Définition d'un modèle opérationnel de gestion dynamique des exceptions dans le cadre d'échanges B2B sans accord mutuel préalable.

Sébastien Picant

▶ **To cite this version:**

Sébastien Picant. Définition d'un modèle opérationnel de gestion dynamique des exceptions dans le cadre d'échanges B2B sans accord mutuel préalable.. Théorie de l'information [cs.IT]. Université de Caen, 2011. Français. ⟨NNT : ⟩. ⟨tel-01080706⟩

HAL Id: tel-01080706

https://hal.science/tel-01080706v1

Submitted on 6 Nov 2014

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.

HAL Authorization

UNIVERSITÉ de CAEN/BASSE-NORMANDIE

U.F.R. : Sciences

ÉCOLE DOCTORALE : SIMEM

THÈSE

présentée par

Sébastien PICANT

et soutenue

le 18 mars 2011

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN

spécialité : Informatique et applications

(Arrêté du 7 août 2006)

Définition d'un modèle opérationnel de gestion dynamique des exceptions dans le cadre d'échanges B2B sans accord mutuel préalable

MEMBRES du JURY

Olivier BOISSIER	Professeur	ENS Mines Saint-Étienne	(rapporteur)
Eugénio OLIVEIRA	Professeur	Université de Porto	(rapporteur)
Brahim CHAIB-DRAA	Professeur	Université de Laval - Québec	
Bruno ZANUTTINI	Maître de Conférence	Université de Caen	
Fabrice BOURGE	Ingénieur de recherche	Orange Labs	(invité)
Abdel-Illah MOUADDIB	Professeur	Université de Caen	(directeur)
François BOURDON	Professeur	Université de Caen	(co-directeur)

Mis en page avec la classe thloria.

Remerciements

Je tiens à témoigner ma reconnaissance envers Fabrice Bourge et Thierry Lemoisson pour m'avoir donné l'opportunité d'effectuer ma thèse au sein d'Orange Labs. Je porte une attention spéciale à Fabrice qui m'a encadré pendant ces trois années dans les locaux d'Orange Labs - Caen. Je remercie également particulièrement Abdel-Illah Mouaddib, mon directeur de thèse, pour m'avoir accompagné tout au long de mes travaux, notamment durant la phase de rédaction qui fut intense. Je remercie aussi François Bourdon, mon co-directeur de thèse, pour avoir accepté de suivre mes travaux.

Je remercie les rapporteurs, Olivier Boissier et Eugénio Oliveira ainsi que les examinateurs, Brahim Chaib-Draa et Bruno Zanuttini pour avoir accepté de faire partie des membres du jury et pour avoir pris le temps de juger mon travail, d'autant plus que certains sont venus de loin pour assister à la soutenance.

Je souhaite également exprimer ma sympathie envers les membres du laboratoire *GREYC* et principalement les membres de l'équipe *MAD*, ainsi que Wigdan et Julien, pour leur accueil chaleureux bien que je n'ai pas passé beaucoup de temps avec eux. Une spéciale dédicace à Romain Brixtel, jeune docteur avec qui j'ai suivi mon parcours universitaire et que je considère comme un modèle (ou pas...).

Je remercie chaleureusement les membres et ex-membres de l'ex-équipe *B2B* maintenant dénommé *NCIS* d'Orange Labs - Caen (vive les réorgs !) avec qui j'ai partagé des moments inoubliables. En particulier Tiphaine qui a partagé mon bureau pendant des mois qui ont dû lui paraître des années !

Enfin je remercie tous mes proches pour m'avoir soutenu, et surtout supporté durant ces 3 années. Je pense bien sûr en priorité à ma femme de pacs (ou Pacs-woman (?) [1]) Charlotte qui a subit mon aigreur quotidienne et aussi à mon chien Gaïa qui avait la vie plus ou moins cool selon mon niveau de stress. Mais je pense également à mes parents qui m'ont permis d'en arriver là et à tous mes amis (big bless aux anciens).

1.

Table des matières

Table des figures **vii**

Liste des Algorithmes **ix**

Introduction **1**

Introduction Générale 3

Objectifs 6

Structure du Manuscrit 7

I État de l'art **9**

Introduction **11**

1 La médiation de processus hétérogènes **17**

1.1 La médiation de processus 18

1.2 OWL-S 19

1.3 WSMO 20

1.4 Analyse 22

2 Les approches orientées SMA **23**

2.1 La communication entre agents 24

2.1.1 Les actes de langages 24

2.1.2 FIPA-ACL 25

2.1.3 L'approche sociale 26

2.2 Les engagements sociaux 27

2.3 Institutions Électroniques 30

2.3.1 Quelques travaux 31

2.3.2 Une institution pour le B2B 35

2.4 Analyse 38

Synthèse **41**

II Contribution **43**

Introduction **45**

3 Interprétation des interactions **51**

3.1 Contexte institutionnel 51

3.2 La sémantique des documents d'affaires 55

3.2.1 Les engagements métier 55

3.2.2 Les informations liées à la chorégraphie 59

3.3 Modélisation d'une chorégraphie 63

3.4 Exemple illustratif 67

3.5 Synthèse 71

4 Comportement de la plateforme **73**

4.1 Interprétation institutionnelle d'une action 73

4.1.1 Mise à jour des engagements courants 74

4.2 Comportement des agents « entreprise » 74

4.3 Comportement de l'agent « médiateur » 76

4.3.1 Règles sur les intentions 77

4.3.2 Règles sur les engagements courants 78

4.3.3 Algorithme de raisonnement de l'agent « médiateur » 79

4.4 Illustration 81

4.5 Capacités de médiation de la plateforme 83

5 Jeu de médiation **87**

5.1 La théorie des jeux 87

5.2 Échanges entre une entreprise et la plateforme : jeu à deux joueurs 89

5.2.1 Exemple illustratif : 91

5.3 Échanges entre les entreprises via la plateforme : jeu de médiation 93

5.4 Arbre d'un jeu de médiation 96

5.5 Exemple illustratif 96

5.6 Discussion 98

6 Interopérabilité entre deux systèmes d'informations **103**
6.1 Interopérabilité constitutive 103
6.2 Interopérabilité de médiation 104

Synthèse **105**

III Implémentation **109**

7 La plateforme de développement multi-agents JIA **111**
7.1 JADE 111
7.2 JSA 112
7.2.1 Agents sémantiques 112
7.2.2 Mécanisme d'interprétation 114
7.3 JIA 115
7.3.1 Capacités institutionnelles 116
7.3.2 Spécification des actions institutionnelles 116
7.3.3 Interprération de l'exécution d'une action institutionnelle 117

8 Mise en oeuvre du prototype **119**
8.1 Extension de la plateforme 120
8.1.1 Extension de la grammaire 120
8.1.2 Gestion du cycle de vie d'un engagement 120
8.2 Spécification des capacités d'interaction 121
8.3 Comportement de la plateforme 121
8.3.1 Comportement de l'agent « médiateur » 121
8.3.2 Comportement des agents « entreprise » 123
8.4 Expérimentation 124

Conclusion et perspectives **127**
Conclusion 129
Perspectives 131

Annexes **133**

Liste des échanges lors de l'expérimentation **135**

Bibliographie **139**

Table des figures

1 Échanges B2B entre processus d'affaires 4

2 Exemples de processus définis par NES (source : www.nesubl.eu) 14

1.1 Représentation d'un service avec OWL-S 19
1.2 *Service model* du service web d'Amazon 20
1.3 Eléments de WSMO 21

2.1 Processus de demande de devis représenté sous la forme d'un protocole d'engagements 28
2.2 Spécification d'un protocole d'engagement dans le langage $C+$ 29
2.3 Liens entre les différentes dimensions de la plateforme OMNI 32
2.4 Architecture de la plateforme AMELI 33

5 Plateforme de médiation jouant le rôle d'intermédiaire entre les SI des entreprises 46
6 Plateforme de médiation multi-agents 48

3.1 Architecture institutionnelle de la plateforme de médiation 54
3.2 Cycle de vie d'un engagement 57
3.3 Chorégraphie d'un processus de commande 68

4.1 Exemple de comportement 82
4.2 Problèmes d'interopérabilité entre chorégraphies 84
4.3 Exemple de problème d'ordonnancement (1) 85
4.4 Exemple de problème d'ordonnancement (2) 86

5.1 Exemple de jeu sous forme extensive 89
5.2 Chorégraphie spécifiant un processus de commande-paiement-livraison 92
5.3 Représentation des échanges entre un agent « entreprise » Agt_E et l'agent « médiateur » Agt_M par un jeu sous forme extensive 93
5.4 Chorégraphie spécifiant un processus de commande-paiement-livraison 99
5.5 Exemple d'un arbre de jeu d'un jeu de médiation 100

