

Formalisation et Réalisation d'un Service de Négociation en Ingénierie Concourante

Karim Baïna

▶ **To cite this version:**

Karim Baïna. Formalisation et Réalisation d'un Service de Négociation en Ingénierie Concourante. [Stage] 99-R-433 || baina99a, 1999, 53 p. inria-00107828

HAL Id: inria-00107828
https://inria.hal.science/inria-00107828

Submitted on 19 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Joseph Fourier

U.F.R Informatique
Mathématiques Appliquées

Institut National Polytechnique
de Grenoble
ENSIMAG

I.M.A.G.

DEA D'INFORMATIQUE :

SYSTÈMES ET COMMUNICATIONS

Projet présenté par :

Karim BAÏNA

Formalisation et Réalisation d'un Service de Négociation en Ingénierie Concourante

Effectué au LORIA (Nancy)

(Laboratoire lorrain de recherche en informatique et ses applications)

Date : 16 juin 1999

Jury : Mme Brigitte PLATEAU
M. Didier BERT
M. Roland BALTER
M. Dominique DECOUCHANT
M. Khalid BENALI

Remerciements

Je tiens à remercier Monsieur Claude Godart pour m'avoir accueilli au sein de l'équipe ECOO du LORIA,

Monsieur Michel Adiba, le Responsable de la formation doctorale en informatique à Grenoble, pour m'avoir permis d'effectuer mon projet de DEA à Nancy,

Madame Brigitte Plateau, Monsieur Didier Bert, Monsieur Roland Balter, Monsieur Dominique Decouchant et Monsieur Khalid Benali, pour avoir bien accepté d'être les membres de mon Jury,

Madame Brigitte Plateau, ma tutrice de projet à l'ENSIMAG, qui m'a conseillé, encouragé et soutenu tout le long du projet,

Monsieur Roland Balter, mon encadrant à Grenoble, qui m'a soutenu, encouragé et suivi depuis Grenoble,

Monsieur Khalid Benali, mon encadrant à Nancy, qui m'a dirigé, soutenu et aidé afin d'aboutir à des résultats solides et rigoureux,

Monsieur Manuel Munier en PostDoctorat à Xerox Research Centre Europe à Grenoble pour le temps qu'il m'a consacré, pour les conseils et l'aide qu'il m'a fournis,

Les membres de l'équipe ECOO qui m'ont aidé et conseillé.

إِلَى أَهلي وَ أَصدِقَائِي ...
إلى القارى

A ma famille et mes amis ...
Au lecteur

Abstract

This report is an introduction to model a negotiation process and to implement it within a co–operative work system. Negotiation has been extensively discussed in a couple of fields for decades : game theory, economics, social–psychology and management science. However, the growing interest involved by distributed applications such as telecommuting, electronic commerce and co–operative work via network, has increased computerised negotiation importance. Co–operating actors need to negotiate each time they do not have sufficient capabilities or resources to complete their own tasks alone. Consequently, to propose a rigorous negotiation model allowing the development of a co–operative work specific service, the study of the various theoretical models of negotiation and human interactions in the everyday life becomes necessary. Actually, no model of negotiation can be put forward as being adaptable to each problem of negotiation. Several parameters are to be taken into account : the language and the vocabulary of the negotiation, the field of the negotiation, the nature of the conflict to be solved and finally the various media of communication.

The report is organised as follows. After the tackled problem presentation, chapter 2 exposes the state of the art, chapter 3 describes the proposed negotiation formal model, finally, chapter 4 presents the implementation.

Keywords: computer supported co–operative work (CSCW), virtual enterprises, telecommuting, concurrent engineering, negotiation process, distributed systems, negotiation support systems (NSS), decision support systems (DSS).

Résumé

Ce rapport présente d'une part la formalisation d'un modèle de négociation et d'autre part, sa mise en application partielle au sein d'un système de travail coopératif. La négociation a été traitée dans divers domaines : théorie des jeux, économie, psycho–sociologie et gestion, pendant des décennies. Cependant, l'explosion des applications réparties, telles le télétravail, le commerce électronique, le travail coopératif via le réseau, a donné une importance accrue à l'automatisation du processus de négociation. L'intérêt de la négociation est de permettre à chaque acteur de mieux coopérer avec un autre acteur chaque fois qu'il n'a plus les capacités ou les ressources suffisantes pour terminer sa tâche indépendamment des autres. Par conséquent, pour proposer un modèle rigoureux permettant le développement d'un service de négociation spécifique au travail coopératif, l'étude des différents modèles théoriques de la négociation et des interactions humaines dans la vie quotidienne s'avère nécessaire. En effet, aucun modèle de négociation ne peut se prévaloir d'être adaptable à chaque problème de négociation. En fait, plusieurs paramètres sont à prendre en compte : le langage et le vocabulaire de la négociation, le domaine de la négociation, la nature du conflit à résoudre et enfin les différents média de communication.

Ce rapport est organisé comme suit. Après la présentation de la problématique, le chapitre 2 expose l'état de l'art, ensuite, le chapitre 3 décrit le modèle proposé et enfin le chapitre 4 présente la réalisation du modèle.

Mots clefs: système d'aide au travail coopératif (CSCW), entreprises virtuelles, télétravail, ingénierie concourante, processus de négociation, systèmes d'aide a la négociation (NSS), systèmes d'aide à la prise de décision (DSS).

Table des matières

1 Introduction et Problématique **3**
1.1 Les Plates–formes d'ingénierie concourante . . . 3
1.2 L'environnement DisCOO . . . 4
1.3 La négociation dans l'environnement DisCOO . . . 5
1.4 Objectif du DEA . . . 6

2 Etat de l'art **9**
2.1 La psycho–sociologie . . . 10
2.2 La théorie des jeux . . . 10
2.3 L'économie . . . 11
2.4 L'Intelligence Artificielle Distribuée . . . 11
2.5 La théorie des actes de langage . . . 13
2.6 L'argumentation . . . 14
2.7 Les systèmes d'aide à la décision de groupe . . . 15
2.8 Synthèse . . . 16

3 Formalisation du modèle de négociation **19**
3.1 Critères de modélisation . . . 19
3.2 Position du modèle . . . 19
3.3 Le modèle générique de négociation . . . 20
3.4 Instanciation du modèle de négociation . . . 21
3.4.1 Approche transactionnelle . . . 21
3.4.2 Pour que DisCOO supporte un service de négociation . . . 22
3.4.3 La base de décision d'une négociation . . . 23
3.4.4 Les actes de négociation . . . 23
3.4.5 Le protocole de négociation . . . 24
3.4.6 L'effet de négociation . . . 26

4 Réalisation du service de négociation au sein de DisCOO **27**
4.1 Critères de réalisation . . . 27
4.2 Intégration du service de négociation à l'architecture de DisCOO . . . 28
4.2.1 Au niveau logique de DisCOO . . . 28
4.2.2 Au niveau de l'interface graphique de DisCOO . . . 29
4.3 Démonstration du service de négociation . . . 30
4.3.1 L'interface initiale de DisCOO . . . 30
4.3.2 L'interface actuelle de DisCOO . . . 31

Conclusion et Perspectives **39**

A Le formalisme ACTA **43**
A.1 Préliminaires 43
A.2 Dépendances Inter-Transactions 44
A.3 Conflits entre Opérations 44
A.4 Visibilité et Ensemble des Conflits 45
A.5 Exemples d'axiomatisation en ACTA 45
A.5.1 Dépendance de validation 45
A.5.2 Dépendance faible d'annulation 45
A.5.3 Commencement–sur–Annulation 45
A.6 Conclusion 45

Bibliographie **47**

Chapitre 1

Introduction et Problématique

La négociation a été traitée dans divers domaines : théorie des jeux, économie, psycho–sociologie et gestion, pendant des décennies. Cependant, l'explosion des applications réparties, telles le télétravail, le commerce électronique, le travail coopératif via le réseau, a donné une importance accrue à l'automatisation du processus de négociation. L'intérêt de la négociation est de permettre à chaque acteur de mieux coopérer avec un autre acteur chaque fois qu'il n'a plus les capacités ou les ressources suffisantes pour terminer sa tâche indépendamment des autres.

On s'intéresse dans ce projet de DEA, d'une part à la formalisation d'un modèle de négociation qui soit **générique** et **extensible**, et d'autre part à sa mise en application au sein d'une plate–forme d'ingénierie concourante. Afin de pouvoir proposer un modèle rigoureux permettant le développement d'un tel service de négociation, l'étude des différents modèles théoriques de la négociation et des interactions humaines dans la vie quotidienne s'avère nécessaire. En effet, aucun modèle de négociation ne peut se prévaloir d'être adaptable à chaque problème de négociation. En fait, plusieurs paramètres sont à prendre en compte : le **langage et le vocabulaire de la négociation**, le **domaine de la négociation**, la **nature du conflit à résoudre** et enfin les différents **média de communication**.

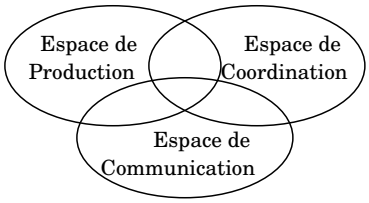
Pour mieux introduire la probématique et les objectifs d'une telle étude, on présente ci–dessous les plates–formes d'ingénierie concourante, ensuite, l'environnement de travail coopératif DisCOO et enfin on expose les besoins d'un meilleur service de négociation au sein de DisCOO.

1.1 Les Plates–formes d'ingénierie concourante

Par plate–forme d'ingénierie concourante, on désigne toutes les plates–formes logicielles permettant à un certain nombre de personnes de travailler en groupe (ex: concevoir un projet commun, résoudre un problème de décision, coéditer un document, etc.). Les systèmes de travail coopératif assisté par ordinateur CSCW[1] représentent l'un des piliers de l'ingénierie concourante. En effet, le CSCW est une discipline rassemblant différents domaines de recherche sur la même idée de "l'importance de la coopération". Les environnements de CSCW sont assez particuliers en comparaison avec les autres systèmes multi–utilisateurs. En effet, dans un environnement de CSCW, l'utilisateur est bien conscient de son appartenance à une communauté, tandis que l'un des objectifs primaires des environnements transactionnels classiques (ex : système de gestion de bases de données), était de permettre à plusieurs acteurs d'interagir simultanément avec le système en préservant l'illusion de systèmes mono-utilisateurs (cf. [Agr90]). Les services offerts par les systèmes de travail coopératif

1. CSCW: Computer–Supported Cooperative Work

(appelés aussi collecticiels) couvrent d'après [Ell94] trois espaces selon un trèfle fonctionnel :



1. *L'espace de production* désigne aussi bien les objets qui résultent d'une activité de groupe (ex: un livre, une œuvre de cinéma, un logiciel, etc.) que l'espace privé de chaque utilisateur du système,
2. *L'espace de coordination* définit les acteurs, les groupes, les rôles, les tâches (notamment leurs relations temporelles), les acteurs responsables des activités et des tâches. Alors que l'espace de production offre une vue statique du système, l'espace de coordination en définit la dynamique,
3. *L'espace de communication* offre aux acteurs du collecticiel la possibilité d'échanger de l'information. La sémantique de l'information échangée n'est pas perceptible par le système (simple messager), elle ne concerne que les acteurs communicants.

Selon les collecticiels, les trois concepts n'ont pas la même importance. *Les systèmes de co–édition de documents* mettent l'accent sur les services de production, *les systèmes à modèle de tâches (workflow)* insistent sur la coordination et enfin *les médiaspaces* sur la communication [Sal95].

1.2 L'environnement DisCOO

L'environnement de travail coopératif DisCOO [Ben98, Mun99], développé au sein de l'équipe ECOO[2] du LORIA, a pour objectif le contrôle des interactions au sein des *entreprises–projet*. Les entreprises–projet sont des entreprises virtuelles créées pour la mise en œuvre d'un projet, par plusieurs acteurs, pouvant être géographiquement éloignés, afin de créer et partager un ensemble de ressources (en l'occurence des fichiers: plans, programmes, articles etc.).

Par opposition à des systèmes existants tels que Continuus [CNT], ClearCase [All95, Atr94] ou Adèle-Tempo [Bel91], DisCOO n'adopte pas une architecture centralisée, ayant l'inconvénient de maintenir les clients continuellement connectés au serveur qui gère les contrôles de cohérence globale du système. Dans DisCOO, le problème de la coopération est abordé selon une approche transactionnelle avancée *(Transactions Coopératives Distribuées)* permettant de décentraliser la vérification de la cohérence, i.e. chaque transaction traite localement les critères de cohérence de ses interactions En effet, DisCOO encapsule chaque acteur de l'entreprise–projet à l'intérieur d'une *DisCOO–transaction*. Il introduit les notions de *référentiel local d'une transaction*, *d'histoire locale* des opérations effectuées par une transaction et *d'opérations de transfert entre transactions*. L'équipe ECOO n'est pas la seule à avoir opté pour un modèle transactionnel avancé: [Rod91] pense aussi que le modèle transactionnel classique, basé sur les propiétés ACID[3], s'avère inadapté aux environnements de travail de groupe, entre autres parce qu'il n'est pas centré utilisateur. En effet, si l'on considère un acteur, pouvant coopérer pendant plusieurs heures voir plusieurs jours, pour l'élaboration d'un document, il est inadmissible d'annuler tout son travail à cause d'un conflit instantané. La propriété d'Atomicité[4] n'est donc pas adaptée à des transactions longues. De plus, pour une meilleure coopération, chaque acteur a besoin d'être tenu au courant des versions partielles élaborées par les autres acteurs. La propriété d'Isolation[5] s'avère donc inadaptée à un comportement coopératif.

2. Environnements pour la COOpération
3. ACID: Atomicité–Cohérence–Isolation–et–Durabilité
4. Atomicité: aucune des opérations effectuées par une transaction n'est validée avant la validation de la transaction (le tout ou rien)
5. Isolation: échange interdit de résultat partiel entre deux transactions en cours d'exécution

DisCOO est basé sur trois piliers : *l'autonomie*, *la distribution* et *la coopération*. **L'autonomie** signifie que chaque acteur dans l'entreprise–projet possède un *espace de travail privatif* qui contient toutes les ressources en cours d'élaboration. Il dispose également d'un *espace de coopération* où il publie les ressources qu'il accepte de partager. **La distribution**, quant à elle, réside dans le fait que les contrôles de cohérence de chaque transaction sont traités localement, i.e. la vérification de la cohérence des interactions d'une transaction t ne concerne que la machine sur laquelle elle s'exécute sans pour autant transgresser une certaine cohérence globale du système. Sans oublier, que toute transaction gère ses propres données sans faire appel aux services d'un référentiel centralisé de données. Enfin, et afin d'assurer un travail coopératif avec consentement, le terme **coopération** signifie que si deux acteurs a_1 et a_2 souhaitent partager une certaine ressource r, ils doivent se mettre d'accord sur **un schéma de coopération** bien défini assurant la cohérence des données distribuées et des transactions les manipulant. Actuellement, les acteurs ont le choix entre trois schémas de coopération : **Ecriture_coopérative**, **Rédacteur/Relecteur** ou **Client/Serveur**, ce n'est pas à titre exaustif, on pourrait éventuellement en définir d'autres qui rapporteraient un plus au niveau de la coopération. Chaque acteur dispose d'une **table de coopération** (ensemble de triplets *(p:partenaire, r:ressources, s:schéma de coopération)*) indiquant ses droits et devoirs lors des échanges des ressources concernés par la coopération. Pour toute ressource r, partagée entre a_i et $a_{j\neq i}$ le schéma de coopéartion s la concernant correspondra à un prédicat vérifiant des propriétés de cohérence sur les opérations d'échange de r entre les deux acteurs. Le schéma *Ecriture_coopérative(r: ressource)* permet à deux acteurs de partager une ressource qu'ils élaborent en commun. Quant au schéma *Client/Serveur(r: ressource, a_1: acteur, a_2: acteur)*, il permet à deux acteurs a_1 et a_2 de partager la ressource r de telle manière que a_1 l'élabore en assurant que a_2 soit au courant de toutes ses mises à jour. Tandis que le schéma *Rédacteur/Relecteur(r_1, r_2: ressource)* permet à deux acteurs a_1 et a_2, de partager deux ressources r_1 et r_2, la première étant la ressource principale que a_1 élabore et la seconde contient les remarques qu'adresse a_2 à a_1 pour l'élaboration de la prochaine version de r_1. Dès qu'un schéma de coopération est négocié entre les deux acteurs, une dépendance est établie entre les opérations qu'ils effectuent. En effet, aucune DisCOO–transaction ne peut être validée (resp. annulée) sans que les autres transactions dont elle dépend à cause des schémas de coopérations négociés ne soient également validées (resp annulées). En fait, ce sont ces schémas de coopération qui distinguent le système de travail coopératif DisCOO, par sa sûreté et fiabilité, des utilitaires d'échange de documents offerts par les systèmes d'exploitation permettant trop de liberté et n'assurant pas toujours des possibilités de reprise.

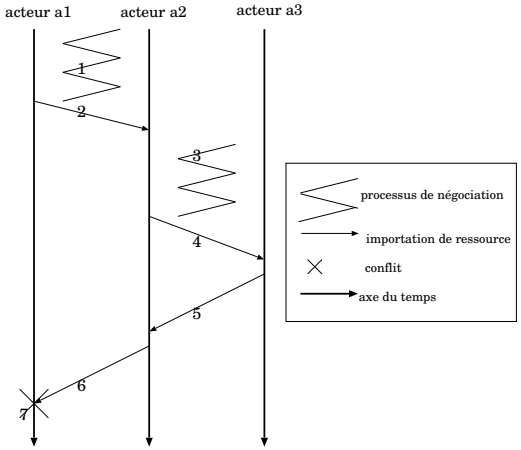
1.3 La négociation dans l'environnement DisCOO

Comme introduit dans la section précédente, chaque acteur a_i, voulant coopérer avec un acteur $a_{j\neq i}$ pour partager un ensemble de ressources ou une ressource particulière, déclenche une phase de négociation. Le service de négociation, tel qu'il figure dans la version actuelle de DisCOO peut être décrit comme suit :

1. a_1 propose, à un partenaire a_2, un schéma de coopération s pour un ensemble de ressources R.
2. a_2 est notifié de la proposition, il a la possibilité de l'accepter ou de la refuser,
   (a) si a_2 accepte la proposition alors la négociation réussira et les tables de coopération des deux négociateurs seront automatiquement mises à jour,
   (b) sinon, la négociation échouera.

Ayant validé le processus de négociation, les deux négociateurs a_1 et a_2 peuvent commencer à s'échanger les versions intermédiaires des ressources R suivant les règles fixées par le schéma de coopération négocié. Durant les échanges, il peut éventuellement se produire des conflits liés aux dépendances entre les ressources

partagées [Mun99]. Le scénario suivant présente une situation conflictuelle :



Conflit de partage de ressources entre acteurs de DisCOO

1. *Si* **Rédacteur/Relecteur**($a_1.doc$, $a_2.revue$) est le schéma de coopération résultant de la négociation entre a_1 et a_2, des ressources *doc* de a_1 et *revue* de a_2, i.e. a_2 sera un relecteur de la ressource *doc* de a_1 et lui fournira en contre–partie des remarques dans la ressource *revue*.
2. a_2 importe *doc* de chez a_1,
3. **Ecriture_coopérative**(*doc*) est le schéma résultant de la négociation entre a_2 et a_3 de la ressource doc de a_2, i.e. les deux acteurs partageront la ressource *doc* qu'ils élaboreront en commun.
4. a_3 importe *doc* de chez a_2,
5. ensuite a_2 importe *doc* de chez a_3 et prend en compte cette version de *doc* pour rédiger sa ressource *revue*,
6. et enfin a_1 essaie d'importer *revue* de chez a_2,
7. *Alors* un conflit sera détecté puisque a_1 aurait voulu importer la revue concernant un document différent du sien.

