac{D(\varphi \wedge \psi)\,,\,C\varphi}{D\psi}$$

D∧E4



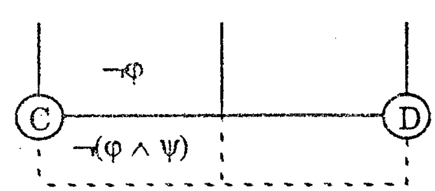
If I'm not committed to being *a good citizen* and *a parent*, and I'm committed to being a *good citizen*, then I'm not committed to being *a parent*.

The two sets of rules above C∨ and D∧ are not duals because of the differences between rules C∨E3 and D∧E3 (C∨E4 and D∧E4). The asymmetry between these rules is caused by the need to weaken rules D∧E. To have them as perfect duals, their conclusions have to be of the form C¬φ (or C¬ψ), which would put ourselves in the position of, from not being committed to something to becoming committed to its opposite. For example, if one is not committed to being a good citizen, this does not imply one being committed to being a bad citizen. The opposite is fine: if one is committed to being a bad citizen, then one is not committed to being a good citizen. Therefore, we transform C¬φ into Dφ (C¬φ in Dψ), which is justified by rule D¬I below.

C¬∧

$$\frac{C\neg\varphi}{C\neg(\varphi \wedge \psi)}$$

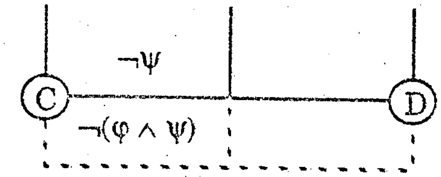
C¬∧I1



Rule C¬∧I1 says that if one is committed to not φ, then one is committed to not φ and ψ.

$$\frac{C\neg\psi}{C\neg(\varphi \wedge \psi)}$$

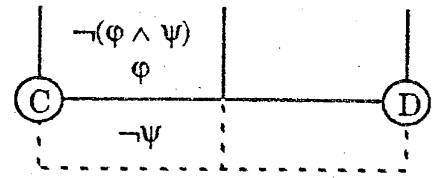
C¬∧I2



If I'm committed to not being *a parent*, then I'm committed to not being *a good citizen* and *a parent*.

$$\frac{C\neg(\varphi \wedge \psi),\ C\varphi}{C\neg\psi}$$

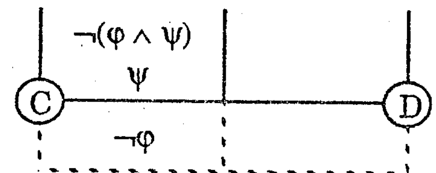
C¬∧E1



Rule C¬∧E1 says that if one is committed to not φ and ψ and committed to φ, then one is committed to not ψ.

$$\frac{C\neg(\varphi \wedge \psi),\ C\psi}{C\neg\varphi}$$

C¬∧E2

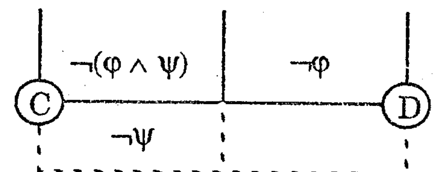


If I'm committed to not being *a parent* and *a good citizen*, and committed to being *a good citizen*, then I'm committed to not being *a parent*.

Next we indicate how rule *C¬∧E3* (and rule *C¬∧E4*) would look and show the reasons for not having it (them).

$$\frac{C\neg(\varphi \wedge \psi),\ D\neg\varphi}{C\neg\psi}$$

C¬∧E3



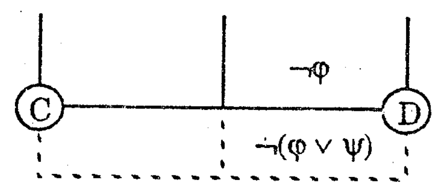
If I'm committed to not being *a parent* and *a good citizen*, and not committed to not being *a parent*, then I'm committed to not being *a good citizen*.

Although the reading that we gave to this rule makes it sound reasonable, there is some strenghening of commitments embedded in it that we want to avoid in the calculus. A formula in the D-store is influencing the conclusion of a formula in the C-store.

D¬∨

$$\frac{D\neg\varphi}{D\neg(\varphi \vee \psi)}$$

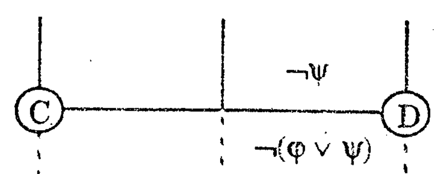
D¬∨I1



Rule D¬∨I1 says that if one is not committed to not φ, then one is not committed to neither φ nor ψ.

$$\frac{D\neg\psi}{D\neg(\varphi \vee \psi)}$$

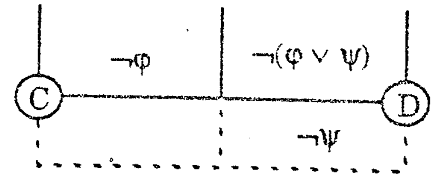
D¬∨I2



If I'm not committed to not being *a parent*, then I'm not committed to neither being *a good citizen* nor *a parent*.

$$\frac{D\neg(\varphi \vee \psi), C\neg\varphi}{D\neg\psi}$$

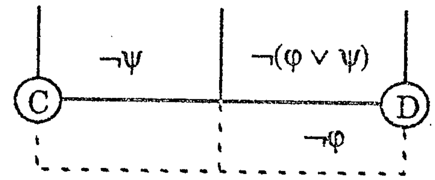
D¬∨E1



Rule D¬∨E1 says that if one is not committed to neither φ nor ψ and committed to not φ, then one is not committed to not ψ. This rule is not so straightforward. First imagine that you are committed not to go to the cinema or to the club. Next you decide to *relax* this commitment and withdraw it (which is similar to the effect caused by double negation in natural language). As a result of it, now you are not committed not to go to the cinema or to the club. Then you remember that you are committed not to go to the cinema because you simply hate black and white movies and there is no colour movie showing in town. You conclude that you are not committed not to go to the club, i.e. there is a chance that you might end up dancing to the sound of lambada but you are definitely not going to watch any movie this evening.

$$\frac{D\neg(\varphi \vee \psi), C\neg\psi}{D\neg\varphi}$$

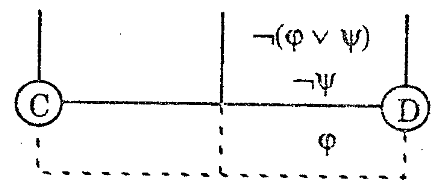
D¬∨E2



Next we indicate how rule *D¬∨E3* (and rule *D¬∨E4*) would look and show the reasons for not having it (them).

$$\frac{D\neg(\varphi \vee \psi), D\varphi}{D\neg\psi}$$

D¬∨E3



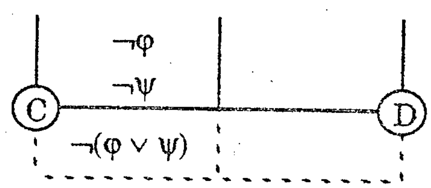
Now instead of being committed not to go to the cinema, you are simply not committed to go to the cinema. We feel that the conclusion is a bit too strong to follow from the premisses, therefore we do not want to have the rule *D¬∨E3* in our calculus.

The two sets of rules C¬∧ and D¬∨ above are asymmetric in the sense that both sets of rules lack their counterparts. Rules C¬∧E1 and C¬∧E2 miss their duals which are rules *D¬∨E3* and *D¬∨E4* respectively. The same happens to rule D¬∨E1 and rule D¬∨E2 which miss their dual rules *C¬∧E3* and *C¬∧E4* respectively.

C¬∨

$$\frac{C\neg\varphi, C\neg\psi}{C\neg(\varphi \vee \psi)}$$

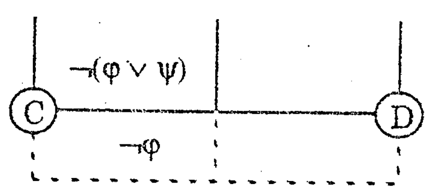
C¬∨I



Rule C¬∨I says that if one is committed to not φ and committed to not ψ, then one is committed to not φ or ψ.

$$\frac{C\neg(\varphi \vee \psi)}{C\neg\varphi}$$

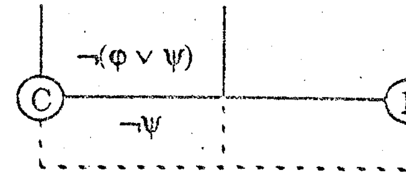
C¬∨E1



Rule C¬∨E1 says that if one committed to not being *a parent* or *a good citizen*, then one is committed to not being *a parent*.

$$\frac{C\neg(\varphi \vee \psi)}{C\neg\psi}$$

C¬∨E2

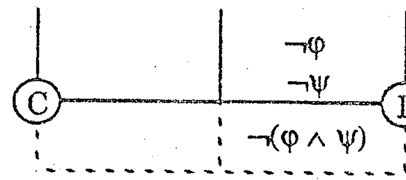


Rule C¬∨E1 says that if one committed to not being *a parent* or *a good citizen*, then one is committed to not being *a good citizen*.