7.1 Diagramme UML d'un agent JSA 113
7.2 Mécanisme d'interprétation de la plateforme JSA 114

8.1 Architecture institutionnelle de la plateforme de médiation 119
8.2 Exemple de fichier de spécification des actions institutionnelles relatives à une sous-institution 122
8.3 Scénario d'expérimentation 124
8.4 Échanges entre les agents de la plateforme lors de l'expérimentation 125

Liste des Algorithmes

1 Interprétation institutionnelle de l'exécution d'une action 74
2 Récupération des pouvoirs institutionnels (méthode SearchInstitutionalPowers) . . 74
3 Mise à jour de l'état des engagements 75
4 Raisonnement de l'agent « entreprise » A_E 76
5 Récupération de la précondition institutionnelle (méthode GetInstitutionalCondition) 80
6 Recherche d'une action institutionnelle (méthode SearchInstitutionalAction) . . . 80
7 Raisonnement de l'agent « médiateur » 81

8 Construction de l'arbre de jeu (méthode GameTree) 97
9 Détermination des actions possibles (méthode GetChoices) 97
10 Construction du noeud fils (méthode ChildNode) 98

Introduction

Introduction Générale

Ces dernières années, le commerce électronique (ou e-commerce), qui désigne l'utilisation des nouvelles technologies dans les relations d'affaires, est devenu pour les entreprises un moyen stratégique dans la conduite de leurs activités d'affaires. En France il a représenté en 2009 un montant de transaction de l'ordre de 25 milliards d'euros (source : FEVAD [24]).

Il y a 6 grands domaines à distinguer dans le commerce électronique :

- le Business-to-Business (B2B) concerne les échanges électroniques interentreprises, i.e. les relations commerciales entre professionnels ;
- le Business-to-Consumer (B2C) correspond aux échanges entre les professionnels et les particuliers ;
- le Business-to-Employee (B2E) représente les relations entre les entreprises et leurs employés ;
- le Business-to-Administration (B2A) se rapporte aux échanges entre les professionnels et les administrations ;
- le Consumer-to-Consumer (C2C) désigne les échanges entre particuliers ;
- le Consumer-to-Administration (C2A) se rapporte aux échanges entre les particuliers et les administrations.

Ces échanges électroniques ont pour but la diffusion d'informations concernant la réalisation de transactions commerciales, la vente ou l'achat de produits ou services, ou encore la transmission de documents officiels. Dans ce manuscrit, nous nous intéressons aux échanges entre professionnels (le B2B).

Le B2B

La plupart des entreprises industrielles disposent d'un Système d'Information (noté SI). Un SI représente l'ensemble des éléments participant à la gestion, au traitement, au transport et à la diffusion de l'information au sein de l'entreprise. Un SI est, entre autres, composé d'un outil de gestion permettant aux entreprises de maîtriser leurs processus. Ces outils, appelés Progiciels de Gestion Intégrés (PGI), sont utilisés à la fois pour gérer les processus internes (production, facturation ...) et les processus externes (commandes, livraisons, approvisionnement, transactions financières...). Le domaine du B2B regroupe l'ensemble des processus externes, relatifs aux interactions entre les SI des entreprises. Historiquement, la mise en oeuvre du B2B a commencé vers la fin des années 1980 avec l'Electronic Data Interchange (EDI). L'EDI est un concept d'

échange de données informatisé visant à transférer d'application à application, à l'aide d'ordinateurs connectés sur un ou plusieurs réseaux, des données structurées selon un langage normalisé. L'idée générale de l'EDI est de dématérialiser les échanges de documents entre les entreprises afin de pouvoir automatiser leur émission et leur réception directement dans les SI. Les échanges se font donc entre les SI des entreprises, au travers de leurs processus d'affaires. Le processus d'affaires d'une entreprise décrit la suite d'activités à effectuer afin d'atteindre un objectif (ex. : le traitement d'une commande). La figure 1 (inspirée du processus de commande du standard UBL [63]) illustre les processus d'affaires de deux entreprises : une entreprise acheteuse et une entreprise vendeuse. L'ensemble des échanges effectués entre les deux entreprises s'appelle la chorégraphie.

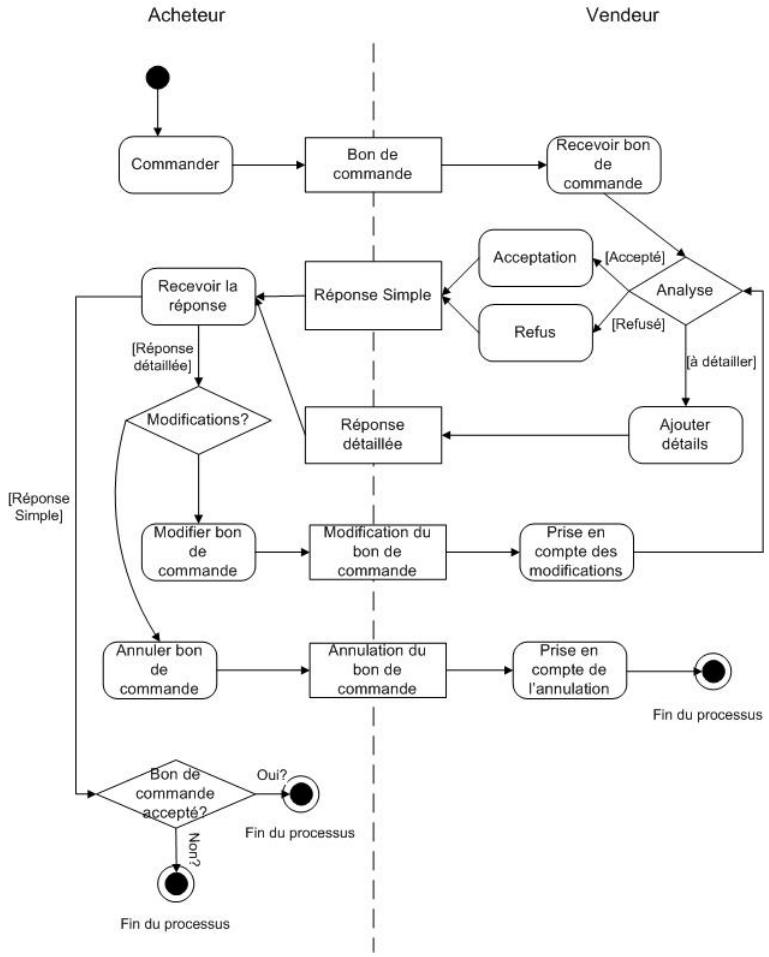


FIGURE 1 – Échanges B2B entre processus d'affaires

Les avantages offerts par l'EDI sont nombreux. Il permet aux entreprises de diminuer les coûts dus à la transmission et au traitement des données ainsi que les erreurs grâce à la diminution des interventions humaines. De plus il permet un meilleur développement de leurs activités commerciales grâce à sa facilité et sa rapidité d'utilisation. Ces avantages ont permis au B2B de largement se développer durant les années 90. Puis, avec l'avènement de l'internet,

l'utilisation des technologies issues du web a ouvert de nouvelles perspectives quant à l'usage du B2B. L'internet permet aux entreprises d'interagir de manière rapide et économique avec potentiellement n'importe quel partenaire dans le monde, ouvrant ainsi la porte aux TPE (Très Petites Entreprises) et PME (Petites et Moyennes Entreprises).

Limitations

La démocratisation du B2B a permis aux entreprises de généraliser les échanges électroniques professionnels en dématérialisant l'envoi et la réception des documents d'affaires. Aussi, l'utilisation de standards de données (p. ex. : UN/EDIFACT), formalisant la spécification des documents d'affaire, a permis la prise en compte du contenu de ces documents de manière automatique, et donc sans intervention humaine, au sein même des systèmes d'information. Cette prise en compte englobe à la fois la construction des documents et leur interprétation. Toutefois, l'automatisation des échanges reste une tâche difficile à réaliser. En effet, afin de pouvoir mettre en oeuvre des échanges électroniques entre leurs systèmes d'informations, deux partenaires commerciaux doivent satisfaire plusieurs conditions préalables. Tout d'abord, afin que les informations échangées puissent être interprétées par leurs deux systèmes d'informations, le langage définissant la structure et le contenu des documents doit être le même pour les deux partenaires (p. ex. : utilisation du même standard). Ensuite, ils doivent se mettre d'accord sur l'ordonnancement des documents échangés, constituant la chorégraphie des échanges. Enfin, ils doivent se décider sur les technologies compatibles à employer afin de pouvoir effectivement se transmettre les informations de manière fiable et sécurisée.