Ceci montre qu'audelà d'un processus de négociation satisfaisant les parties prenantes, il se peut que des conflits, qui ne peuvent pas être prévus à l'avance[6], se produisent et obligent par conséquence les partenaires de l'entreprise–virtuelle à **renégocier** leurs schémas de coopération (ex: le schéma **Ecriture_coopérative**(*doc*) devrait plutôt être renégocié en **Client_Serveur(doc,** a_2**,** a_3**)**). Bien que le processus de **renégociation** soit assez lié au processus de **négociation**, il ne nous concerne pas dans cette étude. Sa formalisation et sa réalisation pourrait faire l'objet d'une étude future.

1.4 Objectif du DEA

La prise de décision de groupe est parmi les tâches les plus difficiles auxquelles les cadres consacrent 30 à 70% de leur temps [Ste87]. La prise de décision représente un goulot d'étranglement considérable pour la coopération du fait qu'elle est indispensable, entre autres, pour attribuer des tâches, évaluer une situation globale, coordonner des travaux interdépendants, décider d'un plan d'actions etc. Les utilisateurs d'un environnement de travail coopératif, pouvant être géographiquement éloignés, ont naturellement besoin

6. il n'est pas dit la négociation de *doc* entre a_2 et a_3 en **Ecriture_coopérative**(*doc*) obligerait a_2 à importer *doc* de chez a_3 et donc à donner une sémantique à la ressource *revue* différente de sa sémantique initiale

de mécanismes d'aide à la négociation afin d'étudier les alternatives possibles pour l'établissement des choix décisionnels communs. Chaque membre de l'équipe, au travers de son rôle, sa responsabilité et sa spécialité, participe au processus de négociation en exposant les problèmes à résoudre, les règles à suivre, les contraintes à respecter, les tâches à remplir, les perspectives à viser, les moyens à déployer, etc. Un processus de négociation peut concerner tous les membres de l'équipe comme il peut en concerner qu'une partie. Les interventions ne sont pas unilatérales, elles émanent des différents participants sous forme de demande, proposition, critique, confirmation, argumentation, interrogation, reformulation etc.

L'objectif poursuivi par l'intégration d'un service de négociation à un système d'aide au travail coopératif est de permettre à chaque acteur de coopérer avec la possibilité d'exprimer ses propositions décisionnelles au sein du groupe. En l'occurence, un tel service pourrait offrir un cadre confortable assistant le groupe dans les situations suivantes :

- décider des utilitaires à mettre en place pour élaborer les ressources partagées (logiciels de dessin, éditeurs de textes etc.),
- décider de l'ensemble de ressources à partager et avec quels partenaires,
- décider suivant quels schémas de coopération, l'on va partager chaque ressource,
- décider, de la dernière version à conserver pour une certaine ressource (dans le cas d'une écriture coopérative),
- décider de la manière de fusionner deux ressoures (voir plusieurs ressources) pour en avoir une version finale,
- etc.

La négociation a été traitée dans le cadre de diverses thématiques de recherche : la psycho–sociologie, la théorie des jeux, l'économie, l'intelligence artificielle distribuée, etc. Actuellement, l'explosion des applications réparties, telles le télétravail, le travail coopératif (CSCW), la téléconférence, le commerce électronique etc. a donné une importance accrue aux systèmes de négociation assistée par ordinateur (NSS[7]) [Jel89, Lim93].

Dans un premier temps, on étudiera les résultats de recherche sur la négociation pouvant servir à spécifier les concepts propres à un système d'aide à la décision et à la négociation. Ensuite, on essaiera d'offrira un cadre théorique formalisant un processus de négociation qui soit **générique** et **extensible**. Puis, en se basant sur le modèle théorique, on concevra un service d'aide à la négociation **extensible**, **fiable**, **portable** et **centré utilisateur** qui sera instancié au sein de l'environnement de travail coopératif DisCOO.

Ce rapport est organisé comme suit. Le chapitre 2 expose l'état de l'art, ensuite le chapitre 3 décrit notre modèle formel de négociation et enfin le chapitre 4 présente la réalisation de ce modèle.

7. NSS: Negotiation Support Systems

Chapitre 2

Etat de l'art

Selon [Spe82] la négociation a été décrite comme étant la science des observations précises, des prétentions réalistes, de l'analyse cohérente des faits, des conclusions logiques, des comportements planifiés et de la prévention optimale contre tout changement de la situation. Etant un processus de communication entre agents, la négociation sert à recadrer les objectifs en conflit, identifier et structurer les différents problèmes, proposer, attaquer et défendre les solutions et enfin atteindre et confirmer les accords [Put87]. [Pru81] présente la négociation comme étant un processus bilatéral ou multilatéral servant à prendre une certaine décision. Les parties commencent par formuler des propositions contradictoires et tendent vers un accord en réalisant des concessions et en recherchant de nouvelles alternatives. La négociation peut être divisée, selon [Jel89], en deux catégories. Elle peut être soit *douce* soit *dure*. La *négociation douce* (ou intégrative) est caractérisée par une situation où les négociateurs ont plusieurs solutions possibles qu'ils adaptent selon leurs contraintes et leurs objectifs afin d'atteindre un état de satisfaction commun *(win-win)*. La *négociation douce* caractérise des comportements coopératifs comme c'est le cas dans les conflits organisationnels où, certes les parties peuvent avoir des opinions opposées, cependant, l'intérêt général (imposé par le système organisationnel) est dans une solution bénifique à toutes les parties. D'un autre côté, *la négociation dure* (ou distributive), rassemble des négociateurs voulant renforcer leur position sans faire de concession *(win-lose)*. La négociation dure est caractérisée par le fait que (1) les objectifs des négociateurs sont en conflit direct, (2) les ressources sont fixées et limitées (ex: temps, mémoire, terrain, part de marchés etc.) et (3) chaque partie vise la maximisation de sa part des ressources. Une telle négociation se produit dans les conflits patronat/salariés, dans les conflits de contrôle d'armes, dans les processus de paix etc.

Il n'existe pas de modèle universel de négociation qui puisse prévaloir pour chaque problème de négociation, à cause de paramètres mettant en jeux la culture des différents négociateurs, le langage et le vocabulaire qu'ils emploient pour négocier, le domaine de la négociation, le médium de la communication, sans parler des problèmes spécifiques à la situation de négociation. De plus, la complexité du processus de négociation est en fonction du nombre de négociateurs, du nombre de coalitions possibles parmi les négociateurs, de la nature du conflit, du rapport de force entre les négociateurs, de la perception du conflit à résoudre, des objectifs à atteindre etc.

Plusieurs disciplines ont analysé le processus de négociation selon deux approches différentes. L'approche *descriptive* vise l'analyse des interactions et des échanges entre les négociateurs, l'approche *perspective* s'intéresse quant à elle à expliquer les intérêts et les objectifs de chaque partie prenante de la négociation.

Dans ce qui suit, sera analysé le processus de négociation selon les disciplines suivantes :

1. la psycho–sociologie,
2. la théorie des jeux,
3. l'économie,

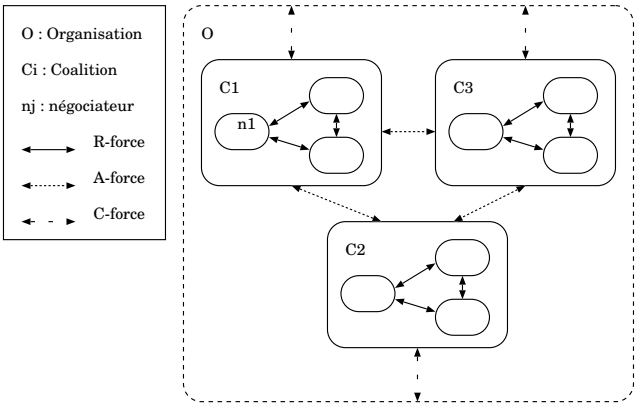
4. l'Intelligence Artificielle Distribuée (IAD)
5. la théorie des actes de langages,
6. l'argumentation,
7. et les systèmes d'aide à la décision de groupe (GDSS) [1].

2.1 La psycho–sociologie

Pour analyser l'état psychologique du négociateur, la psycho–sociologie souligne trois types de forces qui influencent son comportement vis à vis des parties opposées [Mor77] :

1. Les *R–forces* : représentant son lien à la position de la coalition de négociateurs à laquelle il appartient,
2. Les *A–forces* : indiquant le degré de sympathie qu'a le négociateur envers la position de la coalition opposée,
3. Les *C–forces* : régissant le pouvoir du système organisationnel ou social, où coexistent toutes les coalitions, à imposer une issue finale au conflit.

Selon ce modèle, le succès d'une négociation est mesuré par le degré d'équilibre des trois forces.



Le modèle des forces

2.2 La théorie des jeux

Pour la théorie des jeux, un conflit impliquant un ensemble de négociateurs (joueurs) consiste à choisir une possibilité parmi un ensemble d'alternatives. Les choix peuvent mener les joueurs à des situations conflictuelles sans pour autant leur interdire de coopérer [Jel89].

Dans [Mye89] et [Mye91], Myerson a essayé de formaliser la notion de *crédibilté d'une négociation* en termes de "défendabilité", "fiabilité" et "plausibilité" *(tenability, reliability and plausibility)*. Il a ainsi défini la négociation comme étant une *prétention* décrivant les informations privées d'un négociateur, une *promesse* décrivant les futures actions et messages d'un négociateur et une *requête* décrivant les stratégies, dont un négociateur souhaite la réalisation chez d'autres joueurs.

La théorie des jeux fait une analyse perspective de la négociation (contraintes et fonction d'utilité) en étant très limitée si l'on considère les interactions possibles entre négociateurs [Lim93].

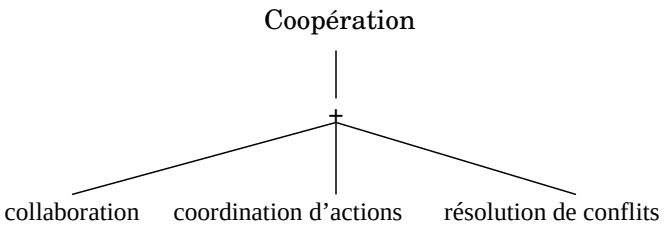
1. Group Decision Support Systems

2.3 L'économie

Selon [Lim93], la plupart des modèles de négociation établis par l'économie dépassent le simple cadre des problèmes spécifiques tels la détermination des grilles de salaires dans les conflits patronat/salariés ou les décisions de prix dans les échanges commerciaux. La dynamique de ces modèles paraît dans le fait qu'ils analysent la négociation aussi bien du point de vue descriptif que du point de vue des perspectives. Parmi les modèles de négociation les plus connus, le modèle de [Zeu68] affirme que chaque négociateur, en fonction de sa startégie, évalue ses gains et ses pertes à chaque étape de la négociation avant d'afficher sa décision. Il a ainsi présenté un modèle de négociation à deux étapes qui, comparé à un second groupe de modèles économiques reste incapable de prédire le résultat du processus de négociation. En effet, [Pen52] reconnait qu'il n'est pas raisonnable de supposer que les conditions d'équilibre des deux négociateurs seraient satisfaites par la même valeur d'un prix ou d'un taux de salaire. Il pense qu'il est indipendsable d'avoir un mécanisme servant à transformer les conditions d'équilibre de l'un ou des deux négociateurs afin d'assurer un éventuel accord sur le prix ou le taux de salaire. Certes, Pen n'a pas donné de réponse permettant de générer une solution bien déterminée de la négociation, mais, il a joué un rôle important dans la conceptualisation de ses aspects.

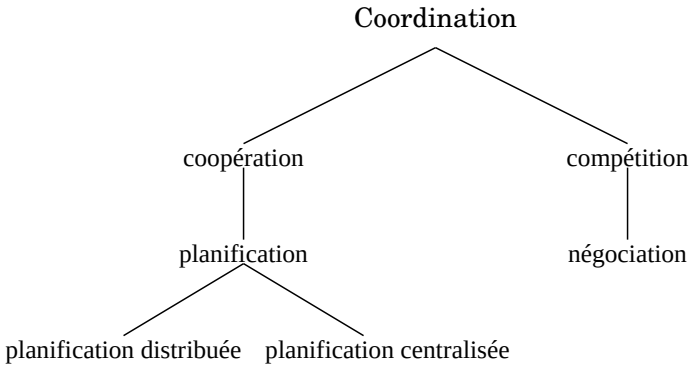
2.4 L'Intelligence Artificielle Distribuée

L'Intelligence Artificielle Distribuée est la discipline qui a pour objectif de réaliser des organisations de systèmes (ou agents) capables de résoudre des problèmes par le biais d'un raisonnement le plus généralement fondé sur une manipulation de symboles [Fer95]. Pour l'IAD, un agent est une unité computationnelle particulière qui a un état, développe une expertise et qui est capable de réagir à des événements externes. [Dem91] distingue entre agents cognitifs et réactifs selon leurs capacités de raisonnement et de représentation de connaissances. Selon [Fer95], pour un agent, interagir avec un autre constitue à la fois la source de sa puissance et l'origine de ses problèmes. C'est en effet, parce qu'ils coopérent que les agents peuvent accomplir plus que la somme de leurs actions, mais c'est aussi à cause de leur multitude qu'ils doivent coordonner leurs actions et résoudre des conflits... Une interaction a deux formes principales, la collaboration et la coordination d'actions. La première s'intéresse à la manière de répartir le travail entre plusieurs agents et la seconde analyse la manière dont les actions des différents agents doivent être organisées dans le temps et l'espace de manière à réaliser les objectifs. Ferber résume sa thèse à propos de la coopération par la formule :



Par ailleurs [Huh99] donne la taxonomie suivante montrant quelques possibilités de coordination des

comportements et tâches entre agents :



Enfin, lorsque des conflits apparaissent, il est important de pouvoir en limiter les effets. Les techniques de négociation servent ainsi à satisfaire les parties impliquées en établissant des compromis ou en dépassant la nature du conflit. La spécification de telles techniques suivant un protocole de négociation entre agents (communicants) utilisera *un langage de communication* dont le rôle est d'assurer les interactions par le biais de transmissions d'informations et de demandes mutuelles de renseignements et de services.

Pour que deux agents différents puissent négocier afin de résoudre un conflit, ils doivent avoir une connaissance du monde dans lequel ils évoluent et des autres agents de ce monde. C'est pour cela, que des cadres théoriques tels les agents BDI [2] essaient de représenter les modèles mentaux des agents en s'intéressant à la schématisation de leurs Croyances, Désirs et Intentions (cf. [Kra98, Par98]). D'autres recherches ont traité une classe particulière de négociation : *la négociation orientée service*. Dans ce contexte, un agent (client), ayant besoin qu'un service lui soit fourni par un autre agent (serveur), doit entrer en négociation avec ce dernier afin de déterminer un contrat sous certains termes et conditions. Un service est une activité spéciale de résolution de problème pour laquelle les points de départ et d'arrivée sont clairement définis. Un exemple de demande de services peut concerner en l'occurence le diagnostic d'une défaillance, l'achat d'un groupe de parts d'un marché, l'allocation d'une bande passante pour la transmission de vidéo–conférence etc. Dans le sens de ce type de négociation, [Far97] définit une tactique comme étant une fonction qui aide un négociateur à déterminer sa proposition en termes de valeurs de variables (prix, volume, durée, qualité, ...) en considérant un critère unique (temps, ressources, ...), sa stratégie étant de combiner le mieux possible ses tactiques afin d'optimiser sa fonction d'utilité. Il définit pour ce faire trois familles de tactiques :

1. les tactiques dépendant du temps,
2. les tactiques dépendant des ressources,
3. et les tactiques imitatives.

Si un agent doit obtenir une réponse avant un délai de rigueur, une tactique de la première classe modélisera le fait que plus le délai approche et plus rapidement l'agent fera des concessions. Les tactiques de la deuxième classe, quant à elles, modélisent la pression imposée par les ressources limitées (bande passante à allouer, budget etc.) et par l'environnement (nombre de clients ou de serveurs, des paramètres économiques etc.). Quant aux tactiques imitatives, elles interviennent dans des situations où l'agent n'entre pas dans l'une des deux premières catégories. Dans ce type de situations, les contre–propositions qu'un agent peut formuler ne dépendent que des propositions qu'il reçoit ; sa seule préoccupation étant d'assurer sa protection contre les autres agents. La recherche en IAD et en conception orientée agents a permis de faire avancer énormément

2. BDI: Believe, Desire and Intention

l'étude du processus de négociation.

Certes, l'intelligence artificielle distribuée est l'une des disciplines qui a énormément traité les problèmes de conflits entre agents et leurs résolutions par le biais de modèles de processus de négociation très riches si l'on considére les concepts qu'ils mettent en œuvre : *réseaux contractuels de distribution de tâches* (Contract Net Protocol [Smi80] et Commitment–Based Communication Model [Koo88]), *mécanismes de négociation par apprentissage* (Bazaar [Zen97]), *médiation ou arbitrage* (Persuader [Syc89]), sans oublier les travaux réalisés dans la structuration et l'enrichissement du langage de communication entre agents (cf. les sections 2.5 et 2.6).

Cependant, les modèles de l'IAD ne sont pas adaptés à la conception d'un service de négociation au sein d'un environnement de travail coopératif (ou CSCW). En effet, l'intégration de tout service à ce type d'environnements doit se baser sur une conception orientée utilisateur[3]. Or, les modèles de processus de négociation définis par l'IAD concernent des agents artificiels (BDI) dont le comportement est géré par des modèles deterministes qui ne supportent pas des agents humains (utilisateurs). Par conséquent, l'intérêt d'avoir étudier les résultats de recherche en IAD a été de découvrir une riche palette de concepts intéressants à mettre en pratique dans un service d'aide à la négociation au sein d'un environnement de travail coopératif.

2.5 La théorie des actes de langage

A travers la théorie des actes de langage (*speech acts*), appelée aussi *"pragmatique du discours"*) développée initialement par Austin et Searle [Aus62, Sea69, Sea75], la linguistique a donné une nouvelle dimension à la modélisation des protocoles de communication symbolique point–à–point dans les applications informatiques. En effet, la théorie des actes de langage n'importe pas seulement en Linguistique : elle a des applications nombreuses dans les sytèmes distribués, le traitement des langues naturelles et dans les protocoles d'échange de données électroniques [Sin93]. En particulier, la plupart des approches d'IAD formalisant la communication entre agents sont fondées sur cette théorie [Kon95]. De plus, [Rod91] évoque la théorie des actes de langage parmi cinq approches de représentation et de contrôle des systèmes d'aide au travail coopératif (CSCW).

L'idée principale de la théorie des actes de langage est que quelqu'un qui prononce une phrase ne fait pas que dire des mots, mais qu'il essaie de changer l'état du monde [Lev83]. En effet, pour une certaine phrase, cette théorie distingue entre les aspects :

- *locutoires* (contenu du message),
- *illocutoires* (effet attendu du message (l'acte)),
- et *perlocutoires* (effet réel sur l'auditeur).

Prenons l'exemple décrit dans [Chi98a] d'une personne qui dit "le ciel est bleu". La dimension locutoire est l'association d'une partie du ciel à une couleur, la dimension illocutoire est une assertion et la dimension perlocutoire peut être, suivant le contexte, la transmission de l'information, une réponse à une question, un refus d'aller au cinéma, une incitation à la randonnée... [Pop93] propose d'exprimer une interaction comme la combinaison du contenu propositionnel P et d'une force d'élocution F exprimant le contenu intentionnel de l'acte de langage.