D¬∧

$$\frac{D\neg\varphi,\ D\neg\psi}{D\neg(\varphi \wedge \psi)}$$

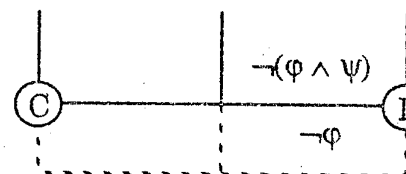
D¬∧I



Rule D¬∧I says that if one is not committed to not φ and not committed to not ψ, then one is not committed to not φ and ψ. If I'm not committed to not being *a good citizen* and not committed to not being *a parent*, then I'm not committed to not being *a good citizen* and *a parent*.

$$\frac{D\neg(\varphi \wedge \psi)}{D\neg\varphi}$$

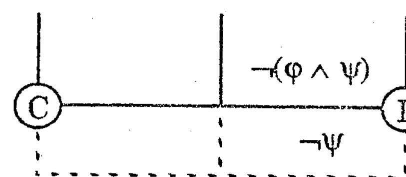
D¬∧E1



If I'm not committed to not being *a good citizen* and *a parent*, then I'm not committed to not being *a good citizen*.

$$\frac{D\neg(\varphi \wedge \psi)}{D\neg\psi}$$

D¬∧E2



If I'm not committed to not being *a good citizen* and *a parent*, then I'm not committed to not being *a parent*.

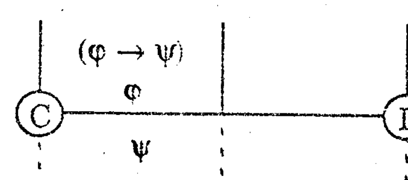
The two sets of rules above C¬∨ and D¬∧ are duals. This implies that the rules for dealing with disjunctions of negative commitments are similar to the rules for dealing with conjunctions of negative D-commitments.

The following rules give *identity* to this commitment calculus. They are not just *bureaucratic* symbol manipulation but actually *mean* something. Moreover, the absence of counterparts for some of the rules below indicate a lot about the interaction between this calculus and the dialogue action component developed in the next section.

C→

$$\frac{C(\varphi \rightarrow \psi),C\varphi}{C\psi}$$

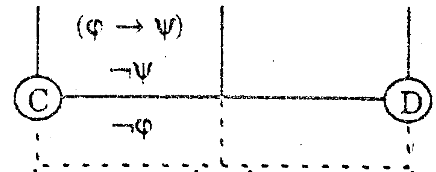
C→E1



If Joan has a commitment to the idea that being *a married person* implies being *a mother*, and Joan has a commitment to becoming *a married person*, then Joan has a commitment to becoming *a mother*.

$$\frac{C(\varphi \rightarrow \psi), C\neg\psi}{C\neg\varphi}$$

C→E2



If Joan has a commitment to the idea that being *a married person* implies being *a mother*, and Joan has a commitment to not becoming *a mother*, then Joan has a commitment to not becoming *a married person*.

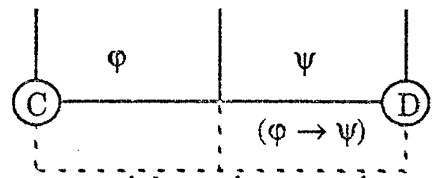
One would expect to find at this point a rule like C→I: a rule that would create conditional commitments from individual commitments. Nevertheless, conditionals will only appear in the commitment store as the result of the assertion of a conditional or as the result of a justification for a challenge. It is the *argumentation step*.

The following two rules, although introducing conditionals, do not contradict what we just said above. What they tell us is that given that one has the stated formulas in one's commitment store, *it would be dangerous* for one to become committed to the stated conditional, because one will be introducing a contradiction in one's store, hence becoming liable to a resolution demand from the other party to put the record straight.

D→

$$\frac{C\varphi, D\psi}{D(\varphi \rightarrow \psi)}$$

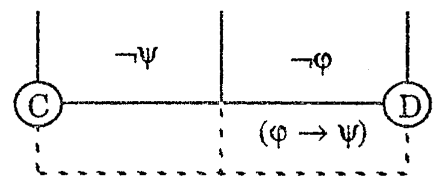
D→I1



If Joan has a commitment to becoming *a married person*, and Joan does not have a commitment to becoming *a mother*, then *it would be dangerous* for Joan to become committed to the idea that being *a married person* implies being *a mother*.

$$\frac{D\neg\varphi, C\neg\psi}{D(\varphi \rightarrow \psi)}$$

D→I2



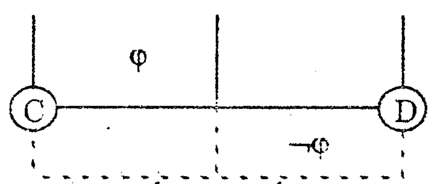
If Joan does not have a commitment to not becoming *a married person*, and Joan has a commitment to not becoming *a mother*, then *it would be dangerous* for Joan to become committed to the idea that being *a married person* implies being *a mother*.

The two rules below show how formulas can move from the C-store to the D-store. It is perceivable from them that this move - from 'C' to 'D' - implies the same sort of relaxation that occurs in natural language provided by the double negation.

D¬

$$\frac{C\varphi}{D\neg\varphi}$$

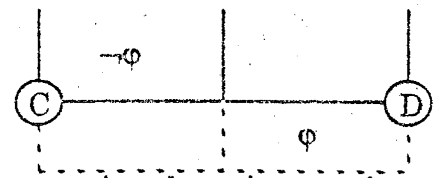
D¬I



If I'm committed to being *a good citizen*, then I'm not committed to not being *a good citizen*.

$C\neg$

$\frac{C\neg\varphi}{D\varphi}$ C¬E

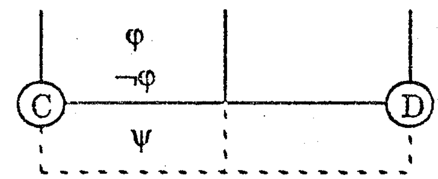


If Joan has a commitment to not becoming *a married person* then Joan does not have a commitment to becoming *a married person*.

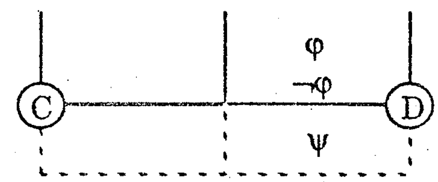
This rule captures an important valid formula $\{C\neg\varphi \rightarrow \neg C\varphi\}$ in the commitment logic (Fuks, Ryan & Sadler 89) that we wanted to preserve - allowing commitments to be weakened. We do not have rule $D\varphi / C\neg\varphi$. Thus, this preserves another important aspect of commitment logic - not allowing commitments to be strengthened - in which $\{\neg C\varphi \rightarrow C\neg\varphi\}$ is not a valid formula.

Finally we do not have the following rules in our calculus. This way we avoid the generation of undesirable implicit commitments from two contradictory explicit ones.

$\frac{C\varphi, C\neg\varphi}{C\psi}$ *C–Absurd*



$\frac{D\varphi, D\neg\varphi}{D\psi}$ *D–Absurd*



Based on the rules given above, a definition for a commitment calculus derivability ($\vdash$CC) is given below:

$\Gamma \vdash_{CC} \varphi$ iff

$\exists\ \varphi_1, \varphi_2, \ldots\ldots, \varphi_n$ where

$\varphi = \varphi_n$ and

for each i, i = 1 to n, $\varphi_i \in \Gamma$ or

was obtained through applications of CC rules.

4.4. Commitment-Consistency

The soundness of this calculus is a trivial result. Regarding completeness of this calculus, we feel that it falls outside the scope of this work. We are still investigating the subject and we do not feel that it is necessary to present a full account of the calculus at this stage. Nevertheless, the present version of the calculus is strong enough to deal with all the aspects brought up in this dissertation.

In this section we show that the calculus is commitment-consistent, to disband any fears that the calculus falls into some contradiction at the propositional level.

We say that this calculus is commitment-consistent, because starting from commitment-consistent commitment stores and using only the rules supplied by the calculus, we never derive both that we are committed to something and to the negation of that thing ($C\varphi$, $C\neg\varphi$).

Commitment-Consistency is defined based on the strongest case which is ($D\varphi$, $D\neg\varphi$). This is the case, because in CC premisses in the D-stores can never influence conclusions that appear in the D-stores, the opposite being permitted.

Definition: *Commitment-Consistency*

$C\Gamma \cup D\Delta$ is commitment-consistent iff it is not the case that
($C\Gamma$, $D\Delta \vdash_{cc} D\varphi$ and $C\Gamma$, $D\Delta \vdash_{cc} D\neg\varphi$).

Based on this definition of commitment-consistency we have two immediate results:

Immediate Result 1:

If $C\Gamma \cup D\Delta$ is commitment-consistent then it is not the case that
($C\Gamma$, $D\Delta \vdash_{cc} C\varphi$ and $C\Gamma$, $D\Delta \vdash_{cc} D\varphi$).