Le développement des Architecture Orientée Services (SOA), et en particulier l'apparition des services web, a grandement contribué à diminuer les problèmes d'incompatibilité technique en proposant un environnement standardisé servant d'interface entre les systèmes d'information. Les services web permettent à des applications distantes de communiquer entre elles à travers le réseau internet, indépendamment de tout langage de programmation et de toute plateforme d'exécution. Ils offrent aux entreprises la possibilité d'exposer publiquement les fonctionnalités d'interactions de leurs SI sous la forme de services pouvant être invoqués par les SI de leurs partenaires commerciaux. Néanmoins les difficultés liées à l'interopérabilité au niveau des données et des processus sont toujours d'actualités.

Ainsi, de nos jours, l'automatisation des échanges électroniques entre entreprises requiert que les systèmes d'informations des partenaires soient compatibles entre eux. Pour cela deux étapes principales, préalables aux échanges, sont nécessaires. Dans la première étape, les partenaires commerciaux se réunissent afin de se mettre d'accord sur le format des données échangées (structure et contenu des documents) ainsi que sur le processus d'affaire utilisé (choix d'une

chorégraphie commune pour les systèmes d'informations se rapprochant le plus des processus métier de chacun des partenaires). Durant cette étape, un contrat d'interchange est établi, définissant les conditions liées au déroulement des échanges, notamment les engagements d'affaires impliqués par l'envoi des documents d'affaires. La deuxième étape consiste alors pour chacun des partenaires à adapter/développer son système d'information afin qu'il soit en adéquation avec les caractéristiques définies lors de la première étape. Ces deux étapes constituent ce qu'on appelle la phase de conception. Celle-ci permet aux partenaires de préparer leurs systèmes d'information de manière à pouvoir s'engager dans des échanges de manière automatique. Une fois cette phase de conception passée, ils peuvent alors entrer dans la phase d'exécution. Cette seconde phase consiste simplement pour les partenaires à exécuter la chorégraphie commune définie et intégrée à leurs systèmes d'information lors de la phase de conception.

Du fait de la grande complexité des processus métiers implémentés dans les systèmes d'information actuels, l'automatisation des échanges électroniques, plus précisément la phase de conception, nécessite une grande expertise. En outre, chaque entreprise ayant un processus métier adapté à ses propres besoins, la conciliation des exigences de chaque partenaire en est rendue d'autant plus difficile. Par conséquent cette tâche requiert beaucoup de ressources à la fois humaines et financières. De plus, cette tâche étant ad-hoc à chaque partenaire, le même investissement est à nouveau nécessaire lorsqu'une entreprise souhaite intégrer un nouveau partenaire. Enfin, un changement dans le système d'information d'un de ses partenaires requiert également de recommencer une nouvelle fois la phase de conception. Dès lors, jusqu'à maintenant les TPE ainsi que la majorité des PME n'ont pas encore pu s'engager dans le B2B.

Objectifs

Comme nous venons de l'exposer, bien que le B2B ait permis de faciliter les interactions entre les partenaires commerciaux en facilitant la transmission des documents d'affaires et en permettant leur manipulation automatique au sein des systèmes d'informations, l'automatisation complète d'un processus d'échange est loin d'être une tâche triviale. Non seulement cette tâche requiert un grand investissement de la part des entreprises, mais elle entraîne également un fonctionnement complètement rigide qui doit être remis en question au moindre changement ou au moindre écart de fonctionnement dans l'exécution des échanges. En conséquence, l'objectif de cette thèse est de proposer un modèle opérationnel permettant à diverses entreprises de s'engager directement dans des échanges B2B avec n'importe quel autre partenaire sans avoir besoin au préalable de modifier leurs systèmes d'information. Ce modèle opérationnel doit donc apporter des solutions aux problèmes d'interopérabilité entre les systèmes d'informations des partenaires. Pour cela, nous devons concevoir un système de médiation permettant de surveiller de manière

dynamique le déroulement des échanges en cours. Ce système doit être capable d'interpréter les échanges afin de pouvoir détecter la survenue éventuelle d'une exception. Nos travaux se concentrent sur les problèmes d'interopérabilité entre les processus d'affaires. On parle alors de médiation de processus. Dans ce problème, une exception peut être due soit au comportement anormal du système d'information d'un partenaire soit à un problème d'incompatibilité entre les chorégraphies des partenaires. Le système de médiation permettra à la fois aux grandes entreprises de faire des économies considérables, mais également et surtout aux petites entreprises de pouvoir participer de manière active aux activités commerciales électroniques, leur permettant ainsi d'étendre leurs perspectives à de nouveaux marchés.

Structure du Manuscrit

Ce manuscrit est divisé en trois parties.

La partie I dresse un état de l'art des travaux visant à automatiser les échanges électroniques entre professionnels. Le chapitre I commence par exposer les activités visant à proposer des processus d'échanges standards afin d'uniformiser les systèmes d'information des entreprises. Ensuite, le chapitre 1 traite de la médiation de processus. Il décrit les travaux dans ce domaine basés sur les technologies du web sémantique tel que OWL-S ou WSMO. Enfin le chapitre 2 s'intéresse aux approches basées sur les systèmes multiagents. Après avoir abordé le thème de la communication entre agents, il s'intéresse aux travaux qui se sont appuyés sur les capacités de communication des agents pour représenter la sémantique des échanges B2B que ce soit sous la forme d'engagements sociaux ou au sein d'une institution électronique.

La partie II expose notre contribution théorique. Tout d'abord, les chapitres II et II présentent notre approche et décrivent les bases de la plateforme de médiation que nous proposons de concevoir. Cette plateforme est un système multiagents au sein duquel la sémantique des échanges de documents d'affaires est représentée. Cette représentation est exposée dans le chapitre 3. En particulier, nous proposons de formaliser la chorégraphie de chaque partenaire sous la forme d'actions institutionnelles en se basant sur un modèle d'engagement social. Le comportement de la plateforme est ensuite spécifié dans le chapitre 4. Les règles régissant le comportement de l'agent médiateur sont présentées avec les algorithmes associés. Dans le chapitre 5, nous définissons le concept de *jeu de médiation*. Ce concept permet de raisonner sur les futurs échanges possibles en représentant les échanges entre les partenaires sous la forme d'un arbre de jeu. Ceci permet de détecter les problèmes d'interopérabilité afin d'anticiper les éventuelles exceptions pouvant être engendrées. Enfin, dans le chapitre 6, nous discutons de la notion d'intéropérabilité dans le cadre de notre plateforme de médiation.

Dans la partie III, nous détaillons l'implémentation de la plateforme basée sur la plateforme JIA (JSA Institutionnal Add-on). JIA est une extension de la plateforme JADE (Java Agent Development Framework) basée sur l'extension JSA (JADE Semantics Add-on). Après avoir présenté ces plateformes, nous décrivons la mise en oeuvre de nos travaux.

Finalement, ce manuscrit se termine par un bilan des travaux effectués et ouvre des perspectives sur la poursuite de ces recherches.

Première partie

État de l'art

Introduction

Depuis les débuts du B2B, de nombreux travaux tentent de résoudre les problèmes d'interopérabilité entre les systèmes d'informations.

Définition. *L'interopérabilité est la capacité que possède un produit ou un système, dont les interfaces sont intégralement connues, à fonctionner avec d'autres produits ou systèmes existants ou futurs et ce sans restriction d'accès ou de mise en oeuvre.*
(source : `http://interoperability-definition.info/`).

Il y a 3 niveaux d'interopérabilité à distinguer : l'interopérabilité technique, l'interopérabilité des données et l'interopérabilité des processus.

L'interopérabilité technique concerne la compatibilité des moyens de transmission (du point de vue applicatif). Ces moyens assurent l'enveloppage, le routage et la sécurité des données, tout en garantissant la fiabilité du transport. Comme nous l'avons dit dans l'introduction, les services web permettent aux entreprises de faire communiquer leurs SI entre eux en exposant publiquement sur internet leurs capacités d'interaction sous la forme de services. Ces services sont décrits selon des standards basés sur le langage XML (p. ex. : WSDL [15]) et utilisent des protocoles ouverts afin d'interagir entre eux (p. ex. : SOAP [48]). La description de l'enchaînement des services est également possible grâce à des langages de description tel que BPEL [68]. Ainsi une entreprise peut décrire le protocole d'interaction de son SI correspondant à son processus d'affaires.