F	**F(P = "the light is on")**
Ask	"is the light on?"
Inform	"the light is on"
Request	"switch on the light, please"

3. La conception d'un système orienté utilisateur se base essentiellement sur un modèle utilisateur qui sera pris en considération pendant toutes les étapes du développement (cf. la théorie de l'action de [Nor86])

Parmi les apports importants de la théorie de langage on peut citer *une classification des actes illocutoires* en sept catégories, (cf. [Sin93] et [Fer95]) :

1. *prohibitifs* interdisant l'accomplissement d'une action (ex: tu ne dois pas veiller le soir, tu n'a pas le droit de voir le contenu de ce document) ;
2. *permissifs* annonçant une permission (ex: tu peux ouvrir la fenêtre, ta présence à cette réunion n'est pas obligatoire) ;
3. *assertifs* servant à donner une information sur le monde en affirmant quelque chose (ex: il fait beau, Jean a 23 ans, les rectangles ont quatre angles droits) ;
4. *directifs* utilisés pour donner des directives au destinataire (ex: donne moi ta montre, lave–toi les mains, quelle est la valeur de la troisième décimale de π) ;
5. *promissifs* engageant le locuteur à accomplir certains actes dans l'avenir (ex: je viendrai à la réunion de 5 heures, je te promets de t'envoyer des cartes postales) ;
6. *expressifs* servant à donner au destinataire des indications concernant l'état mental du locuteur (ex: je suis heureux, je m'excuse pour hier, je te remercie) ;
7. et *déclaratifs* accomplissant un acte par le fait même de prononcer l'énoncé (ex: je déclare la séance ouverte, je vous déclare mari et femme).

Pour formaliser le processus de négociation, cette classification pourrait servir à distinguer les messages échangés par leur type soit en émission ou en réception. Pour la simplification des traitements des interactions de négociation, on utilise souvent des diagrammes états/transitions dont les transitions sont des étiquettes avec des actes de langage ou des règles événement/condition/action dans lesquels un événement correspond en l'occurence à la réception d'un acte de langage et une action désigne la réaction de l'agent à cet acte, émission d'un autre acte de langage (ex: selon [Chi98b], il est plus courant de répondre à un *directif* par un *assertif*). On pourrait aussi définir une relation d'ordre partielle $\preceq_A$ entre les actes illocutoires de même catégorie, tel que $a_1 \preceq_A a_2$ signifie que a_1 est un acte moins puissant que a_2, (ex: l'assertif "demander" $\preceq_A$ l'assertif "insister") (cf. Information lens [Mal89] et SANP [Cha94]).

2.6 L'argumentation

En logique classique, un argument est une séquence d'inférences aboutissant à une conclusion vraie. Par ailleurs, ce qu'on entend par argument dans un système d'argumentation est un élément qui, non seulement, prouve que des propositions sont vraies ou fausses mais aussi informe que des propositions peuvent être vraies ou fausses [Par98]. L'argumentation est un concept sur lequel travaillent différentes disciplines de recherche : la théorie de décision, les logiques non–monotones, le raisonnement par contraintes, la représentation des connaissances etc. [Kar97b]. Certes, les modèles de l'argumentation ont été axiomatisés, dans la plupart des cas, dans l'objectif d'ajouter la dimension de persuasion au dialogue entre agents cognitifs en IAD [Gor97, Kra98, Par98]. Cependant, intégrée aux environnements de planning et de prise de décision de groupe (GDSS cf. la section 2.7), l'argumentation présente un très grand intérêt. Selon le modèle des agents BDI, l'argumentation est un processus itératif d'échanges de messages servant à un agent à persuader un autre agent pour changer ses intentions et donc ses actions qui en dépendent... La possibilité de changer les intentions des agents par argumentation devrait rendre les agents plus coopératifs [Kra98]. En effet, pour mener à bien une négociation, les négociateurs doivent argumenter leurs croyances et objectifs. Ainsi, les autres parties prenantes de la négociation pourront comprendre la logique de tout acte de négociation. Dans [Kra98], Kraus distingue entre six types d'arguments :

1. menacer pour imposer ou éliminer une idée,

2. promettre une future récompense,
3. faire appel à une ancienne récompense,
4. faire appel aux actions passées pour montrer une contradiction avec les actions courantes,
5. faire appel aux *pratiques prédominantes* pour montrer que la prise en compte d'une action éloignera d'un certain objectif comme elle a éloigné d'autres objectifs dans le passé,
6. faire appel à l'intérêt personnel d'autrui pour convaincre que la prise en compte d'une action va dans le sens d'un objectif important.

L'article [Par98] propose un modèle générique de négociation, où un argument construit par inférence peut être joint à toute proposition, critique, acceptation ou annulation lors de la négociation. Selon [Kar97b], l'intérêt de l'argumentation est de permettre aux négociateurs de gérer des informations incomplètes, incertaines ou inconsistantes sans parler de la richesse de l'expression qu'elle implique. Les arguments doivent être structurés de telle manière à ce qu'ils aident à identifier les conflits tout en inspirant aux négociateurs des débuts de solutions [Cha94]. Par ailleurs, si l'on considère que le succès d'une négociation réussie est une fonction décroissante du nombre d'actes échangés et si l'on suppose que l'argumentation éliminera les actes redondants ou incohérents, alors le mécanisme d'argumentation, intégré à un service d'aide à la négociation, fera converger plus rapidement le processus de négociation et donc augmentera son succés.

La formalisation des arguments et des relations les liant dans un processus de négociation a été traitée selon différentes méthodes et structures. Parmi les structures logiques servant à l'évaluation et la génération des arguments pour un traitement automatisé de la négociation on peut citer : le raisonnement à base de cas CBR[4] [Syc89] (Persuader) et [Kar97b], les régles d'inférence [Par98], etc. Par ailleurs, parmi les structures graphiques de représentation de l'argumentation accompagnant une négociation, on a les graphes d'ordre (cognoter, argnoter [Mal89], les graphes dialectiques (Zeno [Gor97]), les structures hypertextes (Hermes [Kar97a]) etc.

2.7 Les systèmes d'aide à la décision de groupe

Les systèmes d'aide à la décision de groupe (GDSS)[5] apportent aussi bien une assistance automatisée que semi–automatisée, dans des situations de conflits de perception d'un problème ou concernant des objectifs opposés. Combinant les techniques informatiques, la communication et les techniques d'aide à la décision, les GDSS ont pour unique but de faciliter à un groupe de personnes la formulation et la résolution de problèmes non structurés [Des87]. En effet, la prise de décision est une tâche énormément coûteuse en temps : [Pan92] rapporte que les managers consacrent 85% de leur temps à communiquer par différentes façons (contact direct, par téléphone, par écrit) dont 55% à communiquer directement (70% pour les cadres).

On s'intéresse plus particulièrement dans notre étude aux systèmes d'aide à la négociation (NSS)[6] qui représentent, selon [Jel89], une sous classe des GDSS. Lim et Benbasat rapportent, selon [Zac86], qu'un système d'aide à la prise de décision peut offrir six genres de supports [Lim93] :

1. *des modèles de processus* servant à prédire les processus complexes,
2. *des modèles de choix* permettant d'intégrer des critères personnalisés à travers des choix alternatifs (ex: les modèles d'utilité multi–attributs/multi–alternatives),
3. *des techniques de contrôle d'informations* gérant la base des données concernées par la décison,
4. *des supports de représentation* regroupant les méthodes et les structures de représentation visuelle du probléme de décision,

4. CBR: Case–Based Reasoning
5. GDSS: Group Decision Support Systems
6. NSS: Negotiation Support Systems

5. *des aides d'analyse et de raisonnement* offrant des mécanismes de traitement mathématique des données de décision (ex: systèmes à base de connaissance, programmation logique),

6. *des techniques de raffinement/amplification des jugements* interprétant et évaluant les jugements en termes de fonctions heuristiques d'utilité.

Deux types de services peuvent être assurés par les environnements d'aide à la décision et à la négociation, les services synchrones et asynchrones. Les services synchrones mettent en place des mécanismes de rendez-vous, nécessitant "la présence" des différents acteurs avant que la négociation ne puisse débuter (ex: la téléconférence, les salles de réunion à réalité augmentée, les systèmes de réunion électronique (EMS[7])). Les services asynchrones, quant à eux, ne préparent pas les acteurs à une prétendue "session de négociation", mais ils offrent un cadre plus confortable de négociation où les acteurs peuvent intéragir librement dans le temps tout en effectuant leur travail habituel (ex: la messagerie électronique ou la communication homme–homme médiatisée (CMC[8])).

On estime que les services synchrones de décision de groupe ne sont pas adaptés au service d'aide à la négociation qu'on veut réaliser du fait qu'il nécessitent un rendez–vous entre les différents interlocuteurs. Or, les services asynchrones d'aide à la décision de groupe ne posent aucune contrainte temporelle sur les interactions de ses usagers. L'intégration d'un service asynchrone d'aide à la décision de groupe au sein d'un environnement de CSCW n'est cependant pas systématique. En effet, les systèmes asynchrones d'aide à la négociation existants sont soit trop génériques : donnant une liberté expressive qui n'est pas forcément permise au sein du système de travail coopératif (ex: SANP [Cha94]), soit trop particuliers : servant à résoudre certains types spécifiques de problèmes (ex: Information lens [Mal89]). Tout cela, sans évoquer les conflits d'interoperabilité venant du fait que l'environnement de travail coopératif et celui d'aide à la négociation ne manipulent pas les mêmes concepts.

2.8 Synthèse

Ayant comme objectif de formaliser un processus de négociation qui soit **générique** et **extensible** afin de réaliser un service d'aide à la négociation dans un environnement de travail coopératif (CSCW), on peut affirmer que les disciplines étudiées, bien qu'elles soient très intéressantes, n'offrent pas de cadre totalement adapté au CSCW.

Si l'on veut exposer les résultats obtenus de notre étude bibliographique, on les résumera en une palette très riche en concepts, modèles et méthodes à mettre en pratique dans le service de négociation à réaliser. En effet, la **psycho–sociologie** nous aide à mieux analyser le comportement du négociateur au sein d'une organisation et à mieux comprendre les forces qui l'y influencent. La **théorie des jeux** impose les notions de *bilateral turntaking (prise de tour), tactiques, stratégies, situations d'équilibre, etc.* L'**économie** nous offre les notions de *fonctions d'utilité, solution optimale, etc.* L'**IAD**, développe des *méthodes de conception orientée agents, modèles formels de négociation entre agents artificiels, notions de médiation et délégation, etc.* La **théorie des actes de langage** quant à elle, étudie les *classifications de verbes de dialogue.* L'**argumentation**, à son tour, offre des *modèles formels des arguments dans une négociation.* Et pour finir, les **systèmes d'aide à la décision de groupe** (GDSS) détaillent les *problèmes spécifiques à la co–décision, besoins en service d'aide à la co–décision, architectures d'un service de négociation, etc.*

Pour justifier le fait que les disciplines étudiées n'offrent pas de cadre totalement adapté à la formalisation d'un service de négociation au sein d'un environnement de travail coopératif (CSCW), on peut annoncer que la plupart des spécifications formelles des services de négociation traitent une classe très particulières de

7. EMS: Electronic Meeting Systems
8. CMC: Computer–Mediated Communication

négociation : "la négociation automatisée entre agents artificiels (BDI)". Or, dans les environnements CSCW, les acteurs sont des agents humains dont on n'a pas de modèle formalisant les croyances, désirs et intentions, d'une part. D'autre part, la négociation au sein d'un environnement CSCW n'est pas censée toujours admettre des mécanismes de sélection de la meilleure solution dans l'**espace des solutions** possibles prétendu ordonné selon une certaine fonction d'utilité. De plus, les protocoles de négociation entre agents artificiels sont généralement représentés par des graphes état/transitions dont les transitions se basent sur des résultats d'évaluation numérique ou logique de l'état courant de l'agent ce qui n'est pas possible dans toutes les situations de négociation. On peut toutefois signaler que *la conception orientée agents* pourrait être très intéressante si l'on voyait un agent comme une unité médiatrice entre l'utilisateur et le système. On ne parlerait alors plus d'agent artificiel cognitif mais d'**agent réactif**.

Par ailleurs, la conception de la plupart des services de négociation étudiés ne distingue pas **l'objectif de la négociation** du **protocole de négociation**. En d'autres termes, les mécanismes de négociation sont généralement "codés en dur" parmi d'autres mécanismes dépendant du domaine d'application. Notre objectif étant de formaliser et de réaliser un modèle de négociation qui soit générique et extensible, il faudra effectuer un travail conceptuel qui découple les **aspects génériques de négociation** des **aspects du domaine d'application**.

De plus, les modèles de négociation existants proposent souvent un **langage de négociation** fixé par le protocole de négociation empêchant par ce fait toute **extensibilité du modèle**. Ce problème peut être résolu par la théorie des **actes de langage**, qui offre un cadre intéressant quant à la spécification de langages de négociation qui soient complets et consistants d'une part et qui simplifient les traitements des **messages échangés** entre les négociateurs d'autre part. Tout cela, sans oublier l'importance des **modèles d'argumentation** axiomatisés, dans la plupart des cas, dans l'objectif d'enrichir l'expressivité du dialogue entre les agents cognitifs (BDI), et qui, intégrés aux environnements d'aide à la prise de décision de groupe (GDSS), présentent un très grand intérêt si l'on considére la dimension de **persuasion** qu'il ajoutent aux actes de langage échangés entre les négociateurs.

Quant aux systèmes d'aide à la prise de décision de groupe, on estime que **les services asynchrones** sont les plus adaptés au service d'aide à la négociation qu'on veut réaliser. En effet, les services synchrones d'aide à la prise de décision de groupe nécessitent un rendez–vous entre les différents interlocuteurs ce qui les contraint à "répondre toujours présents" quand un partenaire les demande en négociation. Ceci est trop contraignant du point de vue des environnements de travail coopératif où les acteurs sont supposés travailler normalement tout en ayant la possibilité d'initier une négociation ou de répondre à une autre.

Dans ce qui suit, on va essayer de **formaliser un modèle générique et extensible de négociation** en prenant en compte quelques concepts parmi ceux présentés dans ce chapitre. Pour formaliser ce modèle de négociation, on adoptera une approche transactionnelle basée sur le formalisme ACTA (cf. annexe A) afin de rester dans le même cadre de formalisation que les DisCOO–Transactions (cf. [Mun99]) formalisé également en ACTA. Ensuite on essaiera de **réaliser une partie du modèle au sein de l'environnement de travail coopératif DisCOO**.

Chapitre 3

Formalisation du modèle de négociation

Après avoir fait le tour des différentes disciplines de recherche ayant traité le processus de négociation selon diverses approches et méthodologies, on va essayer d'englober les concepts de base d'un service d'aide à la négociation dans un **modèle théorique souple et évolutif**.

Dans un premier temps on présentera la position du modèle et on posera les définitions nécessaires à la formalisation du modèle. Ensuite, on détaillera l'intégration du concept de service de négociation et ses implications au sein de DisCOO, et enfin, on instanciera le modèle théorique dans le but d'aboutir à un **service de négociation de schémas de coopération operationnel au dessus de DisCOO**.

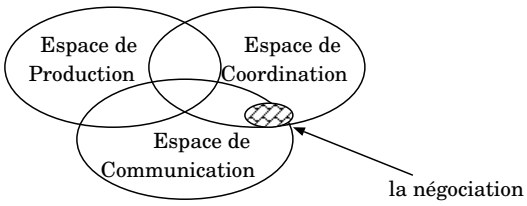
3.1 Critères de modélisation

L'objectif de ce travail est de spécifier d'une part un modèle de négociation qui soit indépendant d'un domaine d'application particulier. En d'autres termes, on vise la modélisation d'un service où on sépare la manière avec laquelle on négocie de ce qu'on négocie. Il s'agit réellement de définir des mécanismes génériques de négociation, utilisables tant pour négocier une qualité de service, l'allocation d'une ressource, un schéma de coopération à respecter pour le partage d'un certain objet, la sélection d'une ressource dans un gestionnaire de ressources que pour marchander le prix d'un produit en commerce électronique, par exemple. Notre modèle doit par conséquence découpler le mécanisme de négociation des concepts pûrement liés au domaine d'application, tout en offrant une ouverture à tout problème réel de négociation. D'autre part, il doit être souple et adaptable à toute sorte de langage de la négociation, assurant ainsi une forte évolutivité.

3.2 Position du modèle

Contrairement au modèle de [Huh99] (cf. section 2.4), la négociation qu'on veut formaliser n'est pas pécifique à des situations de compétition. En **travail coopératif**, la négociation s'avère également indispensable pour aboutir à une **décision commune**. C'est une **négociation douce** où les négociateurs s'échangent des décisions temporaires en essayant d'aboutir à une solution qui les satisfasse tous.

Par rapport au modèle fonctionnel d'Ellis, on situe le service de négociation dans l'intersection des espaces de **coordination** et de **communication** comme ci–dessous :



3.3 Le modèle générique de négociation

Pour pouvoir formaliser notre modèle générique de négociation, il est nécessaire de poser des définitions de base facilitant la compréhension du modèle et sa comparaison à d'autres modèles de négociation. Dans ce qui suit, on définit entre autres les notions de *base de décision de négociation*, de *langage des actes de négociation*, de *protocole de négociation* et de *tactiques de négociation.*

Définition 1 (Contexte d'une négociation) *On désigne par contexte d'une négociation, le 6–uplet* $C = \langle \mathcal{N}, \mathcal{B}, \mathcal{A}, \mathcal{P}, \mathcal{T}, \mathcal{E} \rangle$, *où :*

- $\mathcal{N}$ *: est l'ensemble des négociateurs,*
- $\mathcal{B}$ *: est la base de décision de la négociation,*
- $\mathcal{A}$ *: est le langage des actes de négociation,*
- $\mathcal{P}$ *: est le protocole de la négociation*
- $\mathcal{T}$ *: est l'ensemble des tactiques comportementales des négociateurs*
- $\mathcal{E}$ *: est l'effet de la négociation*

Définition 2 (L'ensemble des négociateurs) *On note* $\mathcal{N}$ *l'ensemble des acteurs, humains ou artificiels, participant à une négociation, tel que* $|\mathcal{N}| \geq 2$ *et la négociation n'est biaisée*[1] *pour l'intérêt d'aucun* $n \in \mathcal{N}$ *(coopération).*

Définition 3 (Base de décision d'une négociation) *On désigne par base de décision* $\mathcal{B}$ *d'une négociation, l'ensemble des couples* $\langle x_i, V_i \rangle$, x_i *étant la ième variable à négocier et* V_i *son domaine de valeurs possibles. (ex:* $\mathcal{B} = \{\langle x_1 =$ *schéma de coopération pour partager l'acticle a,* $V_1 = \{$*Client/Serveur, Ecriture_ Coopérative*$\}\rangle$, $\langle x_2 =$ *logiciel d'édition du plan,* $V_2 = \{$*Latex, Word, Adobe distiller, . . .*$\}\rangle\}$*).* $|\mathcal{B}| \geq 1$ *est une condition nécessaire pour procéder à une négociation.*

Définition 4 (Résultat d'une négociation) *Le résultat de la négociation* $\mathcal{R}$ *est l'ensemble des couples résultant de la négociation, plus formellement :* $\mathcal{R} = \{\langle x_1, v_1 \in V_1 \rangle, \ldots, \langle x_{|\mathcal{B}|}, v_{|\mathcal{B}|} \in V_{|\mathcal{B}|} \rangle\}$ v_i *étant la valeur finale de* x_i *décidée après négociation.*

Définition 5 (Messages d'une négociation) *Si l'on note* $\mathcal{G}$ *un langage des arguments d'une négociation, le langage des messages* $\mathcal{M}$ *d'une négociation sera défini comme suit :* $\forall n_1, n_2 \in \mathcal{N}$, $\forall \langle x_i, V_i \rangle \in \mathcal{B}$, $\forall v_i \in V_i$, $\forall g \in \mathcal{G}$, $\langle n_1, n_2, x_i, v_i, g \rangle \in \mathcal{M}$, *en d'autres termes, tout message envoyé par un négociateur* n_1 *à son interlocuteur* n_2 *contient la valeur* v_i *qu'il propose pour la variable* x_i *et éventuellement un argument supportant sa décision. On note* $var(m \in \mathcal{M})$ *(resp.* $val(m \in \mathcal{M})$*) la variable (resp. valeur) encapsulée dans un message de négociation.*

Définition 6 (Actes d'une négociation) *soit Prohibitives, Permissives, Assertives, Directives, Promissives, Expressives, et Déclaratives les septs ensembles disjoints des forces illocutoires. Et soit* $\mathcal{F}$ *leur union. On définit le langage des actes* $\mathcal{A}$ *d'une négociation comme suit :* $\forall f \in \mathcal{F}$, $\forall\, m \in \mathcal{M}$, $\langle f, m \rangle \in \mathcal{A}$. *On note* $mes(a \in \mathcal{A})$ *le message encapsulé dans un acte de négociation.*

Définition 7 (Protocole d'une négociation) *Par protocole* $\mathcal{P}$ *d'une négociation on désigne l'ensemble des actes que peut effectuer un négociateur à chaque point de la négociation. Le rôle d'un protocole étant d'assurer la cohérence et la convergence du processus de négociation en gérant :*

1. *les coalitions parmi les négociateurs,*

1. *tous les négociateurs ont des chances égales d'exprimer leurs décisions lors de la négociation*

2. *la co–décision d'un élément* $\langle x_i, v_i \in V_i \rangle$,

3. *le passage de parole entre les négociateurs,*

4. *la cohérence des actes émis par le même négociateur,*

5. *la cohérence des actes d'un négociateur par rapport aux actes des autres négociateurs étant intervenus ultérieuerement lors de la même négociation.*

Définition 8 (Tactiques de négociation) *On désigne par tactiques* $\mathcal{T}$ *de négociation un ensemble de règles permettant aux négociateurs, selon leurs contraintes (temps, ressource, ...) de choisir entre les alternatives offertes par* $\mathcal{P}$ *à chaque point de la négociation.*

Soit $t_{i \in \{1..|\mathcal{T}|\}}$ *la ième tactique . Si l'on définit une fonction d'utilité* U_n *pour le négociateur* $n \in \mathcal{N}$, *la stratégie optimale* s^* *de* n *peut être définie comme étant la composition de* p *tactiques permettant de maximiser* U_n, *plus formellement* $s^* = arg(max_s U_n(s))$ *avec* $s = t_{i_1} \circ t_{i_2} \circ .. \circ t_{i_p}$.