Proof: Suppose that $C\Gamma \cup D\Delta$ is commitment-consistent and it is the case that
$C\Gamma$, $D\Delta \vdash_{cc} C\varphi$ and $C\Gamma$, $D\Delta \vdash_{cc} D\varphi$.
Then applying rule $D\neg$ to $C\Gamma$, $D\Delta \vdash_{cc} C\varphi$ we have $C\Gamma$, $D\Delta \vdash_{cc} D\neg\varphi$
which contradicts the definition of commitment-consistency.

Immediate Result 2:

If $C\Gamma \cup D\Delta$ is commitment-consistent then it is not the case that
$C\Gamma$, $D\Delta \vdash_{cc} C\varphi$ and $C\Gamma$, $D\Delta \vdash_{cc} C\neg\varphi$.

Proof: Suppose that $C\Gamma \cup D\Delta$ is commitment-consistent and it i the case that
$C\Gamma$, $D\Delta \vdash_{cc} C\varphi$ and $C\Gamma$, $D\Delta \vdash_{cc} C\neg\varphi$.
Then applying rule $D\neg$ to $C\Gamma$, $D\Delta \vdash_{cc} C\varphi$ we have $C\Gamma$, $D\Delta \vdash_{cc} D\neg\varphi$,
And applying rule $C\neg$ to $C\Gamma$, $D\Delta \vdash_{cc} C\neg\varphi$ we have $C\Gamma$, $D\Delta \vdash_{cc} D\varphi$,
which contradicts the definition of commitment-consistency.

Using similar techniques to (Hughes & Cresswell 68) we show that CC is commitment-consistent.

Theorem: If $C\Gamma$, $D\Delta$ is not commitment-consistent
Then Γ, $\neg\Delta$ is inconsistent in PC (propositional calculus).

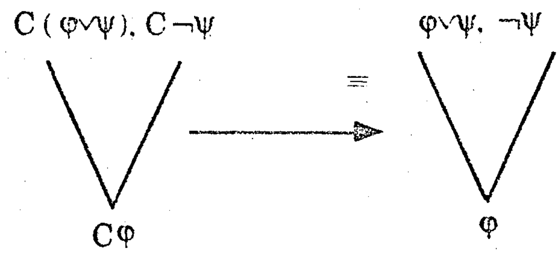
Proof: Suppose that $C\Gamma$, $D\Delta$ is not commitment-consistent.
Therefore $\exists\varphi$ such that $C\Gamma$, $D\Delta$ occ $D\varphi$ and $C\Gamma$, $D\Delta \vdash_{cc} D\neg\varphi$.

For every wff of CL we construct its PC-transform the following way:

1) Deleting C's;
2) Replace D's by $\neg$'s.

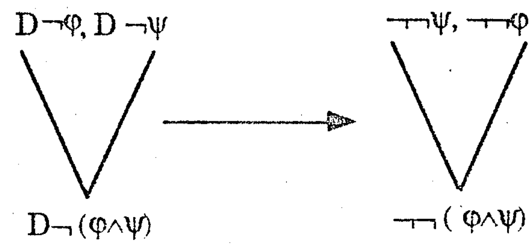
For each rule of CC we see by transforming the premisses and conclusions, that we end up with a valid rule of PC. Below we present two examples of CC rules transformed into PC rules:

1) $\dfrac{C(\varphi \vee \psi)\ .\ C\neg\psi}{C\varphi}$ CvE1



Where $\dfrac{(\varphi \vee \psi), \neg\psi}{\varphi}$ is very similar to one of the '∨' elimination rules for propositional calculus using natural deduction style.

2) $\dfrac{D\neg\varphi\ .\ D\neg\psi}{D\neg(\varphi \wedge \psi)}$ D¬∧I



Where $\dfrac{\neg\neg\varphi, \neg\neg\psi}{\neg\neg(\varphi \wedge \psi)}$ becomes $\dfrac{\varphi, \psi}{\varphi \wedge \psi}$ which is very similar to the '∧' introduction rule for propositional calculus using natural deduction style.

So by induction, any proof in CL can be transformed into a proof in PC.

So by transforming $C\Gamma, D\Delta \vdash_{CC} D\varphi$ we have $\Gamma, \neg\Delta \vdash_{PC} \neg\varphi$ and

by transforming $C\Gamma, D\Delta \vdash_{CC} D\neg\varphi$ we have $\Gamma, \neg\Delta \vdash_{PC} \neg\neg\varphi$.

So $\Gamma, \neg\Delta$ is inconsistent in PC.

4.5. Recapitulation

In this section we developed a Commitment Calculus for dealing with the consequences of commitments inside the commitment stores. Its rules envisage consistency preservation inside the commitment stores. Each party has a commitment store for its positive and negative commitments ('C' commitment store), and a 'D' commitment store as a place for the statements which the party is not committed to. We decided to design the stores in this way, because it is very revealing about the three-valued nature of our notion of commitment. We are not interested in giving a definitive meaning to the notion of commitment, but solely providing a framework within which to operate it.

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. The soundness of this calculus is a trivial result.

Given that we are still experimenting with it, there is no full account of the calculus yet. Thus we did not contemplate giving it a completeness result. Nevertheless, we proved the calculus to be commitment-consistent, showing in this way that it does not fall into a contradiction at the propositional level.

The importance of this calculus to our framework is justified in the next section because of its relevance to the locution modifier Resolution Demand. For its specification the commitment calculus has to be taken into account. A party is only allowed to urge the other party to straigthen its commitment store, if by the sole application of a commitment calculus rule, the latter's commitment store is proved to be contradictory. If the condition above is not met, the dialogue will be brought to an error state.

The commitment calculus however, gives only a static view of commitment. We therefore embed this calculus into a dialogue system which gives a dynamic view of the way that commitments are established and discharged. In the following section a Dialogue Action Component is proposed.

5. Dialogue Action Component

The Dialogue Action Component has axioms of the form {Pre→[Party_AtoParty_B, Locutions] Post}, where Party_A and Party_B are the participants in the dialogue 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 party.

The calculus developed in the previous section, introduces the semantics and the rules for the manipulation of these pre- and post-conditions. Its interaction with DAC is presented in this section.

Two new state of affairs appear in this section, namely *healthy* and *error*. They are introduced in DAC, because in the future we wish to deal with the normative aspects of arguments - valid or invalid, good or bad arguments.

5.1. DAC Axioms

In respect of our framework, DAC is responsible for the insertion/deletion of commitments in the commitment stores. To capture this mechanism we defined it in an axiomatic way using a modal [action] logic style (Goldblatt 82, Maibaum 87).

Its axioms are divided into two different sets: commitment and 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.

These axioms essentially have the format:

Pre → [AtoB, Locution] Post

After Party_A uttered Locution to Party_B, then given Pre happens to be the case prior to Locution, Post happens to be the case after Locution.

Legality axioms define the proper ordering for uttering locutions. Their general shape is:

Pre → [AtoB, Locution] [BtoA, Locution'] Post

After Party_A uttered Locution to Party_B and Party_B uttered Locution' to Party_A then given Pre happens to be the case prior to Locution, Post happens to be the case after Locution'.

At the current stage of this work our dialogues are limited to two parties. However, in the future we would like to extend our framework from two-party to N-party. This justifies the sense of direction given to our locutions: AtoB (from Party_A to Party_B). A possible extension could look like: AtoAudience (from Party_A to Audience) or AtoB_C_D (from Party_A to Parties_B_C_D) which constitutes a qualified audience.

5.1.1. Language for Pre- and Post-conditions

The elements and the rules that form and control the pre- and post-conditions of ACCORD are those defined for the Commitment Calculus. For ACCORD these boundary conditions and the commitment stores are one and the same thing. However, at this level, two new states of affairs are defined - *healthy* and *error* - (remember that commitment stores define s-o-a's) that would be rendered meaningless at the commitment calculus level.

Let us define 'P' - the language of pre- and post-conditions - by extending the language definition in the commitment calculus.

Definition: *Language* L'

Let L' be the propositional language:

$p, q, .. \in L'$ (atomic propositions)

If $\varphi, \psi \in L'$ then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi \in L'$.

Definition: *Language* P

Let P be the language of pre- and post-conditions:

If $\varphi \in L'$ then φ, $C\varphi$, $D\psi$, healthy, error and '$\wedge$' $\in$ P.

Definition: $C\Gamma = \{C\varphi \mid \varphi \in \Gamma\}$

Definition: $D\Delta = \{D\varphi \mid \varphi \in \Delta\}$

Healthy indicates that the dialogue has been played properly until that stage, thus the commitment stores are in good condition. So, empty stores make for healthy conditions. At this point we might be encouraged to think that a party's store describing a contradictory s-o-a, e.g ($C_a\varphi \wedge C_a\neg\varphi$), does not make for a healthy condition, but that is misleading. In the case above, it will be at the other party's discretion to ask Party_A to straighten up its store. Nothing is wrong with the pre- or post-condition.