L'interopérabilité des données se rapporte à la structure et au contenu des documents d'affaires échangés. Chaque document est construit à partir d'un schéma de données, décrivant la syntaxe et la structure du document. De nombreux standards de données ont été développés et sont utilisés par les entreprises afin d'améliorer l'interopérabilité de leur système d'informations. Cependant ces standards sont limités car ils ne modélisent pas la sémantique des messages. Ainsi, pour que deux systèmes soient interopérables au niveau des données, ils doivent utiliser le même schéma afin de pouvoir traiter les informations reçues. C'est pourquoi la notion d'ontologie est

apparue. Une ontologie est un ensemble structuré de concepts, ainsi que de relations entre ces concepts, permettant de formaliser la sémantique exprimée par ces concepts. La spécification de la sémantique des messages permet aux SI d'interpréter les informations contenues dans les documents échangés [4]. Néanmoins, l'utilisation d'ontologies ne résout pas le problème d'interopérabilité. En effet, afin de pouvoir interpréter ces informations, le système doit connaître l'ontologie utilisée pour construire le message. S'il utilise une ontologie différente, il doit être capable de traduire les documents dans son ontologie. Ce problème est traité dans les travaux sur la médiation de données. Ces travaux ont pour objectifs de rendre compatibles deux systèmes utilisant des ontologies différentes. L'idée est alors de construire un alignement entre les deux ontologies, c'est-à-dire un ensemble de règles mettant en correspondance les éléments de ces deux ontologies. A partir de ces règles, il est alors possible de traduire un document créé à partir d'une ontologie en un document décrit selon l'autre ontologie et ayant la même sémantique. Il existe des outils permettant d'automatiser la construction d'un alignement. Cependant, il y a toujours des situations où il n'y a pas de correspondance entre les concepts de deux ontologies et qui requièrent donc l'intervention humaine afin d'être résolues.

L'interopérabilité des processus se rapporte aux processus d'affaires des entreprises. Le processus d'affaires d'une entreprise se compose d'une partie privée propre à son fonctionnement interne et d'une partie publique représentant la chorégraphie des échanges, i.e. l'ordonnancement de l'ensemble des interactions entre le système d'information de l'entreprise et le monde extérieur, c'est-à-dire les systèmes d'information de ses partenaires. Pour que les processus d'affaires de deux entreprises soient interopérables, les chorégraphies définies par leurs parties publiques doivent définir le même ordonnancement.

Les permiers travaux qui ont été effectués afin de résoudre les problèmes d'interopérabilité des processus ont concernés la standardisation de processus. La section suivante présente les démarches dans ce domaine.

La standardisation de processus B2B

Les problèmes d'interopérabilité dans les échanges B2B résultent de l'hétérogénéité des processus d'affaires. En effet, chaque entreprise définit son processus d'affaires en fonction de sa logique métier. Chaque entreprise ayant sa propre manière de fonctionner, les processus d'affaires développés sont spécifiques et les chorégraphies sont donc différentes. C'est pourquoi plusieurs organismes de standardisation mènent des travaux ayant pour but de permettre l'homogénéisation des processus d'affaires. Certains (p. ex. RosettaNet [55]) ont défini des processus standards mais la plupart spécifient des processus fournis simplement à titre indicatif. Ils décrivent les

contextes classiques d'utilisation des messages afin de guider les entreprises utilisant le standard dans la mise en oeuvre de leurs processus d'affaires. Le tableau 1 présente les différents standards définissant des processus. Il est à noter que NES [47] est cité bien que n'étant pas un standard. Il s'agit d'un organisme ayant pour but de faciliter le déploiement de processus utilisant le langage standard UBL [63]. Pour cela, ils ont sélectionné un sous-ensemble de UBL (considérant les messages pertinents réellement utilisés) et ont défini des processus précis à partir de cet ensemble.

Nom	Type de processus	Secteur d'activité
FIX	Indications - cas d'usages	Banques, courtiers, investisseurs
FpML	Normalisés	Finance
GS1	Indications - cas d'usages	Chaine d'approvisionnement dans les domaines de la santé, de la défense, du transport et de la logistique
NES	Normalisés	Facturation, commande
OAGi	Indications	Chimie
PIDX	Indications	Pétrole
RosettaNet	Normalisés	Chaîne d'approvisionnement
TWIST	Normalisés	Chaîne d'approvisionnement, paiement
OASIS UBL	Indications	Facturation, commande

Table 1 – Standards B2B définissant des processus d'affaires

Les processus spécifiés correspondent à des processus d'échanges élémentaires se rapportant à un type d'échange spécifique (p. ex. une demande de catalogue, un processus de paiement...). La combinaison de ces différents processus élémentaires permet alors de définir un processus d'affaires complet. L'objectif est de permettre aux entreprises de concevoir plus rapidement leur processus d'affaires en sélectionnant les types d'échanges qu'ils désirent y intégrer. Un exemple de processus définis par NES est présenté figure 2. Le processus de gauche représente un processus de commande et celui de droite un processus de paiement.

La spécification de processus, standardisés ou non, permet de favoriser l'interopérabilité entre les entreprises en uniformisant leurs processus d'affaires. Cependant, la multitude de compositions de processus élémentaires possibles afin de produire un processus d'affaires complet entraîne des problèmes d'interopérabilité entre les chorégraphies de processus issues du même standard. De plus, les entreprises ne font que s'inspirer de ces processus. En effet, elles doivent généralement les personnaliser afin de les adapter à leurs besoins spécifiques. Il en résulte donc encore des incompatibilités potentielles. Enfin, bien que le fait de se baser sur un standard commun permette un gain de temps non négligeable au moment de la définition d'une chorégraphie commune, la cooccurrence de multiples standards non interopérables entre eux nécessite que les entreprises se mettent d'accord sur un standard commun à utiliser et augmente donc également les problèmes d'interopérabilité.

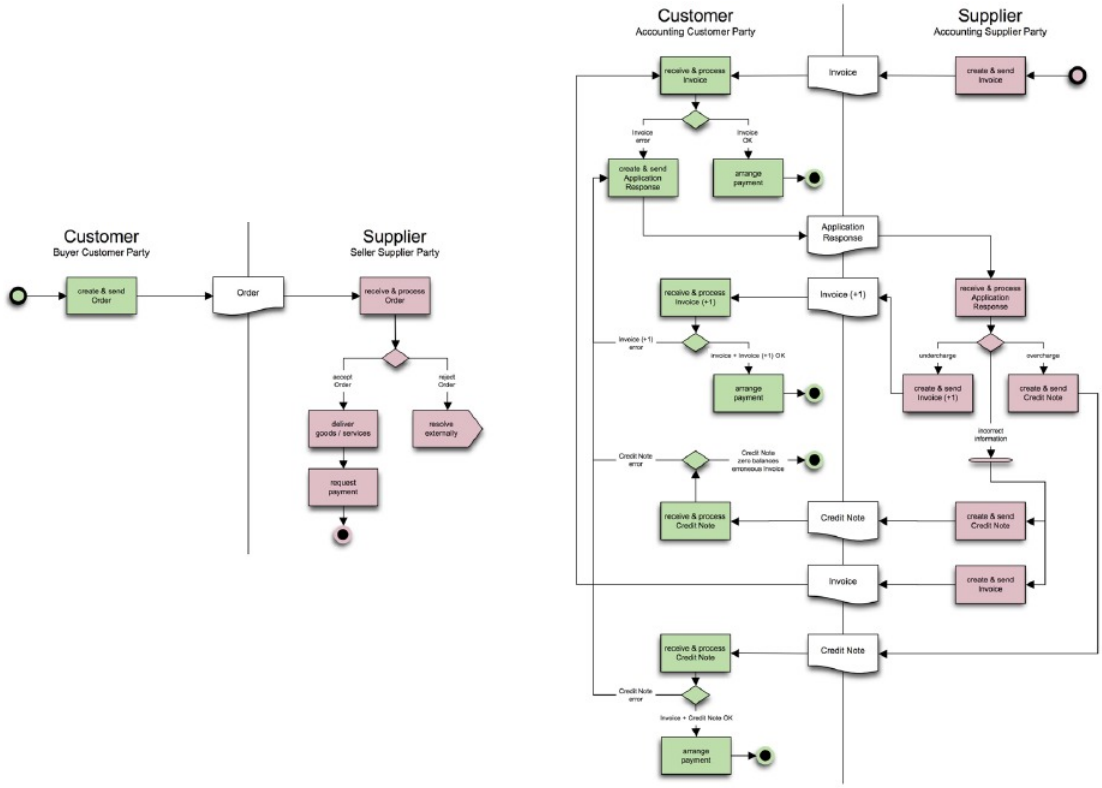


FIGURE 2 – Exemples de processus définis par NES (source : www.nesubl.eu)

Les standards facilitent le travail des entreprises que ce soit pour le développement de leur processus d'affaires ou pour la définition d'une chorégraphie commune. Cependant, ils ne permettent pas aux entreprises de créer indépendamment des processus d'affaires qui soient interopérables entre eux. L'organisme de standardisation de l'ONU, l'UN/CEFACT, a créé un groupe de travail (TBG14 [62]) dont le but initial était de fournir un catalogue de processus d'affaires communs pouvant être paramétrés et/ou enrichis dans le but de simplifier et d'harmoniser les processus d'échanges. Cependant, ces travaux n'ont pas réellement abouti et ont été suspendus depuis environ deux ans. L'homogénéisation des processus est un travail très complexe pour ne pas dire impossible. Les spécificités de chaque entreprise entraîneront toujours la coexistence de processus hétérogènes devant interagir entre eux. Pour régler le problème d'interopérabilité, il faut donc prendre en compte cet obstacle qu'est l'hétérogénéité des processus et essayer de le surmonter.