Définition 9 (Effet d'une négociation) *On désigne par l'effet* $\mathcal{E}$ *d'une négociation l'ensemble des opérations à effectuer conformément au résultat* $\mathcal{R}$ *de la négociation. Pour donner un exemple, on pose* $\mathcal{R} = \{\langle x_1, v_1 \in V_1 \rangle, \ldots, \langle x_{|\mathcal{B}|}, v_{|\mathcal{B}|} \in V_{|\mathcal{B}|} \rangle\}$ *le résultat final de la négociation. Selon chaque décision* $\langle x_i, v_i \in V_i \rangle$ *des opérations* (p_{i_j}) *doivent être exécutées, ainsi l'effet total de la négociation s'écrira :* $\mathcal{E} = \{(p_{1_1} \circ p_{1_2} \circ .. \circ p_{1_i}), .., (p_{|\mathcal{B}|_1} \circ p_{|\mathcal{B}|_2} \circ .. \circ p_{|\mathcal{B}|_j})\}$.

3.4 Instanciation du modèle de négociation

3.4.1 Approche transactionnelle

Sachant que le prototype DisCOO aborde le problème de la coopération selon une approche transactionnelle avancée *(Transactions Coopératives Distribuées)*, (cf. chapitre 1), et que le processus de négociation se prête bien à un comportement transactionnel, on va instancier le modèle théorique de négociation qu'on a proposé dans une approche transactionnelle.

Réellement, chaque acteur de DisCOO voulant participer à une négociation, verra la *DisCOO–transaction* qui l'encapsule s'engager dans ce processus de négociation. On se restreint dans le service de négociation qu'on va développer, à la négociation bilatérale ($|\mathcal{N}| = 2$). Les deux négociateurs sont des agents humains qui sont d'accord préalablement sur la perception du problème de décision à résoudre,

Parmi les concepts introduits par DisCOO (à savoir *référentiel local d'une transaction, d'opérations de transfert entre transactions* et *histoire locale*[2] d'une transaction, etc.), le dernier nous interessera plus particulièrement dans cette réalisation. En effet, l'intégration d'un service de négociation au sein de DisCOO impliquera entre autres l'enrichissement de l'ensemble des opérations d'une DisCOO–transaction. Ces opérations serviront entre autres à définir le protocole de négociation dans une approche transactionnelle.

On a donc choisi ACTA pour formaliser les axiomes de définition du protocole $\mathcal{P}$ de négociation pour mieux expliquer ses particularités et afin de faciliter sa comparaison à d'autres protocoles de négociation. Le formalisme ACTA[3] est basé sur la logique du premier ordre. Les règles ACTA permettent de caractériser les effets que les transactions produisent sur les autres transactions ainsi que sur les objets qu'elles manipulent (cf. annexe A).

2. H_t, l'histoire de la transaction t est l'ensemble ordonné des actions qu'effectue cette transaction et des actions effectuées par d'autres transactions sur les objets de t. L'histoire globale H est définie comme l'union partiellement ordonnée des histoires locales des transactions.

3. le mot ACTA signifie *actions* en Latin

3.4.2 Pour que DisCOO supporte un service de négociation

Pour qu'on puisse intégrer un service de négociation à l'environnement DisCOO, il est nécessaire d'enrichir l'ensemble des opérations que peut effectuer une DisCOO–transaction (cf. [Mun99]) par de nouvelles opérations définissant le processus de négociation. A l'exécution d'une nouvelle opération, celle–ci sera journalisée comme toute autre opération dans les histoires locales des transactions participant à la négociation.

Ci–dessous, les deux notions d'acteur DisCOO et de DisCOO–transaction l'encapsulant seront confondues. Nous utilisons la notation $op_{t_i}[obj_{t_j}, \ldots]$ pour représenter l'invocation, par l'acteur t_i, de l'opération *op* sur l'objet *obj* de l'agent t_j (ex : $read_{t_1}[obj_{t_2}]$ et $write_{t_2}[obj_{t_1}, val]$ correpondent aux deux opérations transactionnelles de base, lecture d'un objet et écriture d'une valeur dans un objet). Ci–dessous, on décrit les opérations de base de la négociation :

- t_1 : la transaction initiatrice de la négociation et H_{t_1} son histoire,
- t_2 : la transaction invitée à la négociation et H_{t_2} son histoire,
- t : une des deux transactions t_1, t_2,
- *nob* : *(negotiation object)* objet permettant de garder une trace de l'historique de la négociation courante et servant ainsi de référentiel d'échanges des décisions entre les deux, négociateurs, on note H_{nob} l'ensemble totalement ordonné des décisions prises par les deux agents.
- *Negop* : *(Negotiation operations)* un ensemble de quatre nouvelles opérations définissant le service de négociation au sein de DisCOO ; **Solicit**, **Assert**, **Accept** et **Kill** :
  - $\mathbf{Solicit}_{t_1}[req \in \mathcal{M}, nob_{t_1}] \in H_{t_1}$ signifie que t_1 sollicite t_2 pour l'inviter à étudier sa requête *req* lors d'une négociation (ex: $req = \langle n_1 = t_1, n_2 = t_2, x =$ "avec quel schéma partager la ressource r?", $v =$ schéma de coppération sollicité$\rangle$). Le *nob* passé en paramètre désigne l'objet de négociation qui sera utilisé par t_1 et t_2 pour échanger leurs décisions. Si l'on se réfère au cadre théorique, l'opération *Solicit* définit la base de la négociation.
  - $\mathbf{Assert}_{t \in \{t_1,\ t_2\}}[nob_t, nat] \in H_t$ signifie que t décide d'une solution $\langle x_i, v_i \in V_i \rangle$ en formulant un acte de négociation $nat \in \mathcal{A}$ *(negotiation act)*. Cette décision est journalisée dans l'objet nob_t. Cette opération correspond à un *write* sur l'objet nob_t [4]. On fait remarquer que t_1 (la transaction initiatrice de la négociation) au même titre que t_2 (la transaction invitée à la négociation) peut agir sur une négociation.
  - $\mathbf{Accept}_{t \in \{t_1,\ t_2\}}[nob_t, nat] \in H_t$ signifie que t accepte la solution $\langle x_i, v_i \in V_i \rangle$ proposée par l'acte *nat* formulé auparavant par la transaction $t' \in \{t_1, t_2\} \setminus \{t\}$. Il est nécessaire que *nat* ne soit pas de force illocutoire prohibitive. On fait remarquer que t_1 au même titre que t_2 peut accepter un acte adverse et donc signer un contrat. (voir axiome 12.)
  - $\mathbf{Kill}_{t \in \{t_1,\ t_2\}}[req] \in H_t$ signifie que t abandonne la négociation. Aucun schéma de coopération n'a été retenu pour partager la ressource concernée par $var(req)$ entre t_1 et t_2. On fait remarquer que t_1 au même titre que t_2 peut abandonner une négociation en cours (InProgress).

On fait remarquer qu'à part l'initiation de la négoctiation, les deux transactions négociatrices ont le droit d'agir au même titre durant le processus de négociation, elles se comportent selon le paradigme de coopération d'**égal–à–égal** (peer–to–peer). Cela renforce le fait que la négociation ne doit pas être biaisée au profit d'un négociateur sans les autres (cf. 3.3).

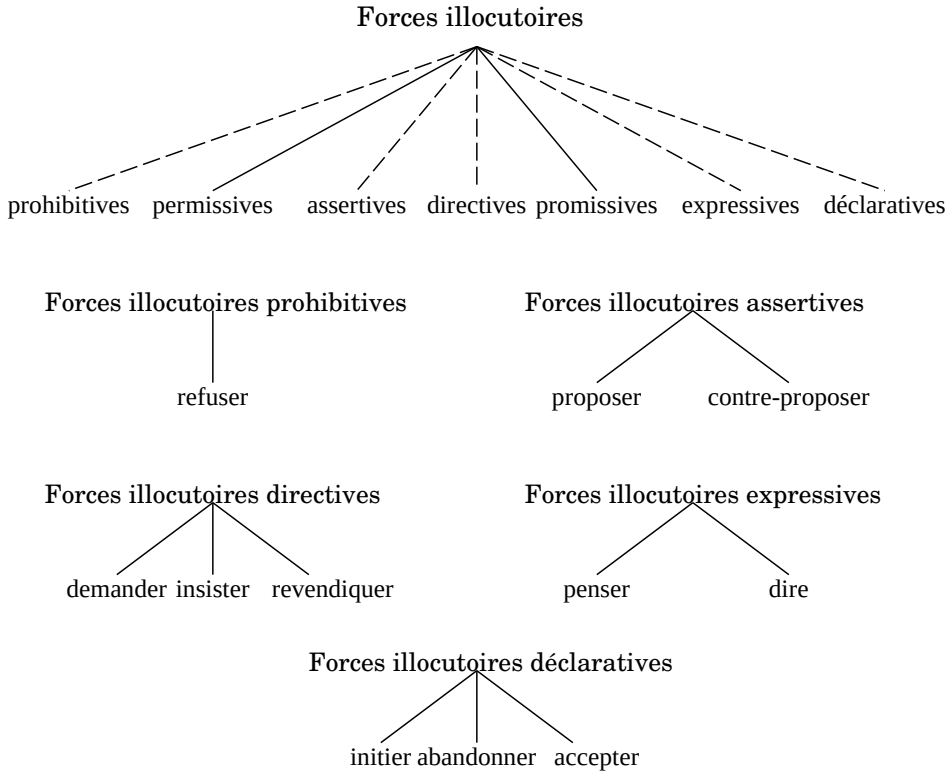
4. Au sein de DisCOO, cela signifie que son partenaire sera notifié de la mise à jour

3.4.3 La base de décision d'une négociation

On se restreint à une base de négociation minimale ($|\mathcal{B}| = 1$) tel que $\mathcal{B} = \{\langle x, V\rangle\}$ où $x =$ *"avec quel schéma partager les ressources R ?"* et $V =$ *{ Ecriture_ Coopérative, Client, Serveur, Rédacteur/Relecteur }*. Les notions de tactiques $\mathcal{T}$ et d'arguments de négociation $\mathcal{G}$, quant à elles n'ont pas été mises en pratique du fait que leur intégration à notre service de négociation demanderait plus de temps et de reflexions. On peut ainsi déduire la définition instanciée du langage de messages $\mathcal{M}$; $\forall n_1, n_2 \in \mathcal{N}, \forall v \in V, \langle n_1, n_2, x, v\rangle \in \mathcal{M}$.

3.4.4 Les actes de négociation

Parmi les septs catégories des forces illocutoires, on a estimé que les forces prohibitives, assertives, directives, expressives et déclaratives suffisent pour exprimer tous les actes d'un processus de négociation. La langue dans laquelle est exprimée la force illocutoire d'un acte importe peu, seule sa force illocutoire en tant que performatif est considérée. Ci–dessous on décrit l'ensemble de verbes exprimant des actes illocutoires de négociation qui ont été réellement utilisés, cet ensemble d'actes est laissé ouvert à toute autre extension :

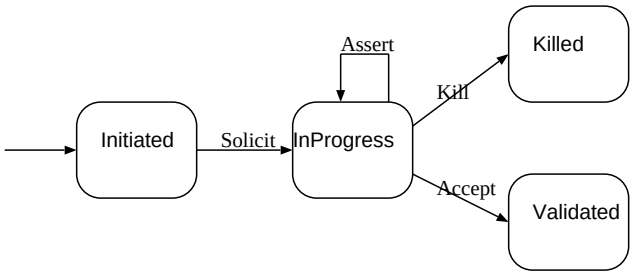


Taxonomie des actes de négociation

Avec la prise en compte de cinq catégories de forces illocutoires parmi sept, $\mathcal{F} =$ *Prohibitives* $\cup$ *Assertives* $\cup$ *Directives* $\cup$ *Expressives* $\cup$ *Déclaratives*, la définition de $\mathcal{A}$ s'en déduit ($\forall f \in \mathcal{F}, \forall m \in \mathcal{M}, \langle f, m\rangle \in \mathcal{A}$).

3.4.5 Le protocole de négociation

Le protocole de négociation qu'on présente est un enchaînement d'opérations de négociation dont le comportement global est décrit par le diagramme suivant :



Comme on l'avait fait remarquer en section 3.4.2, à part l'initiation de la négociation (opération **Solicit**), les deux transactions négociatrices t_1 et t_2 ont le droit d'agir au même titre durant le processus de négociation (opérations **Assert**, **Accept** et **Kill**), elles se comportent selon le paradigme de coopération d'égal–à–égal (peer–to–peer), ce qui montre que notre protocole de négociation est assez général, simple d'utilisation et ne pose pas de restriction forte sur les actions des négociateurs.

On va commencer par quelques définitions, ensuite on détaillera les axiomes, en ACTA (cf. annexe A), de formalisation du protocole $\mathcal{P}$ de négociation dans une approche transactionnelle :

Définition 10 (Précédence causale) *soient e_1 et e_2 deux événements, on dit que e_1 précéde causalement e_2 et on note $e_1 \rightarrow e_2$ si :*

- *e_1 et e_2 sont deux actions effectuées par la transaction t et qui paraîssent dans cet ordre dans H_t*
- *ou si $\exists$ un message m, tel que $e_1 =$ émission(m) et $e_2 =$ réception(m)*
- *ou si $\exists$ un événement e' tel que $e_1 \rightarrow e' \ \wedge \ e' \rightarrow e_2$*

NB : $\rightarrow$ est une relation d'ordre partiel dans H.

Définition 11 (axiomes fondamentaux des transactions) *Soit t une transaction et H_t son histoire locale. Soient α, β, γ et δ des événements significatifs parmi SE[5]. Si l'on note IE[6] l'ensemble des événements de commencement de transaction et TE[7] l'ensemble des événements de terminaison de transaction, SE se définit comme l'union de IE et TE ($SE = IE \ \cup \ TE$). $p_t[ob_{t'}]$ représente l'événement correspondant à l'invocation effectuée par la transaction t de l'opération p sur l'objet ob de la transaction t'.*

NB : Dans ce qui suit le symbole $\setminus$ désignera l'opération de différence ensembliste.

- *(I)* $\forall \alpha \in IE_t \ (\alpha \in H_t) \ \Rightarrow \ \nexists \beta \in IE_t \ (\alpha \rightarrow \beta)$,
  i.e. une transaction ne peut être initiée que par un événement initial unique,

- *(II)* $\forall \delta \in TE_t \ \exists \alpha \in IE_t \ (\delta \in H_t) \ \Rightarrow \ (\alpha \rightarrow \delta)$,
  i.e. une transaction ne peut pas être terminée si elle n'a pas été préalablement initiée,

5. *SE: Significant event*
6. *IE: Initial Events*
7. *TE: Terminal Events*

- *(III)* $\forall \gamma \in TE_t\ (\gamma \in H_t) \Rightarrow \nexists \delta \in TE_t\ (\gamma \rightarrow \delta)$,
  i.e. une transaction ne peut être terminée que par un événement terminal unique,

- *(IV)* $\forall ob_{t'}\ \forall p\ (p_t[ob_{t'}] \in H) \Rightarrow ((\exists \alpha \in IE_t\ (\alpha \rightarrow p_t[ob_{t'}])) \wedge (\exists \gamma \in TE_t\ (p_t[ob_{t'}] \rightarrow \gamma)))$,
  i.e. on ne peut invoquer des opérations que sur des objets d'une transaction en cours d'exécution (Inprogress)

Définition 12 (axiomes du protocole de négociation) *Ci–dessous, on présente les axiomes ACTA formalisant le protocole $\mathcal{P}$ de négociation dans une approche transactionnelle :*