Error indicates that there is something wrong with the conditions, therefore with the dialogue. In the commitment case, an error state occurs when a party chooses a locution that does not match with the given pre-condition. In the legality case, an improper sequencing of locutions leads to an error. At the current stage of this work, whenever an error state occurs, 'error' is placed in the speaker's commitment store - the one responsible for causing the error state. Otherwise, both commitment stores remain the same.

5.1.2. Locution Modifiers

This section introduces some axioms that characterise each of the following locution modifiers: Assertion, Question, Withdrawal, Challenge, Justification (Justification for the Challenge of an Implication), Denial and Resolution Demand.

Concerning the commitment case, the only changes in the commitment stores are those that appear in the axioms, the rest remains the same. It also presents situations where the uttering of some specific locution under certain pre-conditions leads to an *error* post-condition.

Concerning the legality case, we are only interested in capturing which sequences involving each specific locution modifier lead to a healthy post-condition - we always start from a healthy pre-condition. We first present all the sequences which have that specific locution modifier as the first member, then we present the axioms that end with it

Special attention is given to the locution modifier Resolution Demand. For its specification the commitment calculus has to be taken into account. A party is only allowed to urge the other party to straigthen his (the other party's) commitment store, if by the sole application of a commitment calculus rule, the latter's commitment store is proved to be contradictory. If the condition above is not met, the speaker - the one who demanded the resolution - will bring about an error state. For the sake of clarity, when we present its legality axioms, we assume that the contradiction is in the 'C' store.

As to the choice for the locution modifiers, it was in some sense an arbitrary one. We followed Mackenzie's choice, except for the locution modifier *justification (and justification for the challenge of an implication)* which we introduced. We also treat some of the other locution modifiers in a slightly different way - we grant denial the full status of a locution modifier and we only provide for one version of resolution demand. Mackenzie's choice was influenced by Hamblin's selection, which we believe was inspired by the ancient game of *Obligation*, performed in the *Academy*, a garden near Athens where Plato and Aristotle used to teach.

The reader should always have in mind that our aim is to develop a framework for dialogue representation systems and not the specification of a finished dialogue logic, multi-purpose and ready for use. For each different application, a purpose built dialogue system should be set up and each rule or axiom tailored for its specific task.

There is a lot of work done in the theory of speech acts (Austin62, Searle 69, Searle 79, Searle & Vanderveken 87), that could be related to the nature of the locution modifiers used here but at the current stage of this work the former is not particularly relevant.

Finally, it is important to remember that locution modifiers should be seen in context, i.e. while interacting with each other. This was done when we applied the dialogue formalism to office work In the following sub-sections we first look at them individually, trying to bring out some of the common sense embedded in each of them. For each of them we introduce some commitment and legality axioms. In the next section we combine them, showing in this way that they are expressive enough to emulate some patterns of reasoning which are common in cooperative environments.

5.1.2.1. Language for locution modifiers and ACCORD

Before we start giving the axioms for each locution modifier, we present languages for locution modifiers and ACCORD.

Definition: *Language* L_{Loc}

Let L_{Loc} be the language of locution modifiers:

if $\varphi \in L'$ then asserts(φ) $\in L_{Loc}$;

if $\varphi \in L'$ then questions(φ) $\in L_{Loc}$;

if $\varphi \in L'$ then withdraws(φ) $\in L_{Loc}$;

if $\varphi \in L'$ then why(φ) $\in L_{Loc}$;

if $\varphi \in L'$ then justifies(φ) $\in L_{Loc}$;

if $\varphi \in L'$ then justIMP(φ) $\in L_{Loc}$;

if $\varphi \in L'$ then denies(φ) $\in L_{Loc}$;

if $C\varphi$, $C\psi$, $C\gamma$ and '$\wedge$' $\in P$ then resolve($C\varphi \wedge C\psi / C\gamma$) $\in L_{Loc}$.

Definition: *Language* L_{ACCORD}

Let L_{ACCORD} be the language of ACCORD:

If pre $\in$ P and pos $\in$ P and Loc $\in$ L_{Loc}

then pre $\rightarrow$ [AtoB, **Loc**] post $\in$ L_{ACCORD}

and pre $\rightarrow$ [BtoA, **Loc**] post $\in$ L_{ACCORD}.

5.1.2.1.1. Assertion

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

Language: **asserts(s)**: It is the case that s.

Commitment Axioms

healthy $\rightarrow$ [AtoB, **asserts**(φ)] $C_a\varphi \wedge C_b\varphi$

After Party_A uttered asserts(φ) to Party_B, then given that P is a healthy state prior to the assertion, both parties are committed to φ.

A party is committed to anything stated by another party. The placing of a commitment inside the other party'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) Party_B has to **withdraw** it. When given as an answer to a **question**, an assertion is the same as confirmation or a *yes*.

$C_a\varphi$ $\rightarrow$ [AtoB, **asserts**(φ)] error

After Party_A uttered asserts(φ) to Party_B, then given that Party_A is committed to φ prior to the assertion, P' is an error state, both commitment stores remain the same.

One is not allowed to re-assert one's commitment. Being repetitive is not a *healthy* way to advance reasoning.

Legality Axioms

Starting with an assertion.

healthy $\rightarrow$ [AtoB, **asserts**(φ)] [BtoA, **asserts**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **asserts**(φ)] [BtoA, **questions**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **asserts**(φ)] [BtoA, **withdraws**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **asserts**(φ)] [BtoA, **why**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **asserts**(φ)] [BtoA, **resolve**($C\Gamma/C\psi$)] healthy

Ending with an assertion.

healthy $\rightarrow$ [AtoB, **asserts**(φ)] [BtoA, **asserts**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **questions**(φ)] [BtoA, **asserts**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **withdraws**(φ)] [BtoA, **asserts**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **justifies**(φ)] [BtoA, **asserts**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **denies**(φ)] [BtoA, **asserts**(ψ)] healthy

healthy $\rightarrow$ [AtoB, **resolve**($C\Gamma/C\varphi$)] [BtoA, **asserts**(ψ)] healthy

5.1.2.1.2. Question

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

Language: **questions(s)**: Is it the case that s?

Commitment Axioms

$$\text{healthy} \rightarrow [\text{AtoB, questions}(\varphi)]\ \text{healthy=}$$

After Party_A uttered questions(φ) to Party_B, then given that P is a healthy state prior to the question, P' is a healthy state. Both commitment stores remain the same.

The uttering of a question does not affect the commitment stores. After asking a question, it is up to the other party to confirm it (**assert**), deny it (**denial**) or distance himself from it (**withdraw**). 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.

$$C_a\varphi \rightarrow [\text{AtoB, questions}(\varphi)]\ \text{error}$$

After Party_A uttered questions(φ) to Party_B, then given that Party_A is committed to φ prior to the question, P' is an error state.

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

Legality Axioms

Starting with a question.

healthy → [AtoB, **questions**(φ)] [BtoA, **asserts**(ψ)] healthy

healthy → [AtoB, **questions**(φ)] [BtoA, **denies**(ψ)] healthy

healthy → [AtoB, **questions**(φ)] [BtoA, **withdraws**(ψ)] healthy

Ending with a question.

healthy → [AtoB, **asserts**(φ)] [BtoA, **questions**(ψ)] healthy

healthy → [AtoB, **withdraws**(φ)] [BtoA, **questions**(ψ)] healthy

healthy → [AtoB, **justifies**(φ)] [BtoA, **questions**(ψ)] healthy

healthy → [AtoB, **denies**(φ)] [BtoA, **questions**(ψ)] healthy

For future extensions on the subject of *questions*, there is a vast literature that we will take into consideration - notably *New Foundations for a Theory of Questions and Answers* (Hintikka 83), *The Logic of Questions* (Harrah 84) and *On Questions* (Hoepelman 83).

5.1.2.1.3. Withdrawal

To withdraw a statement is to dissociate oneself from the s-o-a described by 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 he (she) is not committed to it any more.

Language: **withdraws(s)**: I am not so sure that s or No commitment to s.

Commitment Axioms

$$C_a\varphi \rightarrow [\text{AtoB, withdraws}(\varphi)]\ D_a\varphi$$

After Party_A uttered withdraws(φ) to Party_B, then given that Party_A is committed to φ prior to the withdrawal, A is not committed to φ.

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

$$\text{healthy} \rightarrow [\text{AtoB, withdraws}(\varphi)]\, D_a\varphi$$

After Party_A uttered withdraws(φ) to Party_B, then given that P is a healthy state prior to the withdrawal, Party_A is not committed to $\overline{\varphi}$.