Comme nous l'avons évoqué précédemment, nos travaux se concentrent sur l'interopérabilité des processus d'affaires. Ce chapitre est consacré à un état de l'art sur les travaux liés à l'interopérabilité des processus. Différents travaux sur la médiation de processus ont émergé afin de rendre interopérables des processus hétérogènes. Le but est de laisser les entreprises développer de manière indépendante leurs processus d'affaires tout en étant capable par la suite d'assurer l'interopérabilité entre leurs processus, malgré leur hétérogénéité. Dans cette partie,

nous décrivons dans un premier temps les approches de médiation de processus s'appuyant sur les technologies du web sémantique. Nous exposons ensuite les techniques d'automatisation des échanges B2B basés sur les systèmes multi-agents.

1

La médiation de processus hétérogènes : approches basées Web Sémantique

La difficulté pour résoudre les problèmes d'interopérabilité entre des processus hétérogènes est de pouvoir interpréter la signification des échanges afin de détecter les incompatibilités éventuelles. L'absence d'information sémantique dans la description des processus d'affaires nécessite une intervention humaine afin de comprendre et de résoudre les problèmes d'interopérabilité. En effet, les technologies utilisées actuellement pour modéliser les processus d'affaires (p. ex. BPEL [68]) fournissent uniquement des informations syntaxiques. Or ces informations syntaxiques ne permettent pas de décrire les fonctionnalités des services ce qui est essentiel afin de déterminer de manière automatique les correspondances entre les différents processus. Le web sémantique permet de s'affranchir de cette limitation en permettant de représenter de manière explicite la sémantique des services afin qu'elle soit interprétable de manière automatique par les machines. Ainsi, les travaux présentés dans cette section s'appuient sur les technologies du web sémantique afin de proposer un système de médiation automatique entre processus.

Dans un premier temps nous définirons de manière plus précise en quoi consiste la médiation de processus. Nous présenterons ensuite les deux principaux standards du web sémantique ainsi que les techniques de médiation associées.

1.1 La médiation de processus

La médiation de processus consiste en un composant spécifique servant d'intermédiaire entre deux services, et plus particulièrement dans le domaine du B2B, entre les processus d'affaires des partenaires commerciaux. Comme son nom l'indique, ce composant joue le rôle de médiateur. Il a pour mission d'analyser les données échangées durant les processus d'interaction afin de détecter d'éventuelles discordances, c'est-à-dire des problèmes d'interopérabilité entre les chorégraphies, et de les résoudre de manière dynamique.

Différents types de problèmes d'interopérabilité peuvent se rencontrer entre deux services. Dans [16], Cimpian et Mocan définissent cinq types de problèmes :

- les messages non-attendus : lorsqu'un service envoie un message que le second service n'a pas prévu de recevoir ;
- les ordonnancements différents de messages : lorsque l'ordre des messages échangés par un service diffère de celui du second service ;
- la fusion de plusieurs messages : lorsque qu'un service attend de recevoir plusieurs messages qui sont envoyés sous la forme d'un seul message par le second service ;
- la scission d'un message : lorsque qu'un service attend de recevoir certaines informations sous la forme d'un unique message alors que le second service les transmet en envoyant plusieurs messages ;
- les messages requis et non envoyés : lorsqu'un service attend de recevoir un message que le second service n'a pas prévu d'envoyer.

Bien que Cimpian et Mocan aient classé tous ces types de problèmes dans le domaine de la médiation de processus, les problèmes de fusion et de scission sont plutôt à considérer dans celui de la médiation de données. En effet, ils concernent la représentation de l'information. Cette représentation étant basée sur les ontologies utilisées par les services, ces problèmes peuvent être réglés grâce à l'alignement d'ontologies, comme il est discuté dans [67].

Les différents travaux sur la médiation de processus qui ont été effectués s'appliquent à l'interopérabilité entre des processus d'interaction de services représentés sous la forme de services web sémantiques. Chacun de ses travaux se basent sur un standard particulier de description de services web sémantiques. Les deux standards principaux existants sont OWL-S et WSMO.

1.2 OWL-S

OWL-S [31] est une ontologie de services web basé sur le langage OWL (Web Ontology Language). Elle permet de décrire la sémantique d'un service web par un ensemble de trois éléments (cf. Figure 1.1) :

- le *service profile* : décrit les fonctionnalités du service Web ainsi que ses caractéristiques générales (propriétés non-fonctionnelles, Qualité de service...) ;
- le *service model* : spécifie comment fonctionne le service, et plus particulièrement comment interagir avec le service ;
- le *grounding* : fait le lien entre la description abstraite du fonctionnement du service (*service model*) et l'implémentation concrète du service Web (en WSDL).

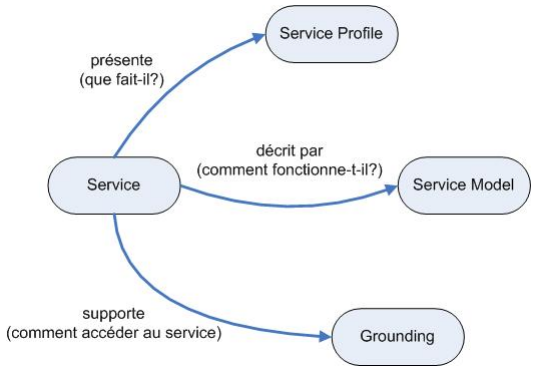


FIGURE 1.1 – Représentation d'un service avec OWL-S

Le *service model* permet de décrire le processus d'interaction du service web. Il est représenté par un processus (le *process model*) formé par la composition de plusieurs processus combinés en utilisant différents types de structures de contrôle : les séquences, les structures conditionnelles, les boucles, les choix non-déterminants et la concurrence. OWL-S distingue les processus atomiques et les processus composés. Un processus atomique est une opération indivisible correspondant à une interaction avec le client. Un processus composé décrit des relations entre différents processus (atomiques ou composés). Il est représenté sous la forme d'un arbre où les feuilles sont des processus atomiques et les noeuds correspondent à une structure de contrôle. Chaque noeud de l'arbre correspond donc à la racine d'un processus composé. En guise d'illustration, une partie du processus défini pour le service Web d'Amazon est présentée à la figure 1.2.

Chaque processus atomique est défini en termes d'entrées, sorties, pré-conditions et effets, ou IOPE (de l'anglais Input, Output, Preconditions, Effects). Les entrées et sorties décrivent la transformation de l'information effectuée par le service web. Les pré-conditions et effets représentent le changement d'état produit à la suite de son exécution.

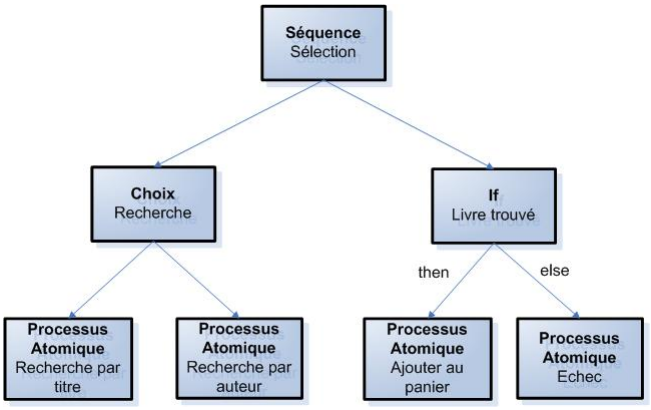


FIGURE 1.2 – *Service model* du service web d'Amazon

Dans le domaine du B2B, le *service model* permet de représenter le processus d'affaires d'une entreprise en décrivant les interactions nécessaires entre elle et son partenaire.