1. $\forall t \in \{t_1, t_2\}$, *t satisfait les axiomes des DisCOO–transactions (cf. [Mun99]),*
2. $\forall i \in \{1, 2\}$, t_i, *augmentée de l'ensemble d'opérations* $Negop = \{$**Solicit**, **Assert**, **Accept**, **Kill**$\}$, *satisfait les axiomes des DisCOO–transactions, i.e. l'enrichissement des opérations des DisCOO–Transactions ne change pas l'exécution cohérente des DisCOO–transactions. Très exactement parce que ces nouvelles opérations n'agissent que sur des instances d'objets particuliers propres à la négociation (nob) sans "toucher" aux objets correspondant aux ressources partagées dont les DisCOO–Transactions gérent le contrôle de la cohérence des échanges.*
3. $\forall i \in \{1, 2\}, \forall p \in \{$**Solicit**, **Accept**, **Kill**$\}, p \in H_{t_i} \Rightarrow p \in H_{t_{i' \in \{1,\ 2\}\setminus\{i\}}}$, *i.e. toutes les opérations significatives à la négociation (commencement et terminaisons) sont journalisées simultanément dans les histoires des deux transactions négociatrices.*
4. *si l'on note* $nego_i$ *une parmi les négociations auxquelles a participé la transaction* t_i *et on schématise son comportement d'un point de vue transactionnel avec un seul événement initial* $IE_{nego_i} = \{$**Solicit**$\}$, *deux événements terminaux* $TE_{nego_i} = \{$**Accept**, **Kill**$\}$ *et comme seule opération possible* **Assert** *alors* $nego_i$ *respectera les axiomes fondamentaux des transactions (I à IV),*
5. $\forall i \in \{1, 2\}, \forall nego_i$, $t_i\ \mathcal{CD}\ nego_i \wedge t_i\ \mathcal{CD}\ nego_{i' \in \{1,\ 2\}\setminus\{i\}}$ *i.e.* t_i *ne peut valider que si toutes les négociations auxquelles elle participe ont terminé par un événement* **Accept** *ou* **Kill**, *(cf. annexe A.5.1)*
6. $\forall i \in \{1, 2\}, \forall nego_i$, $nego_i\ \mathcal{WD}\ t_i \wedge nego_i\ \mathcal{WD}\ t_{i' \in \{1,\ 2\}\setminus\{i\}}$, *i.e. si* t_i *est annulée et* $nego_i$ *n'est pas encore validée, alors* $nego_i$ *est annulée (production de l'événement terminal* **Kill***), (cf. annexe A.5.2)*
7. *si l'on note* $compNego_i$ *une transaction de compensation (annulant l'effet de la négociation* $nego_i$ *supposée validée chez ses deux parties prenantes) (cf. section 3.3) alors on a :*
   $\forall i \in \{1, 2\}$, $compNego_i\ \mathcal{BAD}\ t_i \wedge compNego_i\ \mathcal{BAD}\ t_{i' \in \{1,\ 2\}\setminus\{i\}}$, *i.e. si* t_i *annule son exécution, toutes les transactions de compensation endormies* $compNego_i$ *chez la transaction* t_i *commenceront à s'exécuter, (cf. annexe A.5.3),*
8. $\forall i \in \{1, 2\}$, $\mathbf{Solicit}_{t_i}[req \in \mathcal{M}, nob_{t_i}] \in H_{t_i} \Rightarrow write_{t_i}[nob_{t_i}, nat = \langle initier, req\rangle \in \mathcal{A}] \in H_{t_i}$, *i.e. toute sollicitation de négociation correpond à un ajout dans l'histoire* $H_{nob_{t_i}}$ *d'un acte déclaratif d'initialisation de la négociation, en posant* ($write_{t_i}[nob_{t_i}, nat] \in H_{t_i} \Rightarrow nat \in H_{nob_{t_i}}$),
9. $\forall i \in \{1, 2\}$, $\mathbf{Assert}_{t_i}[nob_{t_i}, nat \in \mathcal{A}] \in H_{t_i} \Rightarrow write_{t_i}[nob_{t_i}, nat] \in H_{t_i}$, *i.e. tout acte formulé par la transaction* t_i *sera ajouté à l'histoire* H_{nobt_i},
10. $\forall i \in \{1, 2\}$, $\mathbf{Accept}_{t_i}[nob_{t_i}, nat \in \mathcal{A}] \in H_{t_i} \Rightarrow$
    $write_{t_i}[nob_{t_i}, nat = \langle accepter, mes(nat)\rangle \in \mathcal{A}] \in H_{t_i}$, *i.e. toute acceptation d'un acte de négociation correpond à un ajout dans l'histoire* $H_{nob_{t_i}}$ *d'un acte déclaratif d'acceptation,*
11. $\forall i \in \{1, 2\}$, $\mathbf{Kill}_{t_i}[req \in \mathcal{M}] \in H_{t_i} \Rightarrow write_{t_i}[nob_{t_i}, nat = \langle abandonner, req\rangle \in \mathcal{A}]$
    $\in H_{t_i}$, *i.e. toute élimination d'une requête de négociation correpond à un ajout dans l'histoire* $H_{nob_{t_i}}$

d'un acte déclaratif d'abandon.

3.4.6 L'effet de négociation

Les tables des deux négociateurs t_1 et t_2 ne sont affectées réellement que lorsque le processus de négociation réussit (acceptation d'un acte de négociation).

12. $\forall i \in \{1, 2\}$, $\mathbf{Accept}_{t_i}[nob_{t_i}, nat \in \mathcal{A}] \in H_{t_i} \Rightarrow$
$(\mathcal{R} = \{\langle var(mes(nat)), val(mes(nat))\rangle\} \wedge Contract[t_i, t_{i' \in \{1, 2\}\setminus\{i\}}, \mathcal{R}] \in H_{t_i})$
i.e. l'effet d'une acceptation d'un acte de négociation correpond à un ajout dans l'histoire H_{t_i} d'un événement $Contract[t_i, t_{i' \in \{1, 2\}\setminus\{i\}}, \mathcal{R}]$, (signature de contrat).
Sachant que la table de coopération de l'acteur t_i est définie comme étant une vue sur H_{t_i} regroupant les contrats signés par t_i, comme suivant :
$table_de_coopération_{t_i} = Projection(H_{ct}, Contract[t_i, t_j, \mathcal{R}'] \in H_{t_i})$, alors les $table_de_coopération_{t_i}$ des deux négociateurs t_1 et t_2 contiendront le résultat de la négociation réussie (persistence du résultat $\mathcal{R}$ après négociation) (cf. annexe A.4). La définition formelle de l'effet de la négociation est : si l'on note p_{i_j} l'opération d'ajout de l'événement $Contract[t_i, t_j, \mathcal{R}]$ dans H_{t_i} alors l'effet de négociation sera défini comme suit : $\mathcal{E} = \{(p_{1_1} \circ p_{1_2})\}$.

Chapitre 4

Réalisation du service de négociation au sein de DisCOO

Après avoir posé les piliers essentiels du modèle de négociation, notre objectif est de réaliser un service de négociation des schémas de coopération pour le partage de ressources entre acteurs de l'environnement de travail coopératif DisCOO.
Toute notion, mentionnée dans ce chapitre, propre à un outil technique sera brièvement expliquée, le but étant d'avoir un discours clair et concis.

4.1 Critères de réalisation

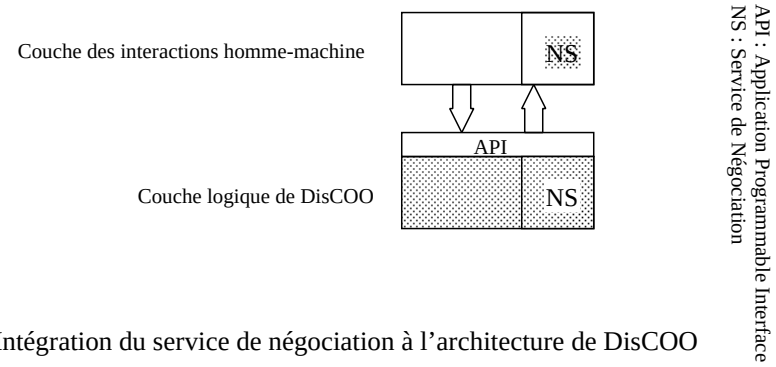
Le service de négociation doit être :

1. extensible,
2. fiable,
3. portable,
4. centré utilisateur.

En effet, le service de négociation à réaliser ne doit pas contenir d'information codée en dur empêchant les extensions futures (l'ensemble des actes de négociation par exemple). De plus, il doit être portable, i.e. ne doit dépendre ni d'une plate–forme matérielle ni d'une plate–forme système précise. Il doit également être basé sur une architecture logicielle assurant sa fiabilité et son extensibilité. Tout cela, sans oublier que la conception d'un service d'aide à la négociation au sein d'un environnement de travail coopératif CSCW doit être centrée utilisateur. On dit qu'un système est centré utilisateur si sa conception se base essentiellement sur un modèle utilisateur qui sera pris en considération pendant toutes les étapes du développement (cf. la théorie de l'action de Norman [Nor86]). Entre autres, le système doit être facile à utiliser si l'on considére un utilisateur ayant lu l'article [Ben98]. La langue du négociateur doit être également prise en compte, i.e. le fait que les négociateurs négocient en deux langues differentes doit leur être transparent ; un négociateur Français (i.e. choisissant le Francais comme langue de son interface) recevra tous les actes de négociation en Francais et un négociateur Espagnol recevra tous les actes de négociation que lui émet le négociateur Francais en Espagnol. Plus conrètement, les forces illocutoires, ne devant pas dépendre d'une certaine langue (cf. section 3.4.4), apparaîtront en fonction de la langue de l'interface.

4.2 Intégration du service de négociation à l'architecture de DisCOO

L'architecture logicielle de DisCOO se compose de deux couches, une *couche logique* gérant tous les services offerts par DisCOO [Mun99], et une *couche des interactions homme–machine* gérant la communication entre la couche logique et un utilisateur. Il est indispensable de signaler que toute l'architecture DisCOO a été développée en langage Java au dessus de la plate–forme de communication CORBA[1] [Gei97]. En effet, les couches logiques de DisCOO chez deux utilisateurs différents communiquent via un bus logiciel CORBA. Le schéma suivant situe les modifications apportées à l'architecture de DisCOO :



Intégration du service de négociation à l'architecture de DisCOO

4.2.1 Au niveau logique de DisCOO

Le développement du service de négociation se compose *essentiellement* de plusieurs classes d'objets et d'interfaces[2] structurées en une hierarchie de paquetages (qui ne vont pas être présentés dans ce qui suit pour ne pas perdre en généralité). Ci–dessous, sont décrites brièvement les interfaces composant le service de négociation réalisé en désignant par *instance concrète* d'une interface une instance d'une classe concrétisant (implantant) cette interface.

- NOB (Negotiation Object) qui est une interface dont toute concrétisation permet de gérer les objets *nob* journalisant les actes d'une négociation (force illocutoire + décision) (cf. section 3.3), et servant ainsi de référentiel d'échanges des actes entre les deux négociateurs, i.e. pour chaque négociation, chacun des deux négociateurs posséde une instance concrète de cette interface. Ces deux instances qui sont pour le moment totalement synchronisées (les modifications sur l'une des instances du NOB chez le négociateur n_1, s'appliquent sur l'autre instance chez n_2, ceci permet d'assurer un ordre total entre les actes formulés par n_1 et n_2). Ceci n'empêcherait pas ultérieurement de voir toute instance concrète de NOB comme étant une ressource DisCOO partagée selon un paradigme parmi { **Ecriture_coopérative**, **Rédacteur/Relecteur**, **Client/Serveur** ou **TurnTaking** (à tour de rôle (cf. section 4.3.2))},
- negotiation qui est une interface dont chaque concrétisation encapsule les propriétés et les méthodes spécifiques à une négociation, i.e. pour chaque négociation, les deux négociateurs possédent deux instances concrètes de cette interface gérant les concepts partagés (ex: NOB) et non partagés (ex: tactiques (non encore développées) (cf. section 4.3.2) entre les deux négociateurs). Cette interface représente, en quelque sorte, le contexte $\mathcal{C}$ de la négociation (cf. section 3.3),

1. CORBA : Common Object Request Brocker Architecture
2. une interface en java est l'ultime classe abstraite, i.e. c'est une classe abstraite (aucune instance possible) qui ne spécifie que les profils des méthodes sans leur corps en plus de propriétés constantes

- negotiationMessage qui est une interface dont toute concrétisation définit un langage de messages de négociation $\mathcal{M}$ particulier à la nature de la négociation (cf. section 3.3),
- illocutionaryForce qui est une interface dont toute concrétisation définit une partie de l'ensemble des forces illocutoires $\mathcal{F}$ (cf. section 3.3). Le seul profil qui y est déclaré est celui de la méthode abstraite public java.lang.String identifier(); ; pour le moment une force illocutoire est définie par un identificateur, sans pour autant interdire l'ajout (cf. 4.1) de nouvelles propriétés (cf. section 4.3.2),
- assertive, declarative, directive et prohibitive qui sont des interfaces étendant l'interface illocutionaryForce, et représentant respectivement les ensembles des forces illocutoires *assertives, déclaratives, directives et prohibitives*,
- negotiationAct qui est une interface dont toute concrétisation définit un langage des actes de négociations $\mathcal{A}$ (cf. section 3.3), elle déclare les profils de deux méthodes
  public DisCOO.Negotiations.NegotiationActs.illocutionaryForce iForce(); et
  public DisCOO.Negotiations.NegotiationActs.negotiationMessage message();, renvoyant respectivement la force illocutoire et le message d'un acte de négociation.

Sans rentrer dans les détails techniques de la programmation à l'aide de l'outil CORBA, on signale que toutes les interfaces ci–dessus sont des extensions de l'interface org.omg.CORBA.CORBject[3]. Ce qui implique que les instances concrètes réalisées de ces interfaces sont des objets manipulables par le protocole de communication CORBA (cf. [Gei97]).

4.2.2 Au niveau de l'interface graphique de DisCOO

L'interface utilisateur initiale de DisCOO[4] est un outil de test et de démonstration du prototype DisCOO. Elle a été structurée et développée de telle manière qu'elle puisse être facilement étendue. Tout nouveau concept ou fonctionnalité, dont l'ajout au sein de DisCOO s'avère intéressant, doit être intégré facilement à l'interface sans remettre en cause les choix déjà faits. En effet, l'interface graphique de DisCOO a été conçue à l'aide des *Design Patterns* (appelés aussi patrons ou motifs de conception).

Les Design Patterns sont des éléments architecturaux servant de briques de base pour la conception logicielle d'un système qui met en avant les critères de réutilisabilité et d'extensibilité pour les logiciels de longue durée de vie. Ces schémas de conception permettent de capturer l'intention sous–jacente à une architecture en identifiant les objets, leurs collaborations, et la répartition des responsabilités; réduisant ainsi la complexité apparente du système. Leur origine remonte à un travail fait par un architecte nommé Christopher Alexander durant les années 70 qui a proposé l'idée d'utiliser un langage de patterns pour l'architecture des bâtiments et des grandes cités. En conception des logiciels, le mouvement des patterns ne commencera que très longtemps après ; en 1987, quand les patterns sont réapparus dans une conférence OOPSLA[5] (un forum majeur pour les chercheurs et les utilisateurs dans le domaine de la technologie à objets). Depuis, il y a eu beaucoup d'articles et de publications liés directement ou indirectement aux Design Patterns. En 1995, le livre [Gam94] a été publié et reste jusqu'à maintenant la référence dans le domaine de conception par Design Patterns.

Grâce à sa capacité à assurer la cohérence des données d'un même modèle indépendamment de leurs représentations visuelles (graphique, numérique, ...), c'est le patron de conception OBSERVER qui importe le plus pour un développement rigoureux des interfaces graphiques. Ce patron permet de définir une relation n–aire entre objets de telle sorte que si un objet change d'état, tous les objets qui en dépendent sont notifiés

3. org.omg.CORBA.CORBject définit les objets du bus logiciel CORBA utilisé
4. conçue et développée par moi même durant le stage de deuxième année ENSIMAG Juin–Septembre 1998
5. OOPSLA Object Oriented Programming Systems, Languages and Applications

et mis à jour automatiquement. Le MVC[6], qui est une structure–objet apparue pour la première fois en Smalltalk80 et qui sert à bien distinguer entre les données, leurs représentations et les interactions les concernant, est parmi les utilisations les plus connues de l'OBSERVER. Le MVC est un schéma de conception qui assure l'extensibilité, i.e. pour modifier la représentation des données visualisées, il suffit de modifier la vue sans toucher au modèle et pour enrichir le dialogue homme–machine concernant une certaine vue, il suffit de modifier le contrôleur, également sans toucher au modèle. Sans parler de la gestion du rafraîchissement automatique des données visualisées et leurs cohérence à un instant donné qui ne devient plus un problème. En effet, le Modèle/Vue/Contrôleur est basé sur le principe *d'inscription/notifications* qui permet de créer un service (le modèle), un inscrit (la vue) et de mettre à jour les données de l'inscrit en fonction des données du service en le notifiant à chaque nouvelle modification.

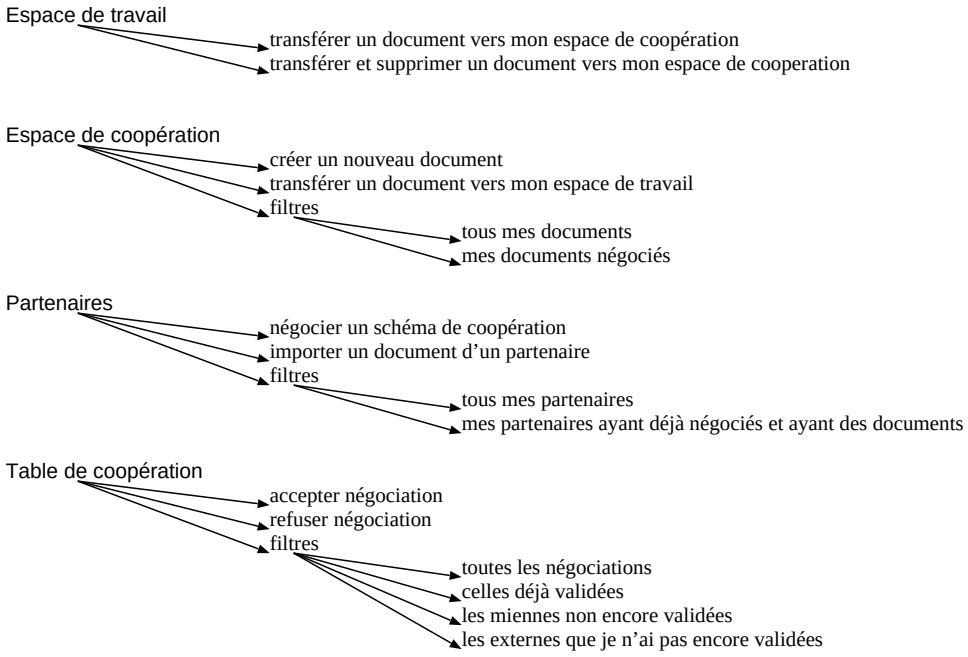
Pour intégrer le service de négociation au niveau de l'interface graphique de DisCOO, les ajouts réalisés concernent trois nouveaux MVC. Un d'entre eux, gère une vue sur l'*Histoire de négociation* qui représente réellement l'hitoire H_{nob} de l'objet *nob* de la négociation et les deux autres gèrent deux nouvelles fonctionnalités *Agir* apparaîssant dans la vue de l'*Histoire de négociation* et *Histoire des actes d'une négociation* apparaîssant dans la vue de la *Table de coopération.* (cf. section 4.3).

Ceci montre que les critères, d'extensibilité et de réutilisabilité, visés par le développement en Design Patterns de l'interface graphique initiale de DisCOO ont été respectés vu la facilité de l'extension de son architecture.