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

$$D_a\varphi \rightarrow [\text{AtoB, withdraws}(\varphi)]\ \text{error}$$

After Party_A uttered withdraws(φ) to Party_B, then given that Party_A is not committed to φ prior to the withdrawal, P' 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

Starting with a withdrawal.

healthy → [AtoB, withdraws(φ)] [BtoA, asserts(ψ)] healthy

healthy → [AtoB, withdraws(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, withdraws(φ)] [BtoA, why(ψ)] healthy

healthy → [AtoB, withdraws(φ)] [BtoA, resolve(CΓ/Cψ)] healthy

healthy → [AtoB, withdraws(φ)] [BtoA, questions(ψ)] healthy

Ending with a withdrawal.

healthy → [AtoB, asserts(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, questions(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, withdraws(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, why(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, justifies(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, denies(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, resolve(CΓ/Cφ)] [BtoA, withdraws(ψ)] healthy

5.1.2.1.4. Challenge

Paired together with justification, challenge forms the basis of the argumentation process. Challenging a statement differs from questioning it, for while in the latter case one is asking for some form of confirmation, in the former case one is asking for *good reasons* to stick to it.

Language: why(s): Why is it to be supposed that s? or How is it known that s?

Commitment Axioms

$$\text{healthy} \rightarrow [\text{AtoB, why}(\varphi)]\, C_a(\text{why}(\varphi)) \wedge C_b\varphi$$

After Party_A uttered why(φ) to Party_B, then given that P is a healthy state prior to the challenge; Party_A is committed to why(φ) and Party_B is committed to φ.

A challenged Statement will be marked - as a why(Statement) - in the commitment store of the speaker. This is part of the mechanism to avoid circularity in answering other challenges (why(a); justifies(b); why(b); justifies(a)). The challenged Statement is placed unmarked into the hearer's commitment store, to force him (her) to react to it. A challenge is a demand for evidence. It is neither a demand for a confirmation nor for a denial like a question.

$$C_a\varphi \wedge C_b\varphi \rightarrow [AtoB, why(\varphi)]\ C_a(why(\varphi)) \wedge C_b\varphi$$

After Party_A uttered why(φ) to Party_B, then given that both parties are committed to φ prior to the challenge, A is committed to why(φ) - but not to φ any more - and Party_B remains committed to φ.

A challenge forces the hearer to disclose the premises that led him to make that statement.

Legality Axioms

Starting with a challenge.

healthy → [AtoB, **why(φ)**] [BtoA, **justifies (ψ)**] healthy

healthy → [AtoB, **why(φ)**] [BtoA, **withdraws(ψ)**] healthy

healthy → [AtoB, **why(φ)**] [BtoA, **resolve(CΓ/Cψ)**] healthy

Ending with a challenge.

healthy → [AtoB, **asserts(φ)**] [BtoA, **why(ψ)**] healthy

healthy → [AtoB, **withdraws(φ)**] [BtoA, **why(ψ)**] healthy

healthy → [AtoB, **justifies(φ)**] [BtoA, **why(ψ)**] healthy

healthy → [AtoB, **denies(φ)**] [BtoA, **why(ψ)**] healthy

5.1.2.1.5. Justification

Given as an answer for a challenge, forms the basis of the argumentation process. It provides *good reasons* for the challenger to stick to the challenged statement.

Language: **justifies(s)**: s is a justification for Challenge.

Commitment Axiom

$$C_a(why(\varphi)) \wedge C_b\varphi \rightarrow [BtoA, justifies(\psi)]$$
$$C_a(why(\varphi)) \wedge C_a(\psi\rightarrow\varphi) \wedge C_a\psi \wedge C_b\varphi \wedge C_b(\psi\rightarrow\varphi) \wedge C_b\psi$$

After Party_B uttered justifies(ψ) to Party_A, then given that Party_A is committed to why(φ) and Party_B is committed to φ prior to the justification, Party_A is committed to why(φ), (ψ→φ) and ψ, and Party_B is committed to φ, (ψ→φ) and ψ after it.

A justification places the answer and the argumentation step (ψ, (ψ→φ)) into both commitment stores. It provides the premises that support the challenged statement.

JustIMP is a variant form of the locution modifier Justification, specially designed for dealing with challenges of formulas containing one implication.

Commitment Axiom for the Justification for the Challenge of an Implication

$$C_a(why(\gamma\rightarrow\varphi)) \wedge C_b(\gamma\rightarrow\varphi) \rightarrow [BtoA, justIMP(\psi)]$$
$$C_a(why(\gamma\rightarrow\varphi)) \wedge C_a(\psi\rightarrow\varphi) \wedge C_a\psi \wedge C_b(\gamma\rightarrow\varphi) \wedge C_b(\psi\rightarrow\varphi) \wedge C_b\psi$$

After Party_B uttered justIMP(ψ) to Party_A, then given that Party_A is committed to why(γ→φ) and Party_B is committed to (γ→φ) prior to the justification, Party_A is committed to why(γ→φ), (ψ→φ) and φ, and Party_B is committed to (γ→φ), (ψ→φ) and ψ after it.

JustIMP was designed in an *ad hoc* way. In the future we might resort to some specific reasoning strategies for justification to achieve the same result. The Legality Axioms are the same for Justification and for JustIMP.

Legality Axioms

Starting with a justification

healthy → [AtoB, justifies(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, justifies(φ)] [BtoA, resolve(CΓ/Cψ)] healthy

healthy → [AtoB, justifies(φ)] [BtoA, questions(ψ)] healthy

healthy → [AtoB, justifies(φ)] [BtoA, why(ψ)] healthy

healthy → [AtoB, justifies(φ)] [BtoA, asserts(ψ)] healthy

Ending with a justification.

healthy → [AtoB, why(φ)] [BtoA, justifies (ψ)] healthy

5.1.2.1.6. Denial

This locution modifier can only be used after a question. It commits both the speaker and the hearer to the negation of the questioned statement.

Language: **denies(s)**: I deny that it is the case that s.

Commitment Axiom

$$\text{healthy} \rightarrow [\text{AtoB, denies}(\varphi)]\ C_a\neg\varphi \wedge C_b\neg\varphi$$

After Party_A uttered denies(φ) to Party_B, then given that P is a healthy state prior to the denial, both parties are committed to ¬φ.

In the commitment axiom above, we are treating the denial of a statement as an assertion of the negation of that statement. Thus, a denial is stronger than a withdrawal.

Legality Axioms

Starting with a Denial.

healthy → [AtoB, denies(φ)] [BtoA, asserts(ψ)] healthy

healthy → [AtoB, denies(φ)] [BtoA, withdraws(ψ)] healthy

healthy → [AtoB, denies(φ)] [BtoA, resolve(CΓ/Cψ)] healthy

healthy → [AtoB, denies(φ)] [BtoA, questions(ψ)] healthy

healthy → [AtoB, denies(φ)] [BtoA, why(ψ)] healthy

Ending with a denial.

healthy → [AtoB, questions(φ)] [BtoA, denies(ψ)] healthy

5.1.2.1.7. Resolution Demand

It is used by the speaker to alert the hearer that his (the hearer's) commitment store is in a inconsistent state. The hearer has to take some action to correct it. CC and DAC are connected via this locution modifier. A party can only demand a resolution from the other party, if by the sole application of a commitment calculus rule, the latter's commitment store is proved to be inconsistent. Otherwise the dialogue is not a *healthy* one any more. The proper uttering of a resolution demand does not affect the commitment stores.

Language: **resolve(Set_of_Statements/Statement)**:

Resolve this set of statements against this specific statement.

The commitment axioms for resolution demand have the general form:

$$C_b(\text{contradictory}) \rightarrow [\text{AtoB, resolve}(C\Gamma/C\varphi)]\ C_b(\text{contradictory})$$

After Party_A uttered resolve($C\Gamma/C\varphi$) - the set of commitments designated by Γ and the specific commitment φ being proved to be contradictory by the sole application of a commitment calculus rule - to Party_B, then given that Party_B has this form of contradiction in his commitment store prior to the resolution demand, both commitment stores remain the same after it.

Below we instantiate this axiom using three potentially contradictory commitment stores:

$$C_b(\varphi\wedge\psi) \wedge C_b\neg\varphi \rightarrow [\text{AtoB, resolve}(C(\varphi\wedge\psi)/C\neg\varphi)]\ C_b(\varphi\wedge\psi) \wedge C_b\neg\varphi$$

After Party_A uttered resolve($C(\varphi\wedge\psi)/C\neg\varphi$) to Party_B, then given that Party_B is committed to $(\varphi\wedge\psi)$ and $\neg\varphi$ prior to the resolution demand, both commitment stores remain the same after it.

From rule C∧E1 we see that Party_B's commitment store is in an inconsistent state.

$$C(\varphi\rightarrow\psi) \wedge C\varphi \wedge C\neg\psi \rightarrow [\text{AtoB, resolve}(C(\varphi\rightarrow\psi),C\varphi/C\neg\psi)]\ C(\varphi\rightarrow\psi) \wedge C\varphi \wedge C\neg\psi$$

After Party_A uttered resolve($C(\varphi\rightarrow\psi),C\varphi/C\neg\psi$) to Party_B, then given that Party_B is committed to $(\varphi\rightarrow\psi)$, φ and $\neg\psi$ prior to the resolution demand, both commitment stores remain the same.