Dans [64], Roman Vaculìn et Katia Sycara se basent sur la description des processus en termes d'IOPE pour proposer un algorithme de médiation entre deux processus d'interactions décrits avec OWL-S. Leur algorithme est principalement composé de deux phases. La première consiste à rechercher tous les scénarii d'échanges possibles entre les deux processus en calculant les correspondances entre les IOPE des sous-processus les composant. Cet ensemble de scénarii est ensuite sauvegardé pour être utilisé lors de la seconde phase. Cette première phase est faite de manière statique préalablement à toute interaction entre les deux services web. La seconde phase consiste, au cours des interactions entre les services web, à déterminer à chaque étape de l'échange les actions d'interaction compatibles grâce à l'ensemble des scénarii pré-calculés. L'algorithme sélectionne alors l'action ayant le plus de chance de mener l'échange à son terme, c'est-à-dire celle qui, dans le plus grand nombre de cas, mènera à un scénario terminant dans un état de succès.

1.3 WSMO

WSMO (Web Service Modeling Ontology [69]) est une ontologie de services web basée sur le langage WSML (Web Service Modeling Language). Elle définit une architecture conceptuelle composée de quatre éléments (cf. Figure 1.3) :

– Les services web : décrivent sémantiquement les services web en termes de fonctionnalités, de propriétés non-fonctionnelles et d'interfaces. Une interface spécifie le processus d'interaction d'un service au travers de sa chorégraphie et de son orchestration [54]. La chorégraphie indique comment interagir avec le service web et l'orchestration précise comment le service web utilise d'autres services afin de satisfaire ses fonctionnalités. Elles

sont représentées sous la forme de machines à états abstraits [8]. Le changement d'état est défini par une transition conditionnelle, décrite sous la forme d'une règle avec des pré-conditions et des effets désignant une transformation d'un état initial en un état final sous une contrainte sur l'état initial. Une interface est l'équivalent d'un *process model* de l'ontologie OWL-S.

- Les objectifs : représentent les tâches effectuées par ces services.
- Les ontologies : spécifient de manière formelle la terminologie et la sémantique utilisées dans les autres éléments pour représenter l'information.
- Les médiateurs : fournissent un environnement pour la définition de mécanismes de médiation. Quatre types de médiateurs sont distingués :
  - OO médiateurs : permettent de faire la correspondance entre les terminologies de différentes ontologies ;
  - GG médiateurs : permettent de caractériser les relations entre objectifs pour savoir : (a) si des objectifs sont équivalents, (b) si un objectif est une spécialisation d'un autre, (c) si deux objectifs ont des sous-objectifs communs, ou (d) si un objectif est une combinaison de plusieurs objectifs ;
  - WG médiateurs : permettent de déterminer si un service web satisfait un objectif (de par ses fonctionnalités) ou si un certain objectif doit nécessairement être accompli pour qu'un service web puisse satisfaire ses fonctionnalités ;
  - WW médiateurs : permettent de résoudre les conflits de composition de services.

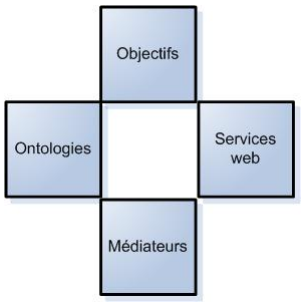


FIGURE 1.3 – Eléments de WSMO

La particularité de l'ontologie WSMO par rapport à OWL-S est que les médiateurs sont définis en tant que composants à part entière de son architecture alors qu'avec OWL-S ils doivent être définis eux-mêmes en tant que services web.

Cimpian et Mocan [16] proposent une solution de médiation de processus entre deux services définis dans la plateforme WSMX. WSMX est une architecture d'exécution implémentant les composants de WSMO. Leur approche consiste en un composant de médiation de processus intégré à la plateforme WSMX. Dans [67], Vitvar propose un modèle formel pour la médiation de services web sémantiques et détaille l'algorithme d'exécution du composant de médiation en se

basant sur ce modèle. Ce médiateur intercepte les messages envoyés par les services et mémorise l'état de leurs chorégraphies. En se basant sur les règles de transition des chorégraphies, il met à jour leurs états en appliquant l'ensemble des règles valides (i.e. les règles dont les pré-conditions sont satisfaites) et détermine les nouvelles données pouvant être envoyées.

1.4 Analyse

Les approches présentées dans cette section se basent sur les informations sémantiques fournies dans la description du protocole d'interaction des services web afin de proposer un système de médiation automatique entre des services ayant des protocoles d'interaction différents. Ces travaux proposent des méthodes permettant de résoudre de manière automatique et dynamique les éventuels problèmes d'interopérabilité au cours d'interactions entre services, en particuler lors d'échanges B2B. Ils montrent les possibilités offertes par l'ajout d'informations sémantiques dans des environnements ouverts où l'hétérogénéité des services a toujours été un handicap pour l'automatisation des interactions. En particulier, cela permet de déterminer les incompatibilités des services au niveau fonctionnel et de calculer les solutions pouvant être mises en oeuvre afin de les contourner.

La principale limitation de ces approches est qu'elles se basent chacune sur un standard particulier. Pour pouvoir en bénéficier, deux entreprises doivent donc avoir défini leurs services selon le même standard. De plus, les technologies du web sémantique ne sont actuellement pas mises en place et leur intégration au sein de systèmes d'informations risque de prendre plusieurs années.

Aussi, dans le cas particulier du B2B, les standards actuels du web sémantique ne permettent pas à ce jour de représenter les engagements commerciaux impliqués par les échanges de document d'affaires. Dès lors, bien que les approches présentées prennent en compte la sémantique des interactions, l'interprétation des conséquences de ces interactions sur les transactions commerciales reste à la charge des entreprises. Automatiser les échanges sans considérer ces conséquences peut alors mener la transaction dans un état de violation des engagements commerciaux impliqués, ce qui peut alors engendrer des coûts importants pour rétablir la situation.

D'autres travaux consistent à proposer de nouvelles approches pour le développement de systèmes d'informations en se basant sur le domaine des systèmes multi-agents. Ces nouvelles approches sont exposées dans la section suivante.

2

Les approches orientées SMA

Un système multi-agents est un système composé d'entités logicielles interagissant entre elles. Ces entités, appelées agents, sont des programmes autonomes évoluant au sein d'un environnement commun. Elles perçoivent certains éléments de leur environnement et peuvent agir sur celui-ci. Les capacités de communication et de raisonnement des agents permettent de concevoir des systèmes informatiques distribuées pouvant, d'une certaine mesure, être qualifiés d'"intelligents". Un système intelligent est un système capable d'interpréter, de manière autonome, les changements de son environnement et d'agir en conséquence en adaptant son comportement afin d'atteindre des objectifs. Ainsi, pour concevoir un tel système, plutôt que de devoir énumérer de manière exhaustive l'ensemble des situations pouvant se produire, il faut spécifier un comportement dynamique pour chacun des agents, déterminant les réactions des agents, i.e. les actions à effectuer pour atteindre leurs objectifs, en fonction de l'évolution de l'environnement.

Dans le domaine des échanges électroniques, et particulièrement dans le B2B, les principales contraintes proviennent de la rigidité des systèmes d'informations (SI). En effet, chaque SI définit un protocole d'interactions (la chorégraphie) qui spécifie de manière stricte comment interagir avec lui. Lors d'un échange, tout écart de comportement par rapport au protocole n'est pas géré par le SI et entraîne donc une exception. En partant de ce constat, l'idée des travaux présentés dans ce chapître est de proposer des environnements permettant le développement de SI flexibles, c'est-à-dire capables de communiquer de manière autonome entre eux en adaptant leur comportement en fonction des messages qu'ils reçoivent. C'est pourquoi ces travaux sont basés sur le domaine des systèmes multi-agents. Ceux-ci s'appuient sur les capacités de communication des agents afin de modéliser les échanges de documents d'affaires. Ce chapître débute donc par une présentation de la communication entre agents.

2.1 La communication entre agents

Afin de collaborer, de coopérer..., des entités indépendantes ont besoin d'intéragir entre elles afin de s'échanger des informations. Ces interactions peuvent être directes (échanges de messages) ou indirectes (par le biais de modifications de l'environnement). La communication désigne les interactions directes. La théorie des actes de langage [3] définit les bases de la communication en considérant la transmission d'un message non seulement par rapport à son contenu, mais également par l'action de communication qu'il représente.

2.1.1 Les actes de langages

La théorie des actes de langages a été étudiée à l'origine dans le domaine de la philosophie du langage. Le pionnier de cette théorie est Austin. Le point de départ de ses travaux consiste à considérer la communication comme étant l'accomplissement de certains types d'actes.