4.3 Démonstration du service de négociation

4.3.1 L'interface initiale de DisCOO

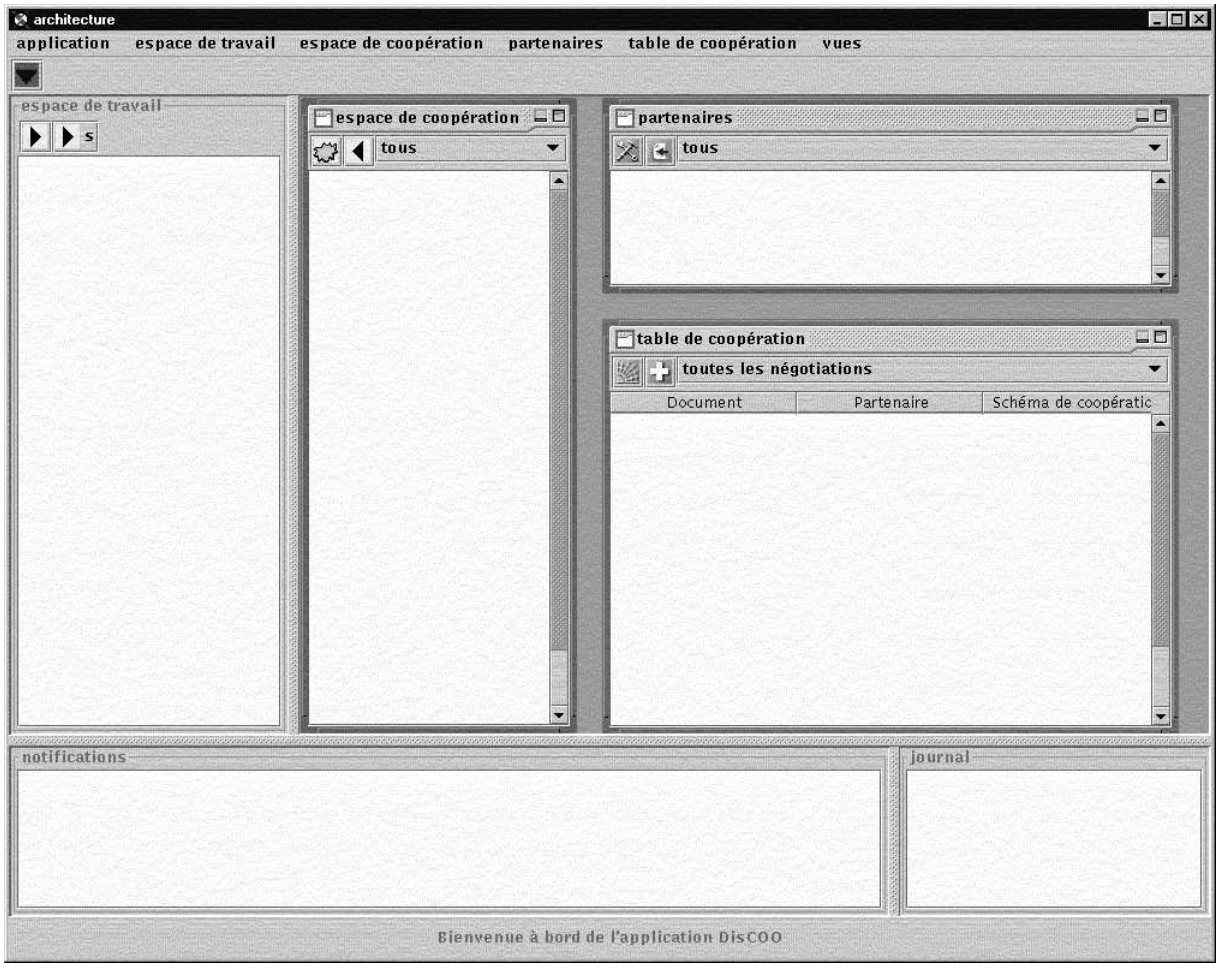
Ci-dessous, l'hiérarachie des fonctionnalités offertes à l'utilisateur de la version initiale de l'interface de DisCOO :



Taxonomie initiale des fonctionnalités principales de l'interface de DisCOO

6. MVC : Model/View/Controller

Ci-dessous, l'interface (MDI) [7] initiale de DisCOO, avec des vues sur **l'espace de travail**, **l'espace de coopération**, **la table de coopération**, **les espaces de coopération des partenaires**, **les notifications des événements externes** et **le journal de la transaction** (son histoire) :

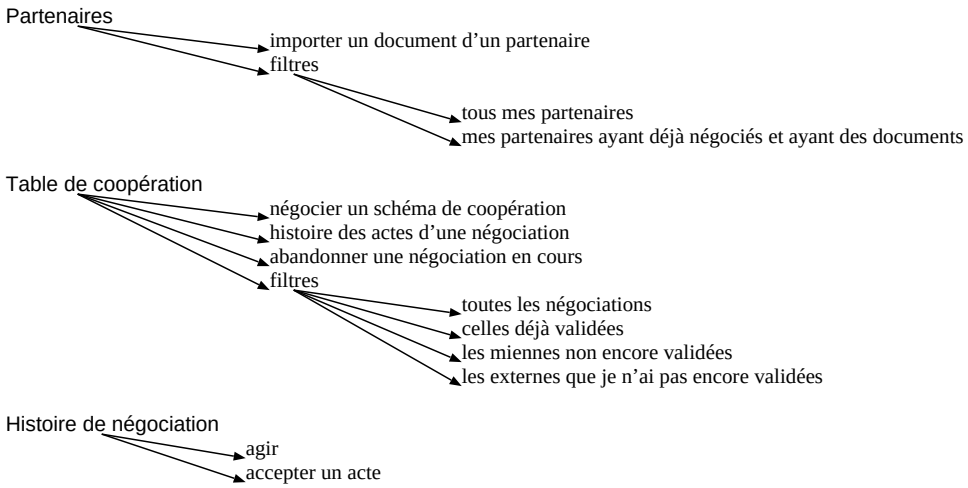


4.3.2 L'interface actuelle de DisCOO

On va essayer de présenter la nouvelle interface de DisCOO en mettant en valeur le service de négociation réalisé. Un scénario de négociation se déroule entre deux acteurs (un **thermicien** et un **architecte**) au sein de DisCOO.

7. MDI : Multiple Display Interface

Ci-dessous, la hiérarchie des nouvelles fonctionnalités offertes à l'utilisateur de l'interface de DisCOO :

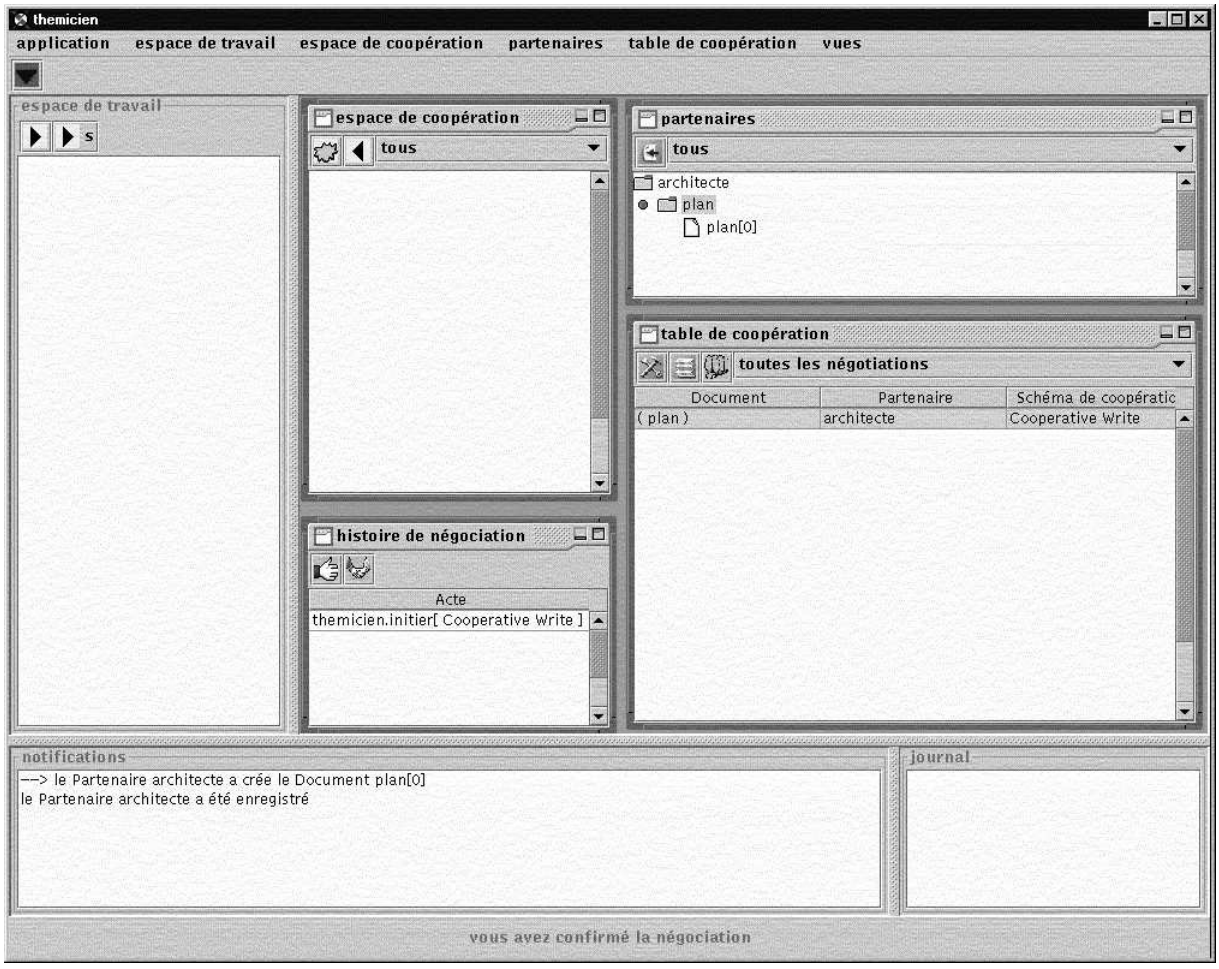


Changements apportés à la taxonomie initiale de fonctionnalités

L'exemple suivant permet de décrire simplement l'utilisation du nouveau service de négociation au sein de DisCOO :

(0) le **thermicien** se connecte en premier (voir **les notifications** sur l'interface du thermicien, "l'architecte est enregistré aprés lui"),

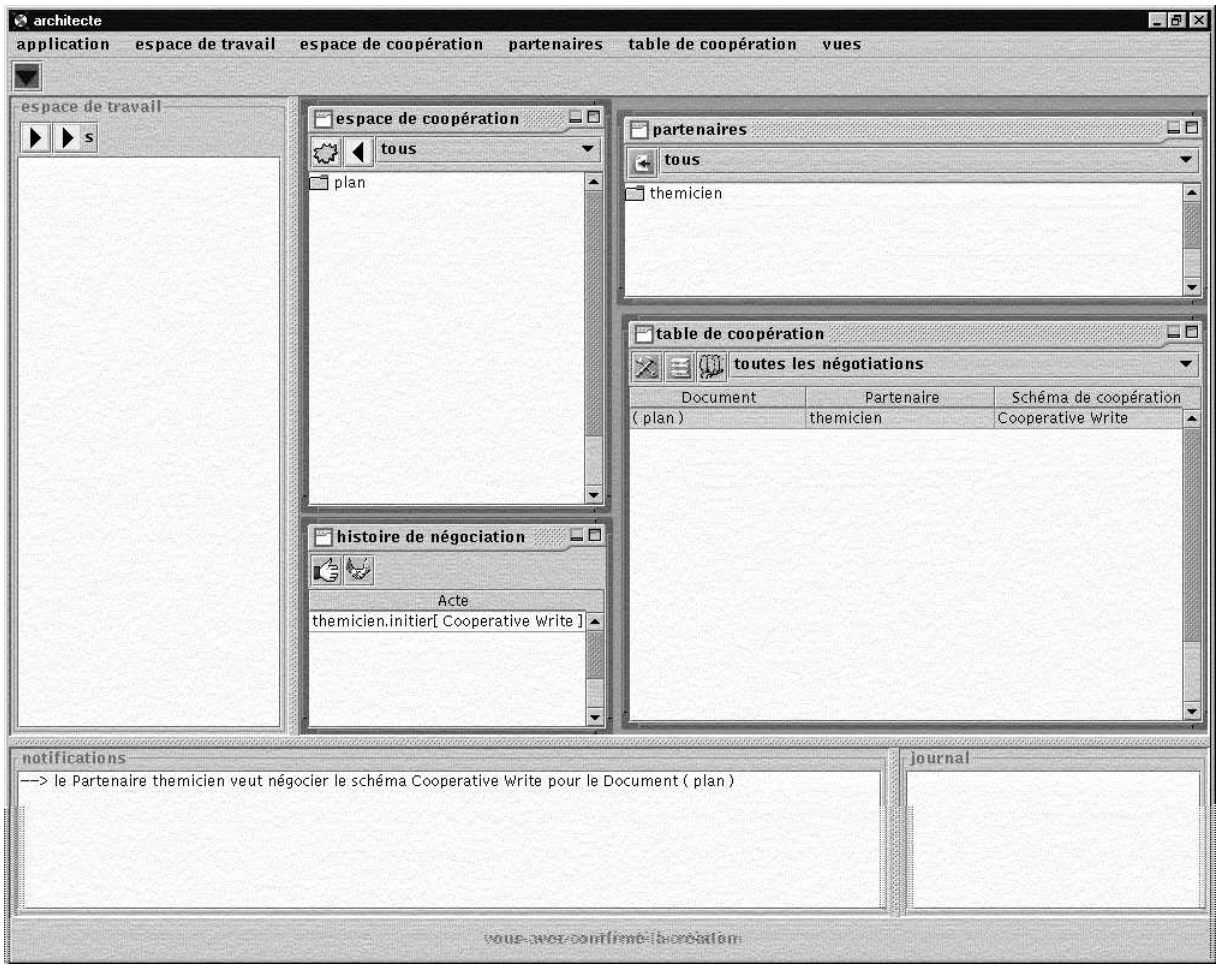
Voiçi ce que voit le **thermicien** sur son interface après qu'il ait initié la négociation :



(1) l'**architecte** se connecte et crée une ressource *plan* (voir **l'espace de coopération** sur l'interface de l'architecte),

(2) voulant partager la ressource *plan* avec l'**architecte**, le **thermicien** initie la négociation, en proposant le schéma de coopération **Ecriture_coopérative** (voir **table de coopération**, **histoire de négociation** et **notifications** sur l'interface architecte)

Voiçi ce que voit l'**architecte** sur son interface à l'initiation de la négociation par le **thermicien** :

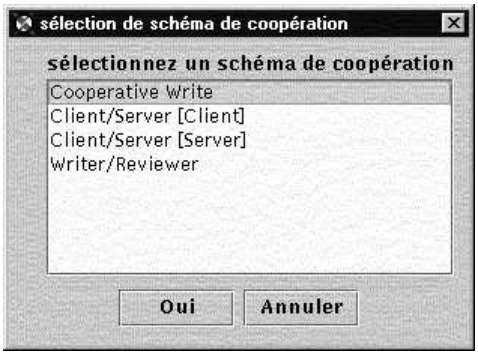


Ci–dessous la liste permettant à l'acteur DisCOO de sélectionner une force illocutoire pour agir lors d'un négociation. Vous remarquerez que les forces déclaratives = { initier, abandonner, accepter } ne figurent pas parmi cette liste, en fait, ces forces ne sont pas manipulées directement par l'utilisateur, elles sont encapsulées dans les opérations *négocier un schéma de coopération* de la vue sur la *Table de coopération*(**Solicit**), *agir* de la vue sur l'*Histoire de négociation* (**Assert**) et *abandonner une négociation en cours* de la vue sur la *Table de coopération* (**Kill**) (cf. section 3.4). Deux langues sont pour le moment prises en compte, aucune restriction n'est censée limiter les langues gérés par le service de négociation :

Version Française de la liste des forces illocutoires :



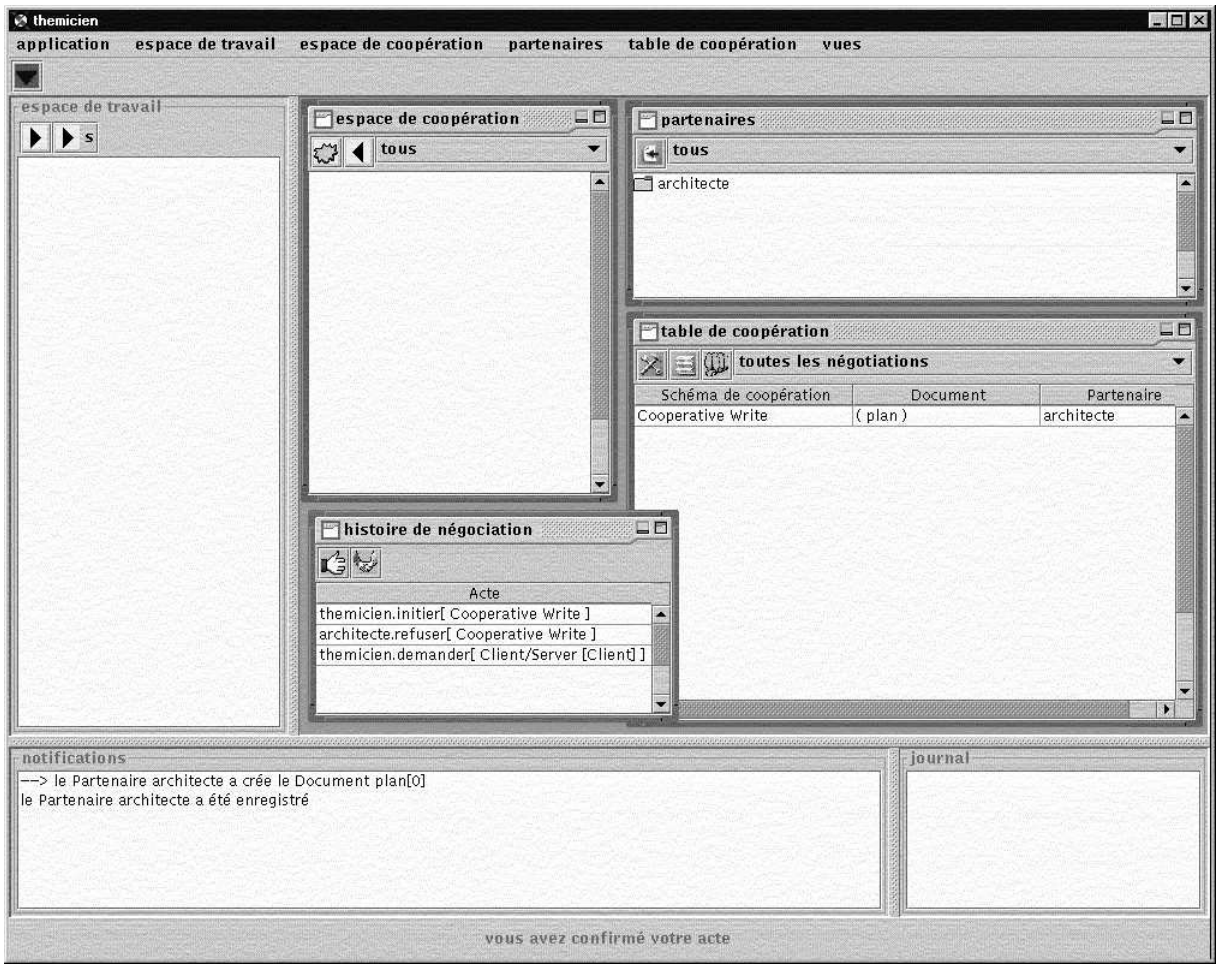
Ci–dessous la liste permettant à l'acteur DisCOO de sélectionner le schéma de coopération correspondant à son acte de négociation :



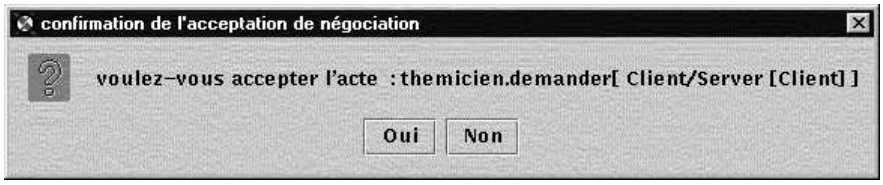
(2) L'**architecte refuse** la proposition du **thermicien**,