From rule C→E1 we see that Party_B's commitment store is in an inconsistent state.

$$D\varphi \wedge D(\varphi\wedge\psi) \rightarrow [\text{AtoB, resolve}(D\varphi/D(\varphi\wedge\psi))]\ \text{error}$$

After Party_A uttered resolve($D\varphi/D(\varphi\wedge\psi)$) to Party_B, then given that Party_A is not committed to φ and $(\varphi\wedge\psi)$ prior to the resolution demand, P' is an error state, i.e. the improper uttering of a resolution demand places 'error' in the speaker's C-commitment store. Party_B remains not committed to φ and $(\varphi\wedge\psi)$.

From rule D∧I1 we cannot detect any inconsistency in Party_B's commitment store.

At the current stage of this work, the party that is asking for the resolution demand, does not have to show which commitment calculus rule he used to spot the inconsistency. In the future we might make this necessary, for in a cooperative environment showing the means by which one detected a mistake is always helpful.

Finally concerning ACCORD, being committed to something and not being committed to the same thing implies having a contradictory commitment store. Thus Party_B can demand a resolution from Party_A if $\{C_a\varphi \wedge C_a\neg\varphi\}$. However, if instead Party_A's commitment store's contents were $\{D_a\varphi \wedge D_a\neg\varphi\}$, then a resolution demand coming from Party_B would bring the dialogue to an error state. For example, if *I'm committed to go to the cinema* and *I'm committed not to go to the cinema*, then there is something wrong with me. On the other hand, if *I'm not committed to go to the cinema* and *I'm not committed not to go to the cinema*, apart from a strange fixation on movies, there is nothing wrong with me.

Legality Axioms

Starting with a resolution demand.

healthy → [AtoB, resolve($C\Gamma/C\varphi$)] [BtoA, asserts(ψ)] healthy

healthy → [AtoB, resolve($C\Gamma/C\varphi$)] [BtoA, withdraws(ψ)] healthy

Ending with a resolution demand.

healthy → [AtoB, asserts(φ)] [BtoA, resolve($C\Gamma/C\psi$)] healthy

healthy → [AtoB, withdraws(φ)] [BtoA, resolve($C\Gamma/C\psi$)] healthy

healthy → [AtoB, why(φ)] [BtoA, resolve($C\Gamma/C\psi$)] healthy

healthy → [AtoB, justifies(φ)] [BtoA, resolve($C\Gamma/C\psi$)] healthy

healthy → [AtoB, denies(φ)] [BtoA, resolve($C\Gamma/C\psi$)] healthy

5.2. Derivability

The only rule necessary for defining the Dialogue Representation derivability relation $\vdash_{ACCORD}$ - is the transitivity rule shown below.

Transitivity rule:

$$\frac{P \rightarrow [AtoB, Loc]\ P' \quad P' \rightarrow [BtoA, Loc']\ P''}{P \rightarrow [AtoB, Loc]\ [BtoA, Loc']\ P''}$$

Given a set Γ of propositions of L_{ACCORD} and $\varphi \in L_{ACCORD}$ then,
$\Gamma \vdash_{ACCORD} \varphi$ iff

$\exists\ \varphi_1, \varphi_2, \ldots, \varphi_n$ where
$\varphi = \varphi_n$ and
for each i, i = 1 to n,
$\varphi_i \in \Gamma$ or
$\varphi_i \in$ logical axioms of ACCORD or
φ_i is the conclusion of the transitivity rule using premisses
φ_j and φ_k for φ, $k < i$.

Definition of Healthy Dialogue:

A dialogue is legal iff

$\vdash_{ACCORD} P \rightarrow [Party_i toParty_j, locution]^n\ P$ or
$\Gamma \vdash_{ACCORD} P \rightarrow [Party_i toParty_j, locution]^n\ P$
where $n \geq 1$ and $j \neq i$.

6. Cliches, Scripts and Patterns of Cooperative Reasoning

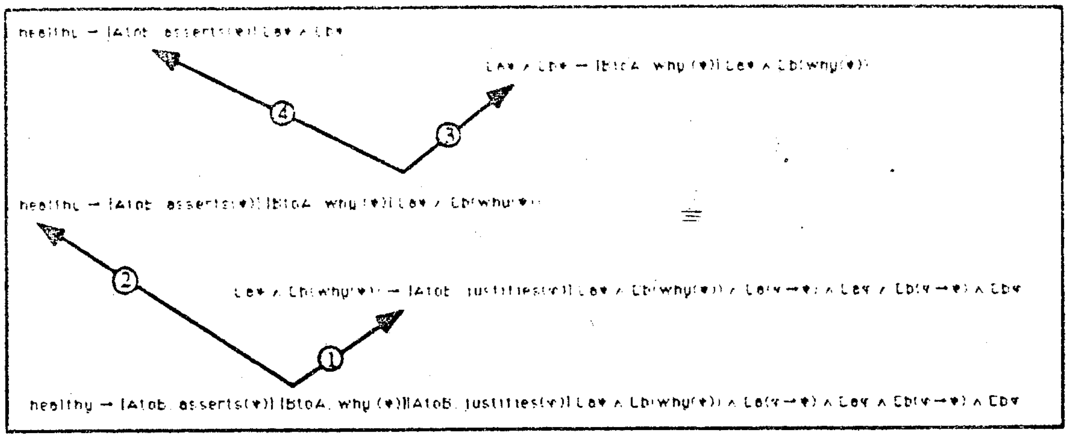
In this section we introduce the notions of cliches, scripts and patterns of cooperative reasoning, for extending our framework with a view to support cooperative work.

Next we present two patterns of cooperative reasoning: Justifying and Resolving. By cooperative reasoning we loosely mean the kind of reasoning that is common in the office and in the classroom for example. Although not particularly relevant at this stage of the work, an interesting *classification of cooperative illocutionary acts* provided by (Hancher 79) is going to be taken into consideration for future extensions.

Below we verify if they are ACCORD-derivable and also if they are *healthy* dialogues. First, we check in all steps if the post-condition is well formed. Then, we check if the sequencing of the dialogue events is sound. For each dialogue we show its derivation tree.

1) The dialogue presented below typifies a **justifying** conversation, where Party_A is challenged by Party_B to justify his (her) previous assertion.

healthy $\rightarrow$ [AtoB, asserts(ψ)]
[BtoA, why(ψ)]
[AtoB, justifies(φ)]
$C_a\psi \wedge C_b(why(\psi)) \wedge C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \wedge C_b(\varphi \rightarrow \psi) \wedge C_b\varphi$



The post-condition $\{C_a\psi \wedge C_b(\mathbf{why}(\psi)) \wedge C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \wedge C_b(\varphi \rightarrow \psi) \wedge C_b\varphi\}$ is well formed.

The sequencing of dialogue events $\{[\text{BtoA}, \mathbf{why}(\psi)]\ [\text{AtoB}, \mathbf{justifies}(\varphi)]\}$ is sound according to one of the legality axioms for Challenge.

We take the post-condition $\{C_a\psi \wedge C_b(\mathbf{why}(\psi)) \wedge C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \wedge C_b(\varphi \rightarrow \psi) \wedge C_b\varphi\}$ together with the second dialogue event from the chosen pair $\{[\text{AtoB}, \mathbf{justifies}(\varphi)]\}$ and transform it to the pre-condition that makes the commitment axiom for that specific locution {justifies} correct. Thus we get $\{C_a\psi \wedge C_b(\mathbf{why}(\psi))\}$ as the pre-condition for (1).

The pre-condition for (1) is the post-condition for (2), and it is correct by construction (or, more appropriately, by destruction).

The sequencing of dialogue events $\{[\text{AtoB}, \mathbf{asserts}(\psi)]\ [\text{BtoA}, \mathbf{why}(\psi)]\}$ is sound according to one of the legality axioms for Assertion.

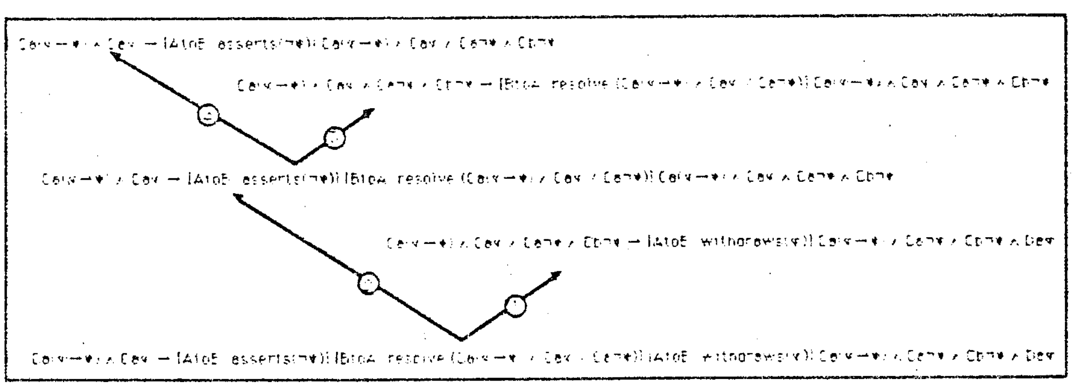
We take the post-condition $\{C_a\psi \wedge C_b(\mathbf{why}(\psi))\}$ together with the second dialogue event from the chosen pair $\{[\text{BtoA}, \mathbf{why}(\psi)]\}$ and transform it to the pre-condition that makes the commitment axiom for Challenge correct. Thus we get $\{C_a\psi \wedge C_b\psi\}$ as the pre-condition for (3).