Austin distingue trois niveaux dans les actes de langages. L'*acte locutoire* désigne l'acte physique correspondant à la transmission de la suite de caractères constituant le message. L'*acte illocutoire* représente l'acte accompli par la transmission du message. Cet acte tient compte de la sémantique du message. Enfin, l'*acte perlocutoire* correspond aux effets produits sur les interlocuteurs suite à la transmission du message.

Le but de l'accomplissement d'un acte de langage est représenté par l'acte illocutoire. Il est appelé *but illocutoire*. Il désigne le but du locuteur lors de l'éxcécution de l'acte de langage (c'est-à-dire, pourquoi il réalise cet acte). Searle [57] propose de classer les actes de langages en fonction de leur but illocutoire. Il distingue cinq types d'actes :

- assertifs : engagent le locuteur sur l'existence d'un état des choses ;
- directifs : forcent l'interlocuteur à effectuer une action particulière ;
- commissifs : obligent le locuteur à réaliser une action particulière ;
- expressifs : expriment l'état psychologique du locuteur ;
- déclaratifs : changent l'état de l'environnement.

L'expression d'un acte de langage est caractérisée par une *performative*, représentant le but illocutoire de l'acte, et une proposition spécifiant le contenu de l'acte. Par exemple, l'énoncé « je te demande de fermer la porte » est un acte directif où « je te demande » est la performative et « de fermer la porte » est la proposition. Le but du locuteur est alors de forcer l'interlocuteur à réaliser la proposition, c'ést-à-dire à fermer la porte.

La théorie des actes de langages permet de formaliser la sémantique des actes de communication. Dans le domaine des systèmes multiagents, cette formalisation est essentielle pour permettre aux agents de communiquer entre eux en étant capables d'interpréter les messages qu'ils s'échangent. C'est pourquoi de nombreux travaux se sont basés sur cette théorie pour définir des langages de communication entre agents (ACLs). Par exemple, $FIPA-ACL$ [25] définit des « performatives » de bases en spécifiant formellement leur sémantique.

2.1.2 FIPA-ACL

L'organisation FIPA (Foundation for Intelligent Physical Agents [30]) a spécifié le langage de communication entre agents $FIPA-ACL$. Ce langage comporte vingt actes communicatifs, exprimés par des « performatives ».

Un message FIPA-ACL contient différents paramètres. Ainsi, dans un message FIPA-ACL sont spécifiés l'expéditeur et le destinataire du message, la « performative » utilisée (par exemple : *request*, *inform*, *propose*, *accept*...), le contenu du message, ainsi que d'autres informations pouvant être nécessaires à la compréhension du message.

Un acte de langage est caractérisé par le but de l'accomplissement de l'acte (appelé effet rationnel) et les conditions devant être satisfaites pour que l'acte soit exécuté (appelées préconditions de faisabilité). Ces caractéristiques définissent la sémantique de l'acte de langage. Elles sont décrites par un ensemble de formules exprimées dans un langage spécialisé, le langage FIPA-SL [26]. Ce langage, basé sur la logique BDI [52] provient principalement des travaux de Sadek [56].

Exemple : « Performative » *Inform*

$< i,\ inform(j, \phi) >$

$FP : B_i\phi \ \wedge \ \neg B_i(Bif_J\phi \ \vee \ Uif_j\phi)$ (i croit que ϕ et i ne croit pas que j croit ϕ ou il en a des doutes)

$RE : B_j\phi$ (j croit que ϕ)

Exemple d'un message FIPA-ACL : L'agent A informe l'agent B que la porte est verrouillée selon lui :

```
(inform
        :sender (agent-identifier :name A)
        :receiver (agent-identifier :name B)
        :content (door locked)
        :langage fipa-sl)
```

La précondition de faisabilité de cet acte de langage spécifie que, lors de l'envoi de ce message, l'agent A croit que la porte est verrouillée et il ne croit pas que l'agent B croit si la porte est verouillée ou non. L'effet rationnel indique qu'à la reception de ce message l'agent B croit que l'agent A croit que la porte est verrouillée.

Cette approche pour représenter les actes de langage permet de spécifier formellement la sémantique de ces actes en fonction des croyances et intentions des agents. Elle est qualifiée de mentaliste. Le fait que cette sémantique soit définie en fonction de l'état interne des agents est à l'origine de la plupart des critiques du langage FIPA-ACL. En particulier l'utilisation de ce langage est limitée dans un environnement ouvert où des agents hétérogènes peuvent communiquer entre eux. En effet, dans un tel environnement, l'état interne d'un agent n'est pas vérifiable ce qui ne permet pas d'assurer que le comportement des agents est conforme avec la semantique spécifiée. D'autres travaux se sont donc orientés sur l'aspect social des communications entre agents.

2.1.3 L'approche sociale

L'approche sociale, d'abord proposée par Singh dans [59], propose de formaliser la sémantique d'un acte d'un point de vue public. Cette approche est principalement basée sur la notion d'*engagement social*. La notion d'*engagement social* désigne un engagement entre deux agents (cf. Section 2.2). Cette notion permet de définir une sémantique formelle vérifiable pour la communication entre agents [32, 28]. Ainsi la communication entre les agents est modélisée sous la forme d'un ensemble d'engagements sociaux spécifiant l'état public du système en fonction de la sémantique des messages échangés.

L'avantage de l'approche sociale est de définir la sémantique des messages indépendamment de l'état interne des agents. Ainsi les agents n'ont pas besoin de raisonner en fonction des croyances ou intentions des autres agents, autorisant la communication entre des agents hétérogènes.

Dans la suite de ce chapitre, nous exposons différents travaux dans le domaine du B2B basés sur la communication entre agents. L'idée de ces travaux est de représenter les échanges entre

les partenaires commerciaux sous la forme de communications entre agents représentant ces partenaires. La section suivante présente de manière plus détaillée la notion d'engagement social et décrit son utilisation dans la représentation des échanges B2B.

2.2 Les engagements sociaux

Depuis plus de dix ans maintenant, la notion d'engagement est reconnue comme un concept important pour les systèmes multi-agents. Différents types d'engagements ont été mis en évidence, allant des engagements individuels (interne à l'agent) aux engagements collectifs (interne à un groupe d'agents), en passant par les engagements sociaux (entre 2 agents) [11]. Comme nous l'avons vu précédemment, la notion d'engagement social a été utilisée dans différents travaux pour modéliser la communication entre les agents [32, 28].

Un engagement social est un concept relationnel liant deux agents. Il représente le fait qu'un agent soit engagé envers un autre. Dans [60], Singh propose un modèle formel d'engagement social. Il définit un engagement social comme un état social entre deux agents où un agent (appelé débiteur) est engagé envers l'autre agent (appelé créditeur) à satisfaire une propriété (l'objet de l'engagement) dans un certain contexte. Ici, le contexte représente les normes et conventions s'appliquant dans le groupe dans lequel l'engagement est créé. Par la suite, ce dernier élément a été ignoré et un autre élément est apparu : la condition de validité. Ainsi, dans [41], un engagement social est défini par un quadruplet $C(x, y, p, q)$ où x et y représentent respectivement le débiteur et le créditeur de l'engagement, p la condition de validité de l'engagement et q l'objet de l'engagement. Dans ce cas, l'agent débiteur x est engagé envers l'agent créditeur y à satisfaire l'objet de l'engagement q à condition que la condition p soit vérifiée. On parle alors d'engagement conditionnel. Singh a également défini les actions permettant de manipuler les engagements. Ainsi, un engagement peut être créé, déchargé (lorsque l'objet de l'engagement est satisfait), annulé (selon des clauses d'annulation), dégagé (ce qui libère le débiteur de l'engagement), et délégué ou assigné à un autre agent (changement de débiteur ou de créditeur respectivement).

Dans le contexte du B2B, les échanges entre les partenaires reposent sur la notion de contrat. Un contrat est une convention entre deux ou plusieurs personnes définissant les engagements de ces personnes les une envers les autre. Le contrat portant sur les échanges B2B entre deux partenaires (le contrat d'interchange) définit donc les engagements sociaux impliqués au cours de ces échanges. Par conséquent, c'est naturellement que les travaux sur cette notion d'engagement social ont été appliqués au domaine du B2B.

Dans [42], Mallya et Singh définissent les protocoles d'engagements. Un protocole d'engage-

ment est un protocole d'interaction décrivant l'évolution des engagements engendrés au cours des interactions. Un exemple de protocole d'engagement est présenté à la figure 2.1. Dans [19] [18], Desai *et al.* appliquent ce concept au domaine du B2B en définissant les chorégraphies des échanges à un haut niveau d'abstraction. Ils proposent de spécifier un protocole d'engagement dans le langage de description d'actions $C+$ [35]. Nous présentons leurs travaux ci-dessous et donnons en exemple un protocole d'engagement exprimé dans le langage C+ à la figure 2.2.