(3) le **thermicien**, demande cette fois le schéma de coopération **Client** pour partager la ressource *plan*

de l'**architecte**,

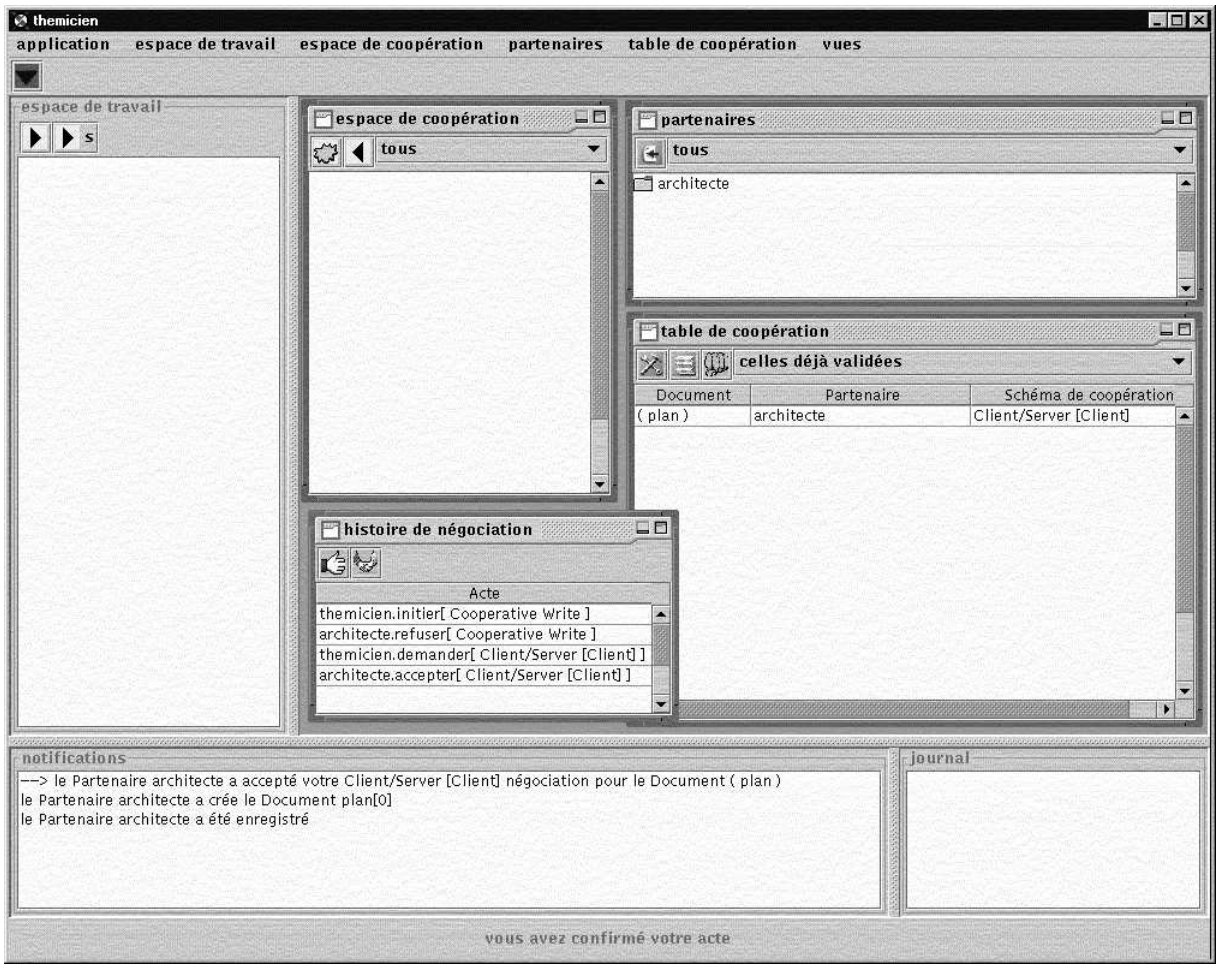


(4) l'**architecte**, accepte la demande du schéma de coopération **Client** de la part du **thermicien**,



Voiçi ce que voit le **thermicien** sur son interface après que l'**architecte** ait validé la négociation en

acceptant sa demande :



Conclusion et Perspectives

La négociation a été traitée pendant des décennies dans divers domaines ; la théorie des jeux, économie, psycho–sociologie et la gestion. [Spe82] la décrit comme étant la science des observations précises, des prétentions réalistes, de l'analyse cohérente des faits, des conclusions logiques, des comportements planifiés et de la prévention optimale contre tout changement de la situation. Elle représente un domaine de recherche intéressant si l'on considére ses multiples applications notamment dans le domaine des systèmes répartis (ex: télétravail, commerce électronique, travail coopératif via le réseau, etc.).

La négociation est un concept indipensable si l'on considère des situations conflictuelles de forte fréquence entre des acteurs qui évoluent dans un même environnement, l'objectif étant de coopérer pour obtenir un résultat commun. On peut imaginer l'exemple, donné par [Kra98], de plusieurs robots mobiles qui doivent coopérer pour rassembler des échantillons spécifiques sur la planète Mars avec tous les problèmes de **coordination** et de **communication** qui peuvent se produire.

Dans la littérature, on distingue entre deux grandes classes de négociation; la **négociation douce** caractérisée par une situation de **comportements coopératifs** où les négociateurs ont plusieurs solutions possibles qu'ils adaptent selon leurs contraintes et leurs objectifs afin d'atteindre un état de satisfaction commun et la **la négociation dure** rassemblant des négociateurs voulant maximiser leur part des ressources sans faire de concession, c'est le cas des négociateurs dont les objectifs sont en conflit direct et les ressources sont fixées et limitées (ex: temps, mémoire, terrain, part de marchés etc.)

Aucun modèle de négociation ne peut se prévaloir d'être adaptable à chaque problème de négociation. En fait, plusieurs paramètres sont à prendre en compte : le **langage de la négociation**, le **domaine de la négociation**, la **nature du conflit** à résoudre sans oublier les **média de communication**.

Nous nous sommes intérêssé durant ce projet de DEA à l'application des mécanismes émanant des recherches dans le domaine de la négociation en **ingénierie concourante**. Ce terme désigne le fait qu'un ensemble de personnes travaillent en groupe pour aboutir à un objectif commun (concevoir un projet, résoudre un problème de décision, coéditer un document, etc.). La classe de négociation qui nous intéresse est la **négociation douce** du fait que les environnements d'ingénierie concourante ou (CSCW) font apparaître le concept de **coopération** plus que celui de compétition. La plupart des modèles formels de négociation existants favorisent le concept de **compétition** et donc offrent des cadres théoriques non adaptés au but de l'étude, bien qu'ils exposent des notions intéressantes à la mise en œuvre dans un service d'aide à la négociation au sein d'un environnement d'ingénierie concourante (CSCW).

Notre démarche a visé, dans un premier temps, l'étude des résultats de trois disciplines de recherche : **la théorie des jeux**, **l'intelligence artificielle distribuée** et **la théorie des actes de langages** l'objectif étant d'aboutir à un modèle formel axiomatisant un processus de négociation qui soit **générique** et **extensible**. Une deuxième étape était de réaliser un service d'aide à la négociation opérationnel au sein de l'environnement de travail coopératif DisCOO (cf. [Mun99]) développé au sein de l'équipe d'accueil ECOO.

Le modèle de négociation formalisé et réalisé n'est évidemment pas parfait, il serait intéressant d'explorer les innombrables extensions possibles afin d'aboutir à un service d'aide à la négociation plus riche. Ci–dessous, on présente quelques perspectives de recherche de nouveaux concepts et mécanismes à intégrer au sein de notre service de négociation :

- avoir plusieurs protocoles de négociation et commencer par en négocier un, avant même de commencer la négociation du partage des ressources, ça sera une application du processus de négociation qu'on a proposé aux protocoles de négociation (intérêt de la généricité du modèle de négociation défini),
- pouvoir négocier la taxonomie des forces illocutoires de négociation avant de commencer la négociation du partage des ressources (application du même processus de négociation aux actes de négociation),
- pour l'instant, l'objet NOB[8] permet d'assurer un ordre total entre les actes formulés par deux parties prenantes de la négociation. Ceci n'empêcherait pas ultérieurement de voir toute instance NOB comme étant une ressource DisCOO partagée selon un paradigme parmi {**Ecriture_coopérative**, **Rédacteur/Relecteur**, **Client/Serveur** ou **TurnTaking** (à tour de rôle)}. L'ordre total ente les actes de négociations formulés par les deux négociateurs ne sera pas toujours respecté, donc, l'histoire du H_{nob} ne serait plus linéaire ...,
- semi–automatiser le processus de négociation, i.e. un agent artificiel réactif aidera l'acteur humain à négocier en fonction de tactiques comportementales *(voir point suivant)*,
- pour chaque négociation, offrir aux deux négociateurs des propriétés et méthodes privées, i.e. non partagées entre les deux négociateurs), (ex : tactiques de négociation, ...), les tactiques peuvent également être globales à toute les négociations d'un même négociateur,
  - chaque négociateur n_i devrait pouvoir définir le nombre maximum d'actes au bout desquels il commencerait à se poser des questions sur la convergence de la négociation. On note S_i : ce seuil. Un agent réactif l'informerait que ce seuil est atteint et lui demanderait de confirmer l'abandon de la négociation.
  - chaque acteur devrait pouvoir spécifier pour chacune de ses ressources x_i (resp. des ressources de ses partenaires) l'ensemble des schémas de coopération qu'il autorise (resp. propose) V_i, en quelque sorte, il définit les bases de ses futures négociations $\mathcal{B}_i = \{\langle x_i, V_i \rangle\}$, L'agent réactif pourrait procéder aux négociations des ressources externes et contrôler les négociations des ressources locales, bien évidemment l'agent réactif devrait être muni d'un service d'explicabilité (communication et argumentation) qui se chargerait d'informer l'acteur humain des traitements effectués par l'agent réactif.
  - des conditions de coordination des différentes négociations en cours pourraient être spécifiées (ex: sur les résultats des négociations en cours, sur leur ordre de terminaison, ...),
- pour chaque force illocutoire, en plus de l'identificateur, on pourrait attribuer une valeur numérique qui aidera l'agent réactif à sélectionner un acte dans la taxonomie des actes en fonction de l'histoire de la négociation,
- enrichir le langage des actes de négociation par de nouvelles forces illocutoires :
  - pour intégrer des *(mécanismes d'aide à l'argumentation)* { questionner (directive) / expliquer (assertive)},
  - pour permettre plus de contrôle sur la cohérence des actes de langages { réserver (déclarative) / confirmer (déclarative), reporter (déclarative) / continuer (déclarative)},

8. représenté par deux instances totalement synchronisées

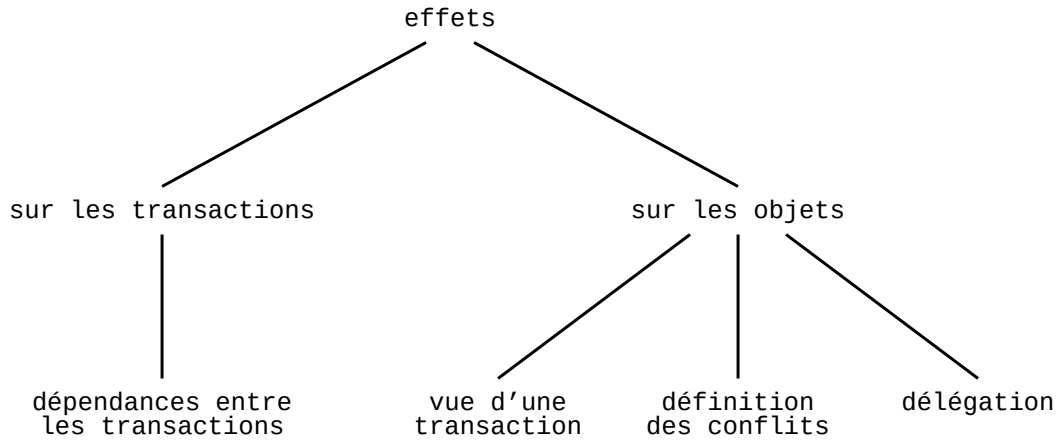
- pour pouvoir étendre le nombre de participants à la négociation { déleguer (déclarative), arbitrer (déclarative)}.

- ...

Annexe A

Le formalisme ACTA

La diversité des modèles de transactions étendus, leur relative complexité et, pour certains d'entre eux, leur manque de formalisation, rendent la caractérisation et la comparaison de ces modèles délicate. Devant cet état de fait, ACTA a été proposé comme un outil de spécification formelle permettant de caractériser le comportement et les propriétés des modèles de transactions [Chr94, Sch98]. Le formalisme ACTA est basé sur la logique du premier ordre. Les règles ACTA permettent de caractériser les effets que les transactions produisent sur les autres transactions ainsi que sur les objets qu'elles manipulent.



Dans un système, les transactions produisent des effets sur des objets en déclenchant des événements objets (ex: opérations lire, écrire) et des effets sur d'autres transactions en déclenchant des événements significatifs (ex: début, validation, abandon). Les effets sur les objets mettent en jeu les concepts de visibilité, de conflits et de délégation d'effets, tandis que les effets sur les transactions mettent en jeu des dépendances entre les transaction. La figure A illustre la classification de ces effets dans l'environnement ACTA.

A.1 Préliminaires

ACTA considère l'appel d'une opération sur un objet comme un événement objet. L'événement $p_t[ob]$ correspond à l'invocation de l'opération p par la transaction t sur l'objet ob. L'effet d'une opération sur un objet n'est pas rendu permanent au moment de l'exécution de l'opération. L'opération doit être explicitement validée ou annulée pour que l'effet devienne permanent.

Le formalisme ACTA repose sur la définition de prédicats portant sur l'occurence d'événements objets et d'événements significatifs dans l'historique d'exécution (noté H) constitué de tous les événements produits

par l'ensemble des transactions du système et indiquant l'ordre partiel dans lequel ces événements ont été produits[1]. H_{ct} représente l'histoire courante, c'est à dire l'histoire des événements à un moment donné.

Un modèle transactionnel est défini en ACTA à l'aide de prédicats et de règles ACTA caractérisant l'ensemble des histoires que peut produire ce modèle. Cela peut se représenter comme des invariants définis sur les histoires ou comme des pré et post conditions sur les événements objets et les événements significatifs. Les propriétés de correction apportées par les différents modèles sont alors exprimées en terme de propriétés sur les histoires.

A.2 Dépendances Inter-Transactions

A la base les dépendances sont des contraintes sur les histoires, produites par l'exécution concurrente de transactions interdépendantes. Par exemple une dépendance de terminaison entre deux transactions peut s'exprimer de la manière suivante: $t_j\ \mathcal{CD}\ t_i$ et signifie que si t_i et t_j terminent alors la terminaison de t_i précède celle de t_j, i.e.:

$$Commit_{t_j} \in H \Rightarrow (Commit_{t_i} \in H(Commit_{t_i} \rightarrow Commit_{t_j})$$

De même un dépendance d'annulation peut s'exprimer de la manière suivante: $(t_j\ \mathcal{AD}\ t_i)$ signifie que si t_j est annulée alors t_j doit également être annulée.

$$Abort_{t_i} \in H \Rightarrow Abort_{t_j} \in H$$

Les dépendances entre transactions peuvent être structurelles ou développées dynamiquement au cours de l'exécution:

Dépendance de structure Par exemple, dans les transactions emboîtées, la création d'une sous-transaction crée une dépendance de terminaison entre le père et le fils.

$$Spawn_{t_p}[t_c] \in H \Rightarrow (t_c\ \mathcal{WD}\ t_p) \wedge (t_p\ \mathcal{CD}\ t_c)$$

Dépendances de comportement Les dépendances peuvent aussi être produite à l'exécution par des interactions entre transactions sur des objets partagés. Par exemple:

$$(p_{t_i}[ob] \rightarrow q_{t_j}[ob] \Rightarrow (t_j\ \mathcal{D}\ t_i)$$

indique que si t_i invoque l'opération p sur ob et qu'ensuite t_j invoque l'opération q sur le même objet, alors t_j développe une dépendance de type $\mathcal{D}$ vers t_i.

A.3 Conflits entre Opérations

Une histoire $H^{(ob)}$ d'opérations sur un même objet ob est représentée comme la composition fonctionnelle des différentes opérations ordonnées dans le temps: $H^{(ob)} = p_1 \circ p_2 \circ \ldots \circ p_n$ avec $p_i \rightarrow p_{i+1}$. Chacune de ces opérations retourne une valeur et produit un état. Si s est l'état d'un objet, $return(s,p)$ renvoie le résultat produit par l'opération p sur cet objet. $state(s,p)$ retourne l'état de cet objet produit après l'exécution de l'opération p. L'état s d'un objet ob produit par une séquence d'opérations à partir d'un état initial s_0 est donc égal à $state(s_0,H^{(ob)})$.

Définition 13 (Le prédicat conflict) *Deux opérations p et q sont dites conflictuelles pour un état $H^{(ob)}$ (ce qui sera noté $conflict(H^{(ob)},p,q)$ ou plus simplement $conflict(p[ob],q[ob])$) ssi*

$$\begin{aligned}(&state(H^{(ob)} \circ p,\ q)\ \neq\ state(H^{(ob)} \circ q,\ p)) \vee \\ &(return(H^{(ob)},\ q)\ \neq\ return(H^{(ob)} \circ p,\ q)) \vee \\ &(return(H^{(ob)},\ p)\ \neq\ return(H^{(ob)} \circ q,\ p))\end{aligned}$$

1. Le prédicat $e \rightarrow e'$ est vrai si l'événement e précède l'événement e' dans l'histoire H. Il est faux dans le cas contraire.

Deux opérations qui ne sont pas en conflit sont dites compatibles.

En cas de conflit entre deux opérations, c'est-à-dire si le prédicat $conflict(H^{(ob)},p,q)$ vaut vrai, $return_value_independent(H^{(ob)},p,q)$ est vrai si la valeur de retour de q est independante du fait que p précède q ou non, i.e., $return(H^{(ob)} \circ p,\ q) = return(H^{(ob)},\ q)$; sinon q est *"return-value dependent"* de p (noté $return_value_dependent(H^{(ob)},p,q)$).

A.4 Visibilité et Ensemble des Conflits

La visibilité d'une transaction caractérise l'ensemble des effets sur les objets qui sont visibles pour cette transaction. Elle exprime l'accessibilité de la transaction aux mises à jour courantes subies par les objets.

Définition 14 (Vue d'une transaction) *La vue d'une transaction, notée $View_t$, spécifie les objets et l'état de ces objets visibles par une transaction t à un instant donné. $View_t$ est en fait une projection de l'histoire courante H_{ct} selon un prédicat. Plus formellement:*

$$View_t = Projection(H_{ct},\mathcal{P}redicate)$$

Définition 15 (Conflits d'une transactions) *L'ensemble des conflits d'une transaction t, noté $ConflictSet_t$, contient les opérations de l'histoire courante avec lesquelles les opérations invoquées par t peuvent entrer en conflit. Cet ensemble est en fait un sous-ensemble des événements présents dans H_{ct} satisfaisant un prédicat:*

$$ConflictSet_t = \{p_{t_i}[ob] \mid \mathcal{P}redicate\}$$

A.5 Exemples d'axiomatisation en ACTA

A.5.1 Dépendance de validation

$(t_j\ \mathcal{CD}\ t_i)$ [2] $\equiv$ si deux transactions t_i et t_j sont validées, alors la validation de t_i précéde la validation de t_j, plus formellement :
$(t_j\ \mathcal{CD}\ t_i)\ \equiv\ \mathbf{Commit}_{t_j} \in H \Rightarrow (\mathbf{Commit}_{t_i}\ \in\ H \Rightarrow (\mathbf{Commit}_{t_i} \rightarrow \mathbf{Commit}_{t_j}))$

A.5.2 Dépendance faible d'annulation

$(t_j\ \mathcal{WD}\ t_i)$ [3] $\equiv$ si t_i est annulée et t_j n'est pas encore validée, alors t_j est annulée, plus formellement :
$(t_j\ \mathcal{WD}\ t_i)\ \equiv\ \mathbf{Abort}_{t_i} \in H \Rightarrow (\neg(\mathbf{Commit}_{t_j} \rightarrow \mathbf{Abort}_{t_i}) \Rightarrow (\mathbf{Abort}_{t_j} \in H))$

A.5.3 Commencement–sur–Annulation

$(t_j\ \mathcal{BAD}\ t_i)$ [4] $\equiv$ la tansaction t_j ne peut pas s'exécuter avant que t_i ne soit annulée, plus formellement :
$\mathbf{Begin}_{t_j}\ \in\ H\ \Rightarrow\ \mathbf{Abort}_{t_i}\ \rightarrow\ \mathbf{Begin}_{t_j}$

A.6 Conclusion

Un modèle transactionnel formalisé en ACTA est donc composé des éléments suivants:

– La liste des événements significatifs permettant de créer, de terminer ou de valider une transaction,

2. $\mathcal{CD}$: *(Commit Dependency)*
3. $\mathcal{WD}$: *(Weak–Abort Dependency)*
4. $\mathcal{BAD}$: *(Begin–On–Abort Dependency)*

- La définition de la vue d'une transaction: $View_t$,
- La spécification de l'ensemble des conflits d'une transaction: $ConflictSet_t$,
- La liste des prédicats et de règles ACTA (les axiomes) caractérisant les histoires correctes.