The pre-condition for (3) is the post-condition for (4).

Given that (4) is the first dialogue event, we only have to take its post-condition $\{C_a\psi \wedge C_b\psi\}$ together with the dialogue event $\{[\text{AtoB}, \mathbf{asserts}(\psi)]\}$ and transform it to the pre-condition that makes the commitment axiom for Assertion healthy. Thus we get {healthy}, which matches with our *dialogue* pre-condition. *QED*.

2) The dialogue presented below typifies a **resolving** conversation, where Party_A is urged by Party_B to resolve a contradiction in his (her) commitment store, that was created by his (her) previous assertion.

$$C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \rightarrow [\text{AtoB}, \mathbf{asserts}(\neg\psi)]$$
$$[\text{BtoA}, \mathbf{resolve}(C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \,/\, C_a\neg\psi)]$$
$$[\text{AtoB}, \mathbf{withdraws}(\varphi)]\ C_a(\varphi \rightarrow \psi) \wedge C_a\neg\psi \wedge C_b\neg\psi \wedge D_a\varphi$$



The post-condition $\{C_a(\varphi \rightarrow \psi) \wedge C_a\neg\psi \wedge C_b\neg\psi \wedge D_a\varphi\}$ is well formed.

The sequencing of dialogue events {[BtoA, resolve($C_a(\varphi \rightarrow \psi) \wedge C_a\varphi$ / $C_a\neg\psi$)] [AtoB, withdraws(φ)]} is sound according to one of the legality axioms for Resolution Demand.

We take the post-condition $\{C_a(\varphi \rightarrow \psi) \wedge C_a\neg\psi \wedge C_b\neg\psi \wedge D_a\varphi\}$ together with the second dialogue event from the chosen pair {[AtoB, withdraws(φ)]} and transform it to the pre-condition that makes the commitment axiom for that specific locution {withdraws} correct. Thus we get $\{C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \wedge C_a\neg\psi \wedge C_b\neg\psi\}$ as the pre-condition for (1).

The pre-condition for (1) is the post-condition for (2), and it is correct by construction.

The sequencing of dialogue events {[AtoB, asserts($\neg\psi$)][BtoA, resolve($C_a(\varphi \rightarrow \psi) \wedge C_a\varphi$ / $C_a\neg\psi$)]} is sound according to one of the legality axioms for Assertion.

We take the post-condition $\{C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \wedge C_a\neg\psi \wedge C_b\neg\psi\}$ together with the second dialogue event from the chosen pair {[BtoA, resolve($C_a(\varphi \rightarrow \psi) \wedge C_a\varphi$ / $C_a\neg\psi$)]} and check if there is a commitment calculus rule that given $\{C_a(\varphi \rightarrow \psi) \wedge C_a\varphi\}$ as a premise, it concludes an implicit commitment that contradicts $\{C_a\neg\psi\}$. We find rule C→E1.Then we transform it to the pre-condition that makes the commitment axiom for Resolution Demand correct. Thus we get $\{C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \wedge C_a\neg\psi \wedge C_b\neg\psi\}$ as the pre-condition for (3).

The pre-condition for (3) is the post-condition for (4).

Given that (4) is the first dialogue event, we only have to take its post-condition $\{C_a(\varphi \rightarrow \psi) \wedge C_a\varphi \wedge C_a\neg\psi \wedge C_b\neg\psi\}$ together with the dialogue event {[AtoB, asserts($\neg\psi$)]} and transform it to the pre-condition that makes the commitment axiom for Assertion correct. Thus we get $\{C_a(\varphi \rightarrow \psi) \wedge C_a\varphi\}$, which matches with our *dialogue* pre-condition. *QED*.

We checked elsewhere (Fuks 91) that the examples B and C in section 2 are ACCORD-derivable. Some changes to the commitment stores were made necessary, because after the development of the commitment calculus in section 4, we introduced the 'C' and 'D' notation for commitments. First we translated the commitment stores into the new notation. Then, starting from the resulting commitment stores, by only applying ACCORD rules, we returned to empty - thus *healthy* - commitment stores. These healthy dialogues were presented as proof trees using a backwards deduction style. For each example we tried to identify bits of this decomposed chain of reasoning, that are similar to the patterns of cooperative reasoning presented above.

The identification of these sequences - patterns - of locution modifiers is the first step towards the construction of higher level structures like scripts and cliches. These structures should be able to enact these patterns of cooperative reasoning in an *intelligent* way. We are still at a very early stage of the development of this framework, and at the moment we are only able to provide the representation schemes that can capture these patterns of reasoning. Other possible patterns of cooperative reasoning are: **clarifying**; **searching**; **question-answering**; **observing**; **negotiating**; **bargaining**, etc.

Currently we are extending the framework for the development of groupware for cooperative software design (Lucena, Leite, Schwabe & Fuks 91). We follow a path similar to the one taken by (Rittel & Kunz 70, Lowe 85 and Sculer & Smith 90) where design decisions and rationales are logged using argumentation structures. For this purpose, we are working on cliches and scripts for the provision of conversation stereotypes that could *guide* the software designers in their tasks. The following is a sample script:

A-Script

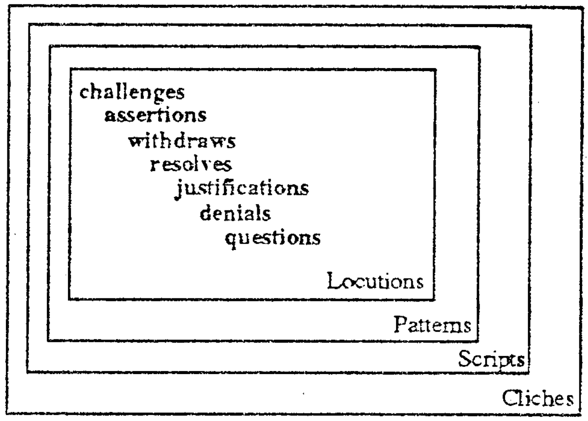
Current S-O-A: Inconsistency in Party_A's commitment store

If Party_B wishes

Then Party_B to Party_A {resolving}

Cliches are state transition diagrams where state transformations occur by the uttering of dialogue events. An interesting example of cliche is the *Conversation for Action* diagram in (Winograd 88) That particular cliche could have its offer and counter-offer steps enacted by using argumentative structures of the kind presented by (van Eemeren & Grootendorst 84). The conversation built around a valid argument would progressively construct a dialogue structure, which should map to a well formed Toulmin structure (Toulmin 64) or other similar structure.

A further improvement will be the incorporation of cognitive heuristics to the framework in order to provide it with the ability of learning cliches from existing dialogues. This will also make the framework more flexible, for *a priori* cliches might not be appropriate for sundry domain specific applications. This proficiency will be especially useful for user interface - and *group*face - design, where the users' profile and the group's social structure emanate from their conversations.



The figure above illustrates the interrelation between locutions, patterns, scripts and cliches. Cliches are made out of other cliches, scripts, patterns and locutions. At the current stage of this work, we are working 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.

7. Conclusion

Negotiation permeates many aspects of our everyday life. Normally, the accomplishment of a task involving more than one party is by means of agreements, commitments and compromises. Human organizations, being based on team activity, strongly rely upon negotiation. Computational models based on dialectical reasoning can be successfully applied to problems requiring negotiation.

Cooperation in the work place is increasingly becoming a major issue in business, industry and in academe. As a result of this, different types of users are sharing and working on common information. Tools and methodologies are needed to instigate, support and structure this interdisciplinary debate. Cooperative work can be seen as the generation of language acts and conversations (Winograd 88). Conversation is a mechanism for generating commitments.

In this work we developed ACCORD - a framework for dialogue representation systems. It evolved from a formalism for the analysis of dialogue created by Hamblin and extended by Mackenzie. It comprises a commitment calculus and a dialogue action component.

We applied this framework to a *cooked* case originating from an office environment, focussing on its cooperative and group aspects. We tried to show that this framework is suitable for capturing and representing some of the cooperative aspects characteristic of group activities.

ACCORD as it stands now is still in an embryonic stage and a lot remains to be done.

CC is still incomplete. It needs a lot of testing and there is plenty of room for improvement. We want to reassess the work done on Commitment Logic (Fuks, Rysan & Sadler 89) and try to link it to the calculus. This development together with the use of deontic operators within DAC's axioms, will probably make this framework more capable of handling problems that involve negotiation. A further improvement will be the extension of ACCORD from a *two-party* system into a *N-party* one.

We also intend to extend this dialogue representation system, by using the notions of intention (Cohen & Levesque 87, Bratman 87) belief and knowledge in a way similar of that we have been using commitment. Intention stores, knowledge stores and belief stores will be created and intentions, knowledge and beliefs will be established and discharged by locution modifiers in a way similar to that envisaged for commitments. A new set of axioms or rules shall be devised for interconnecting theses stores, i.e. for prescribing the effects that updatings in any of the stores will cause in the other stores.