Bibliographie

[Agr90] D. Agrawal and A. El. Abbadi. Transaction Management in Database Systems. In A. K. Elmagarmid, editor, *Database transaction models for advanced applications*. Morgan Kauffman, 1990.

[All95] Larry Allen, Gary Fernandez, Kenneth Kane, David Leblang, Debra Minard, and John Posner. ClearCase MultiSite: Supporting geographically-distributed software development. In Jacky Estublier, editor, *Software Configuration Management: Selected Papers of the ICSE SCM-4 and SCM-5 Workshops*, number 1005 in Lecture Notes in Computer Science, pages 194–214. Springer-Verlag, October 1995.

[Atr94] Atria Software Inc. ClearCase product summary. Technical report, Atria Software Inc., 24 Prime park Way, Natick, Massachusetts 01760, 1994.

[Aus62] J. L. Austin. *How to Do Things with Words*. Oxford University Press, Oxford, England, 1962.

[Bel91] N. Belkhatir, J. Estublier, and W. Melo. Adèle 2 : A Support to Large Software Development Process. In *Proceedings of the First International Conference on the Software Process*, 1991.

[Ben98] K. Benali, M. Munier, and C. Godart. Cooperation models in co-design. In *International Conference on Agile Manufacturing*, Minneapolis, USA, June 1998.

[Cha94] M. K. Chang and C. C. Woo. A speech-Act-Based Negotiation Protocol : Design, Implementation and Test Use. *ACM Transactions on Information Systems*, 12(4):360–382, October 1994.

[Chi98a] G. Chicoisne. Conversation et relations sociales pour des agents moins artificiels. Master's thesis, Institut national Polytechnique de Grenoble, 1998. DEA Sciences Cognitives.

[Chi98b] G. Chicoisne, P-M. Ricordel, S. Pesty, and Y. Demazeau. Outils et piste pour la pratique du dialogisme entre agents. *Journées francophones Intelligence Artificielle Distribuée et Systèmes multi-agents*, 1998.

[Chr94] P.K. Chrysanthis and K.Ramamritham. Synthesis of Extended Transaction Models. *ACM Transactions on Database Systems*, 19(3):451–491, september 1994.

[CNT] Continuus/CM: Change Management for Software Development. *http://www.continuus.com/developers/developersACED.html*.

[Dem91] Y. Demazeau and J-P. Müller. From Reactive to intentional Agents. *Decentralized Artificial Intelligence*, 2, 1991.

[Des87] G. Desanctis and R. B Gallupe. A Foundation for the Study of Group Decision Support Systems. *Management Science*, 33(5):589–609, May 1987.

[Ell94] C. Ellis and J. Wainer. A Conceptual Model of Groupware. In *Proceedings CSCW'94, ACM Conference on Computer Supported Cooperative Work*, pages 79–88, 1994.

[Far97] P. Faratin, C. Sierra, and N. Jennings. Negotiation Decision Functions for Autonomous Agents. *Int. Journal of Robotics and Autonomous Systems*, 1997.

[Fer95] J. Ferber. *Les Systèmes multi-agents: Vers une intelligence collective*. InterEditions, Paris, 1995.

[Gam94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns, Elements of Reusable Object–Oriented Software*. Addition–Wesley, Massachusetts, 1994.

[Gei97] J-M. Geib, C. Gransart, and P. Merle. *CORBA : des concepts à la pratique*. Editions MASSON, Paris, France, 1997.

[Gor97] T. F. Gordon and N. Karacapilidis. The Zeno Argumentation Framework. In *Proc. Int. Conf. On AI and Law (ICAIL)*, Melbourne, 1997.

[Huh99] M. N. Huhns and L. M. Stephens. Multiagent Systems and Societies of Agents. In G. Weiss, editor, *MultiAgent Systems, A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, Massachussets, 1999.

[Jel89] M. T. Jelassi and A. Forroughi. Negotiation support systems : An overview of design issues and existing sorftware. *Decison Support Systems*, 5:167–181, 1989.

[Kar97a] N. I. Karacapilidis and B. Trousse. Computer-supported argumentation for cooperative design on the world-wide web. In *Proceedings of the 2nd International Workshop on CSCW in Design (CSCWD'97)*, pages 96–103, Bangkok, Thailand, November 26-28 1997. International Academic Publishers.

[Kar97b] N. I. Karacapilidis, B. Trousse, and D. Papadias. Using Case-Based Reasoning for Argumentation with Multiple Viewpoints. In *Case-Based Reasoning Research and Development, Proceedings of the 2nd Int. Conference on Case-Based Reasoning (ICCBR-97)*, Lecture Notes in AI 1266, pages 541–552, Providence, Rhode, Island, July 25–27 1997. Springer-Verlag.

[Kon95] J.L. Koning, Y. Demazeau, B. Esfandiari, and J. Quinqueton. Quelques perspectives d'utilisation des Langage et Protocoles d'Interaction dans le contexte des Télécommunications. *Actes 3èmes Journeés Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents AFCET, AFIA, LIA*, pages 41–52, 1995.

[Koo88] C. C. Koo. A commitment-based communication model for distributed office environments. In *proc. of Conference on Office Information System*, pages 291–298, New York, 1988. ACM Press.

[Kra98] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[Lev83] S. C. Levinson. *Pragmatics*. Cambridge University Press, New York, 1983.

[Lim93] L. H. Lim and I. Benbasat. A theoretical perspective of negotiation support systems. *Journal of management Information Systems*, 9(3), 1993.

[Mal89] T. W. Malone, K. R. Grant K. Lai, R. Rao, and D. A. Rosenblitt. The Information Lens:An Intelligent System for Information Sharing and Coordination. *Technical Support for Work Group Collaboration*, pages 65–88, 1989.

[Mor77] I. Morley and G. Stephensen. *The Social Psychology of Bargaining*. Allen and Unwin, London, 1977.

[Mun99] Manuel Munier. *Une architecture pour intégrer des composants de contrôle de la coopération dans un atelier distribué*. Thèse en informatique, Université Henry Poincaré - Nancy I, Loria, janvier 1999.

[Mye89] R. B. Myerson. Credible negotiation statements and coherent plans. *Journal of economic theory*, 48:264–303, 1989.

[Mye91] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

[Nor86] D. A. Norman. Cognitive Engineering, User Centred System. In *New Perspectives on Computer Interaction*, pages 31–61. Lawrence Erlbaum Associates, New Jersey, 1986.

[Pan92] R. R. Panko. Extending the Use of Time Studies. In *Proceedings of the 25th Annual Hawaii International Conference on Systems Sciences*, 1992.

[Par98] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[Pen52] J. Pen. A general theory of bargaining. *American Economic Review*, 42:24–42, 1952.

[Pop93] P. Populaire, Y. Demazeau, O. Boissier, and J. Schiman. Description et Implementation de Protocoles de Communication en Univers Multi-Agents. *1ères Journées francophones Intelligence Artificielle Distribuée et Systèmes multi-agents*, Avril 1993.

[Pru81] D. G. Pruitt. *Negotiation Behavior*. Academic Press, 1981.

[Put87] L. L. Putnam and M. S. Poole. Conflict and negotiation. *In Handbook of Organizational Communication: An Interdisciplinary Perspective*, pages 344–369, 1987.

[Rod91] T. Rodden and G. Blair. CSCW and Distributed Systems: The problem of control. In J. Bannon, M. Robinson, and K. Schmidt, editors, *Proceedings of the second european conference on CSCW*, 1991.

[Sal95] D. Salber, J. Coutaz, D. Decouchant, and M. Riveill. De l'observabilité et de l'honnêteté : le cas du contrôle d'accès dans la communication homme-homme médiatisée. In *Proceedings IHM'95*, pages 27–34. CEPAD ed., 1995.

[Sch98] K. Schwarz, C. Türker, and G. Saake. Execution Dependencies in Transaction Closures. In M. Halper, editor, *Proc. of the 3rd Int. Conf. on Cooperative Information Systems, CoopIS'98, August 20–22, 1998, New York City, USA*, pages 122–131, Los Alamitos, CA, 1998. IEEE Computer Society Press.

[Sea69] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, New York, 1969.

[Sea75] J. R. Searle. A taxonomy of illocutionary acts. *Language Mind and Knowledgen, Minnesota Studies in the Philosophy of Science*, 7:344–369, 1975.

[Sin93] M. P. Singh. A Semantics for Speech Acts. *Annals of Mathematics and Artificial Intelligence*, 8(I–II):47–71, 1993.

[Smi80] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computing*, 29(12):1104–1113, 1980.

[Spe82] P. Sperber. *Fail-Safe Business Negotiating : Strategies and Tactics for Success*. Prentice Hall, December 1982.

[Ste87] M. Stefik, G. Foster, D. G. Borrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1), January 1987.

[Syc89] K. Sycara. Multiagent compromise via negotiation. *In Distributed Artificial Intelligence*, 2:119–137, 1989.

[Zac86] W. Zachary. A cognitively based functional taxonomy of decision support techniques. *Human–Computer Interaction*, 2(1):25–63, 1986.

[Zen97] D. Zeng and K. Sycara. Benefits of Learning in Negotiation. In *Proceedings of AAAI-97*, 1997.

[Zeu68] F. Zeuthen. *Problems of Monopoly and Economic Warfare*. Routledge and Kegan Paul, London, 1968.

Bibliographie

[Agr90] D. Agrawal and A. El. Abbadi. Transaction Management in Database Systems. In A. K. Elmagarmid, editor, *Database transaction models for advanced applications*. Morgan Kauffman, 1990.

[All95] Larry Allen, Gary Fernandez, Kenneth Kane, David Leblang, Debra Minard, and John Posner. ClearCase MultiSite: Supporting geographically-distributed software development. In Jacky Estublier, editor, *Software Configuration Management: Selected Papers of the ICSE SCM-4 and SCM-5 Workshops*, number 1005 in Lecture Notes in Computer Science, pages 194–214. Springer-Verlag, October 1995.

[Atr94] Atria Software Inc. ClearCase product summary. Technical report, Atria Software Inc., 24 Prime park Way, Natick, Massachusetts 01760, 1994.

[Aus62] J. L. Austin. *How to Do Things with Words*. Oxford University Press, Oxford, England, 1962.

[Bel91] N. Belkhatir, J. Estublier, and W. Melo. Adèle 2 : A Support to Large Software Development Process. In *Proceedings of the First International Conference on the Software Process*, 1991.

[Ben98] K. Benali, M. Munier, and C. Godart. Cooperation models in co-design. In *International Conference on Agile Manufacturing*, Minneapolis, USA, June 1998.

[Cha94] M. K. Chang and C. C. Woo. A speech-Act-Based Negotiation Protocol : Design, Implementation and Test Use. *ACM Transactions on Information Systems*, 12(4):360–382, October 1994.

[Chi98a] G. Chicoisne. Conversation et relations sociales pour des agents moins artificiels. Master's thesis, Institut national Polytechnique de Grenoble, 1998. DEA Sciences Cognitives.

[Chi98b] G. Chicoisne, P-M. Ricordel, S. Pesty, and Y. Demazeau. Outils et piste pour la pratique du dialogisme entre agents. *Journées francophones Intelligence Artificielle Distribuée et Systèmes multi-agents*, 1998.

[Chr94] P.K. Chrysanthis and K.Ramamritham. Synthesis of Extended Transaction Models. *ACM Transactions on Database Systems*, 19(3):451–491, september 1994.

[CNT] Continuus/CM: Change Management for Software Development. *http://www.continuus.com/developers/developersACED.html*.

[Dem91] Y. Demazeau and J-P. Müller. From Reactive to intentional Agents. *Decentralized Artificial Intelligence*, 2, 1991.

[Des87] G. Desanctis and R. B Gallupe. A Foundation for the Study of Group Decision Support Systems. *Management Science*, 33(5):589–609, May 1987.

[Ell94] C. Ellis and J. Wainer. A Conceptual Model of Groupware. In *Proceedings CSCW'94, ACM Conference on Computer Supported Cooperative Work*, pages 79–88, 1994.

[Far97] P. Faratin, C. Sierra, and N. Jennings. Negotiation Decision Functions for Autonomous Agents. *Int. Journal of Robotics and Autonomous Systems*, 1997.

[Fer95] J. Ferber. *Les Systèmes multi-agents: Vers une intelligence collective*. InterEditions, Paris, 1995.

[Gam94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns, Elements of Reusable Object–Oriented Software*. Addition–Wesley, Massachusetts, 1994.

[Gei97] J-M. Geib, C. Gransart, and P. Merle. *CORBA : des concepts à la pratique*. Editions MASSON, Paris, France, 1997.

[Gor97] T. F. Gordon and N. Karacapilidis. The Zeno Argumentation Framework. In *Proc. Int. Conf. On AI and Law (ICAIL)*, Melbourne, 1997.

[Huh99] M. N. Huhns and L. M. Stephens. Multiagent Systems and Societies of Agents. In G. Weiss, editor, *MultiAgent Systems, A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, Massachussets, 1999.

[Jel89] M. T. Jelassi and A. Forroughi. Negotiation support systems : An overview of design issues and existing sorftware. *Decison Support Systems*, 5:167–181, 1989.

[Kar97a] N. I. Karacapilidis and B. Trousse. Computer-supported argumentation for cooperative design on the world-wide web. In *Proceedings of the 2nd International Workshop on CSCW in Design (CSCWD'97)*, pages 96–103, Bangkok, Thailand, November 26-28 1997. International Academic Publishers.

[Kar97b] N. I. Karacapilidis, B. Trousse, and D. Papadias. Using Case-Based Reasoning for Argumentation with Multiple Viewpoints. In *Case-Based Reasoning Research and Development, Proceedings of the 2nd Int. Conference on Case-Based Reasoning (ICCBR-97)*, Lecture Notes in AI 1266, pages 541–552, Providence, Rhode, Island, July 25–27 1997. Springer-Verlag.

[Kon95] J.L. Koning, Y. Demazeau, B. Esfandiari, and J. Quinqueton. Quelques perspectives d'utilisation des Langage et Protocoles d'Interaction dans le contexte des Télécommunications. *Actes 3èmes Journeés Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents AFCET, AFIA, LIA*, pages 41–52, 1995.

[Koo88] C. C. Koo. A commitment-based communication model for distributed office environments. In *proc. of Conference on Office Information System*, pages 291–298, New York, 1988. ACM Press.

[Kra98] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[Lev83] S. C. Levinson. *Pragmatics*. Cambridge University Press, New York, 1983.

[Lim93] L. H. Lim and I. Benbasat. A theoretical perspective of negotiation support systems. *Journal of management Information Systems*, 9(3), 1993.

[Mal89] T. W. Malone, K. R. Grant K. Lai, R. Rao, and D. A. Rosenblitt. The Information Lens:An Intelligent System for Information Sharing and Coordination. *Technical Support for Work Group Collaboration*, pages 65–88, 1989.

[Mor77] I. Morley and G. Stephensen. *The Social Psychology of Bargaining*. Allen and Unwin, London, 1977.

[Mun99] Manuel Munier. *Une architecture pour intégrer des composants de contrôle de la coopération dans un atelier distribué*. Thèse en informatique, Université Henry Poincaré - Nancy I, Loria, janvier 1999.

[Mye89] R. B. Myerson. Credible negotiation statements and coherent plans. *Journal of economic theory*, 48:264–303, 1989.

[Mye91] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

[Nor86] D. A. Norman. Cognitive Engineering, User Centred System. In *New Perspectives on Computer Interaction*, pages 31–61. Lawrence Erlbaum Associates, New Jersey, 1986.

[Pan92] R. R. Panko. Extending the Use of Time Studies. In *Proceedings of the 25th Annual Hawaii International Conference on Systems Sciences*, 1992.

[Par98] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[Pen52] J. Pen. A general theory of bargaining. *American Economic Review*, 42:24–42, 1952.

[Pop93] P. Populaire, Y. Demazeau, O. Boissier, and J. Schiman. Description et Implementation de Protocoles de Communication en Univers Multi-Agents. *1ères Journées francophones Intelligence Artificielle Distribuée et Systèmes multi-agents*, Avril 1993.

[Pru81] D. G. Pruitt. *Negotiation Behavior*. Academic Press, 1981.

[Put87] L. L. Putnam and M. S. Poole. Conflict and negotiation. *In Handbook of Organizational Communication: An Interdisciplinary Perspective*, pages 344–369, 1987.

[Rod91] T. Rodden and G. Blair. CSCW and Distributed Systems: The problem of control. In J. Bannon, M. Robinson, and K. Schmidt, editors, *Proceedings of the second european conference on CSCW*, 1991.

[Sal95] D. Salber, J. Coutaz, D. Decouchant, and M. Riveill. De l'observabilité et de l'honnêteté : le cas du contrôle d'accès dans la communication homme-homme médiatisée. In *Proceedings IHM'95*, pages 27–34. CEPAD ed., 1995.

[Sch98] K. Schwarz, C. Türker, and G. Saake. Execution Dependencies in Transaction Closures. In M. Halper, editor, *Proc. of the 3rd Int. Conf. on Cooperative Information Systems, CoopIS'98, August 20–22, 1998, New York City, USA*, pages 122–131, Los Alamitos, CA, 1998. IEEE Computer Society Press.

[Sea69] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, New York, 1969.

[Sea75] J. R. Searle. A taxonomy of illocutionary acts. *Language Mind and Knowledgen, Minnesota Studies in the Philosophy of Science*, 7:344–369, 1975.

[Sin93] M. P. Singh. A Semantics for Speech Acts. *Annals of Mathematics and Artificial Intelligence*, 8(I–II):47–71, 1993.

[Smi80] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computing*, 29(12):1104–1113, 1980.

[Spe82] P. Sperber. *Fail-Safe Business Negotiating : Strategies and Tactics for Success*. Prentice Hall, December 1982.

[Ste87] M. Stefik, G. Foster, D. G. Borrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1), January 1987.

[Syc89] K. Sycara. Multiagent compromise via negotiation. *In Distributed Artificial Intelligence*, 2:119–137, 1989.

[Zac86] W. Zachary. A cognitively based functional taxonomy of decision support techniques. *Human–Computer Interaction*, 2(1):25–63, 1986.

[Zen97] D. Zeng and K. Sycara. Benefits of Learning in Negotiation. In *Proceedings of AAAI-97*, 1997.

[Zeu68] F. Zeuthen. *Problems of Monopoly and Economic Warfare*. Routledge and Kegan Paul, London, 1968.