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. The future use of this framework for the development of groupware could give us more insight into the process of software design, especially from the team activity point of view (Lucena, Leite, Schwabe & Fuks 91), and could also give valuable feedback for improving the framework itself. At the moment we are trying to develop cliches and scripts for this purpose.

An immediate advantage of using ACCORD is that group awareness is a natural consequence, and that group work support systems based on it will naturally tend to meet the end-users' language capabilities. We believe that this framework could be a source of inspiration for the design - and in the future for the development - of *groupface* management systems.

There are many other applications to this framework in computing. In the subject of distributed reasoning, it could be applied to the detection of mutual consistency in distributed databases. It could help in the design of collaborative editing systems. Finally it could be used in the fast developing field of multimedia, for the provision of argumentative structures for computerised conferencing systems.

Acknowledgements

I would like to thank Tom Maibaum, Martin Sadler, Dov Gabbay, Anthony Finkelstein and Daniel Schwabe for their constructive comments, interest and help.

Financial support has come from CNPq - *Conselho Nacional de Desenvolvimento Científico e Tecnológico* - Brasilian national council for scientific research and technological development, grants 300608/90.0 and 20.2471/86-CC.

Most of this work was developed whilst the author was doing his Ph. D. at the Department of Computing, Imperial College of Science Technology and Medicine, London University.

References

Allwood, J.: Logic and Spoken Interaction in Reasoning and Discourse Processes, ed. Myers, Brown & McGonigle, pp 67-94, Academic Press Cognitive Science Series, 1986.

Anderson, A. R.: On the logic of commitment, Philosophical Studies V10, pp 23-27, 1959.

Austin, J. L.: How to Do Things with Words, ed: J. O. Urmson and M. Sbisa: 2nd edition 1980, Oxford University Press, 1962.

Barth, E. M. & Krabbe, E. C. W.: From Axiom to Dialogue: A Philosophical Study of Logics and Argumentation; de Gruyter, 1982.

Bond, A. H.: A Computational Model for Organizations of Cooperating Intelligent Agents: Proceedings Conference on Office Information Systems, SIGOIS Bulletin, V11, N2&3, pp 21-30, 1990.

Bratman, E. W.: What Is Intention: Report nº CLSI-87-90, January 1987.

Castaneda, H. N.: The Logic of Obligation. Philosophical Studies V10, pp 17-23, 1959.

Carbonell, J.G.: Meta-Language Utterances in Purposive Discourse; Carnegie-Mellon Univ Tech. Rep, CMU-CS-82-125, 1982.

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.

Davis, R. & Smith, R. G.: Negotiation as a Metaphor for Distributed Problem Solving; Artificial Intelligence V20, pp 63-109, 1983.

Erman, L. & Lesser, V.: A Multi-Level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge, Proc. IJCAI-75, pp 483-489, 1975.

Finkelstein, A. & Fuks H.: Multi-party Specification: Proc 5th International Workshop on Software Specification & Design, pp 185-195; IEEE CS Press, 1989.

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.

Gabbay, D. M.: N-Prolog; An Extension of Prolog with Hypothetical Implications 2, Journal of Logic Programming, V2, pp 251-283, 1985.

Gardenfors, P.: Knowledge in Flux: Modelling the Dynamics of Epistemic States, The MIT Press, 1988.

Goldblatt, R.: Axiomatising the Logic of Computer Programming. XI; Lectures Notes in Computer Science V130, Springer-Verlag,1982.

Green, M.: *Report on Dialogue Specification tools* In Proc. Workshop. User Interface Management Systems; ed: Pfaff; Springer-Verlag, 1983.

Hamblin, C. L.: Fallacies; Methuen, 1970.

Hamblin, C. L.: Mathematical models of dialogue, Theoria V37, pp 130-155, 1971.

Hamblin, C. L.: Imperatives, Basil Blackwell Ltd, 1987.

Hancher, M. : The classification of cooperative illocutionary acts, Language in Society V8, pp 1-14, Cambridge University Press, 1979.

Harrah, D.: *The Logic of Questions* in Handbook of Philosophical logic, V2, pp 715-764; ed: D. Gabbay and F. Guenthner; D. Reidel, 1984.

Hintikka, J.: *Some main problems of Deontic Logic* in Deontic Logic: Introductory and Systematic Readings, pp 225-248; ed: R. Hilpinen; D. Reidel, 1971.

Hintikka, J.: *New Foundations for a Theory of Questions and Answers* in Questions and Answers, pp 159-190; ed: Ferenc Kiefer; D. Reidel, 1983.

Hoepelman, J.: *On Questions* in Questions and Answers, pp 191-227; ed: Ferenc Kiefer; D. Reidel, 1983.

Hughes, G. E. & Cresswell, M. J.: An introduction to Modal Logic, Methuen, 1968.

Kornfeld, W. & Hewitt, C.: The Scientific Community Metaphor; IEEE Trans Systems, Man & Cybernetics, SMC-11,1, pp 24-32, 1981.

Koo, C. & Wiederhold, G.: A commitment-based communication model for distributed office environenments; Proceedings Conference on Office Information Systems, SIGOIS Bulletin, V9, N2&3, pp 291-298, 1988.

Ladriere, J.: *Logic and Argumentation*; in From Metaphysics to Rhetoric, ed: Michel Meyer, pp 15-35, Synthese Library, 1989.

Lenat, D.: Beings: Knowledge as Interacting Experts; Proc. IJCAI-75, pp126-133, 1975.

Lucena, C. J. P., Leite, J., Schwabe, D. & Fuks, H.: A Research Agenda on Software Design; Research Report nº 29/91, Departamento de Informática, Catholic University of Rio de Janeiro, March 1991.

Lorenz, K.: *On the Criteria for the choice of Rules of Dialogic* in Argumentation: Approaches to Theory Formation; SLCS V8, ed: E.M Barth. & J. L. Martens; John Benjamins B.V., 1982.

Lowe, D. G.: Co-operative stucturing of information: the representation of reasoning and debate; International Journal of Man-Machine Studies V23, pp 97-111, 1985.

Mackenzie, J.: How to stop talking to tortoises; Notre Dame Journal of Formal Logic, V20, pp 705-717, 1979.

Mackenzie, J.: Why do we number theorems?; Australasian Journal of Philosophy, V58, pp 135-149, 1980.

Mackenzie, J.: The Dialectics of Logic; Logique et Analyse, V24, pp 159-177, 1981.

Mackenzie, J.: Begging the question in dialogue; Australasian Journal of Philosophy, V62, N2, pp 174-181, 1984.

Mackenzie, J.: No Logic before Friday; Synthese, V63, pp 329-341, 1985.

Maibaum, T. S E.: A logic for the Formal Requirements Specification of Real-Time/Embedded Systems. FOREST internal report, Imperial College, London, 1987.

Pollock, J. P.: The Foundations of Philosophical Semantics, Princeton University Press, 1984.

Rescher, N.: An Axiom System for Deontic Logic, Philosophical Studies V9, pp 24-30, 1958.

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.

Robinson, W. N.: Negotiation Behavior During Multiple Agent Specification: A Need for Automated Conflict Resolution; CIS-TR-89-13; Dept Computer and Information Science, Univ of Oregon, 1989.

Sculer, W. & Smith, B. J.: *Author's Argumentation Assistant (AAA)*: A Hypertext Based Authoring Tool for Argumentative Texts; in HYPERTEXT: Concepts Systems and Applications, (eds.) A. Rizk, N.Streitz and J. André, The Cambridge Series in Eletronic Publishing, 1990.

Schneiderman, B.: Multiparty Grammars and Related Features for Defining Interactive Systems; IEEE Trans Systems, Man & Cybernetics, SMC-12,1, pp 148-154, 1982.

Searle, J. R.: Speech Acts: An Essay in the Philosophy of Language; Cambridge University Press, 1969.

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

Searle, J. R. & Vanderveken, D.: Foudations of Illocutionary Logic; Cambridge University Press, 1987.

Smith, R.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver; IEEE Trans. Computers, C-29,12, pp 1104-1113, 1980.

Smith, R. & Davis, R.: Frameworks for Cooperation in Distributed Problem Solving; IEEE Trans Systems, Man & Cybernetics, SMC-11,1, pp 61-69, 1981.

Schneiderman, B.: Multiparty Grammars and Related Features for Defining Interactive Systems; IEEE Trans Systems, Man & Cybernetics, SMC-12,1, pp 148-154, 1982.

Toulmin, E. S.: The Uses of Argument, Cambridge University Press, 1964.

van Eemeren, F. H., Grootendorst, R.: Speech Acts in Argumentative Discussions: A Theoretical Model for the Analysis of Discussions Directed towards Solving Conflicts of Opinion; Foris Publications, 1984.

von Wright, G. H.: Deontic Logic, Mind V60, pp 1-15, 1951.

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.