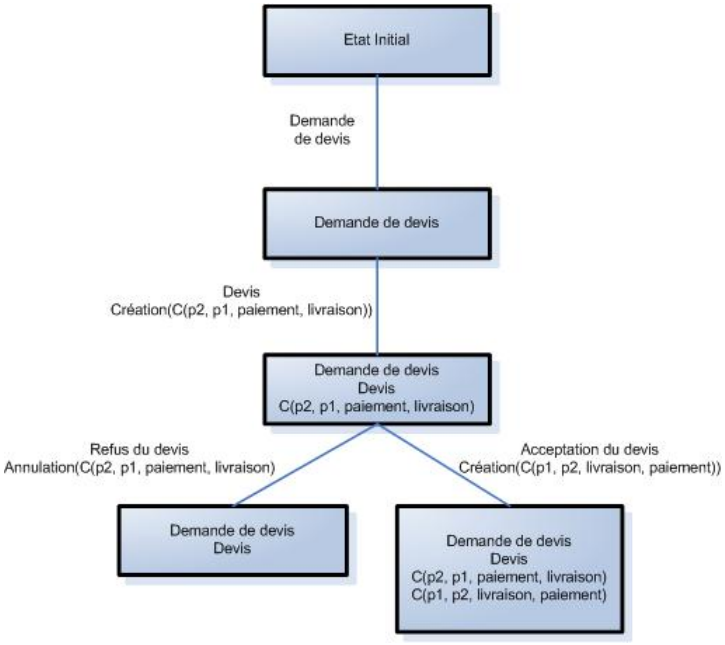


FIGURE 2.1 – Processus de demande de devis représenté sous la forme d'un protocole d'engagements

Un échange B2B est un échange de documents d'affaires déterminé par une chorégraphie et encadré par un contrat d'interchange. La chorégraphie décrit le protocole d'interaction (i.e. quels documents doivent être échangés et à quel moment dans la chorégraphie) et le contrat d'interchange sert de support juridique en définissant notamment les engagements métier impliqués par l'envoi de ces documents. Leurs travaux proposent donc de définir les processus d'affaires des entreprises à l'aide de protocoles d'engagements. Ainsi le processus d'affaires d'une entreprise est représenté par un ensemble d'interactions représentant les échanges des documents d'affaires. Les effets de ces interactions sont définis en terme d'opérations sur les engagements métier engendrés par les échanges B2B. Un exemple de protocole d'engagements représentant un processus de demande de devis entre deux partenaires, un acheteur p1 et un vendeur p2, est présenté à la figure 2.1. L'acheteur envoie une demande de devis au vendeur auquel le vendeur répond en envoyant un devis. L'envoi du devis crée un engagement du vendeur à l'acheteur à envoyer les produits concernés si le prix est payé. Le vendeur peut alors accepter ou refuser le devis. S'il l'accepte, il est à son tour engagé à payer le prix demandé pour la livraison des produits. S'il refuse le devis, l'engagement du vendeur est annulé. La spécification de ce protocole

```
:- include "commitment_ontology".

:- sorts
        Slot >> Item;    Slot >> ItemPrice;
        Role >> Customer;        Role >> Merchant.

:- objects
        demandeDevis(Acheteur, Vendeur, Produit) :: Message;
        devis(Vendeur, Acheteur, Produit, PrixProduit) :: Message;
        acceptationDevis(Acheteur, Vendeur, Produit, PrixProduit) :: Message;
        refusDevis(Acheteur, Vendeur, Produit, PrixProduit) :: Message;
        livraison(Produit) :: Condition;
        paiement(PrixProduit) :: Condition;
        p1 :: Acheteur; p2 :: Vendeur;
        monProduit :: Produit;  monPrix :: PrixProduit.

:- variables
        produit :: Produit;      prix :: PrixProduit.

nonexecutable act(demandeDevis(p1,p2,produit)) if
        fl(demandeDevis(p1,p2,produit)).

nonexecutable act(devis(p2,p1,produit,prix)) if
        -fl(demandeDevis(p1,p2,produit)) ++
        fl(devis(p2,p1,produit,prix)).

nonexecutable act(acceptationDevis(p1,p2,produit,prix)) if
        -fl(devis(p2,p1,produit,prix)) ++
        fl(acceptationDevis(p1,p2,produit,prix)) ++
        fl(refusDevis(p1,p2,produit,prix)).

nonexecutable act(refusDevis(p1,p2,produit,prix)) if
        -fl(devis(p2,p1,produit,prix)) ++
        fl(refusDevis(p1,p2,produit,prix)) ++
        fl(acceptationDevis(p1,p2,produit,prix)).

act(devis(p2,p1,produit,prix)) causes
        Creation(C(p2,p1,paiement(prix),livraison(produit))).

act(refusDevis(p1,p2,produit,prix)) causes
        Annulation(C(p2,p1,paiement(prix),livraison(produit))).

act(acceptationDevis(p2,p1,produit,prix)) causes
        Creation(C(p2,p1,livraison(produit),paiement(prix))).

:- query
        label :: 5;      maxstep :: 4;   0 : initial;
        maxstep: fl(acceptationDevis(p1,p2,produit,prix)) ++
                 fl(refusDevis(p1,p2,produit,prix)).
```

FIGURE 2.2 – Spécification d'un protocole d'engagement dans le langage $C+$

dans le langage $C+$ est présentée à la figure 2.2 (le fichier « commitment_ontology » inclus au début de l'exemple est un fichier définissant un engagement dans le langage $C+$).

L'intérêt de la description des processus d'affaires sous la forme de protocoles d'engagements est de s'affranchir des contraintes liées à la définition d'une chorégraphie rigide décrivant de manière stricte les messages à échanger. Ainsi, seuls les engagements métiers engendrés au cours des échanges sont considérés, laissant aux agents le choix des actions à effectuer pour les satisfaire.

Pour compléter ces travaux, Chopra et Singh [13] proposent une méthode permettant de vérifier l'interopérabilité entre deux partenaires. Le but est de s'assurer que les interactions entre les deux partenaires ayant défini leur processus d'affaires sous la forme de protocoles d'engagements engendrent les mêmes engagements. Il existe également d'autres travaux complémentaires portant sur des méthodes de modélisation des processus d'affaires à partir de « patrons » de protocoles d'engagements [61], sur la modification dynamique de protocoles existants [12] ou sur l'alignement des engagements au cours d'un échange [14].

Ces travaux sur la représentation des échanges B2B en termes d'engagements sociaux se basent sur le fait qu'un contrat est une notion essentielle dans le domaine du B2B. D'autres travaux (cf. Section 2.3) sont partis de cette même idée avec l'intention de concevoir un environnement institutionnel formalisant la sémantique des contrats d'interchange. Cet environnement a pour but de faire appliquer ces contrats et ainsi d'assurer le bon déroulement des échanges entre des partenaires B2B.

2.3 Institutions Électroniques

Nous évoluons au quotidien dans une société basée sur la notion d'institution. Une institution est une structure sociale composée d'un modèle organisationnel et d'un contexte institutionnel. Le modèle organisationnel spécifie l'organisation de la société d'agents (membres de l'institution) en terme de rôles, de contraintes, de relations entre les rôles et d'objectifs ; formant ainsi la structure de l'ensemble d'agents. Le contexte institutionnel est, quant à lui, constitué d'un ensemble de règles (appelées normes institutionnelles). Ces règles définissent les droits et les devoirs des agents.

Dans les relations d'affaires, cette notion d'institution est fondamentale car elle permet de représenter les règles auxquelles sont soumis les différents partenaires (p. ex. : les règles du commerce international). Une institution constitue donc un cadre réglementaire auquel ses membres peuvent se référer en cas de problèmes. Le but d'une institution électronique est alors de contrôler que les agents respectent les normes auxquelles ils sont soumis au sein de celle-ci. Nous présentons dans cette section quelques modèles d'institution. Nous détaillons ensuite une institution électronique conçue spécifiquement pour le B2B.

2.3.1 Quelques travaux

De nombreux travaux ont été menés afin de transposer la notion d'institution dans le domaine des systèmes multi-agents (cf. [34]). Nous présentons dans cette section trois approches principales pour concevoir une institution électronique.

Plateforme OMNI

La plateforme OMNI (Organizational Model for Normative Institution [66]) est basée sur le modèle OperA de Virginia Dignum [20], apportant la **dimension organisationnelle**, et le modèle HarmonIA de Javier Vàzquez-Scaleda [36], apportant la **dimension normative** (correspondant au contexte institutionnel). Nous présentons ces deux dimensions ci-dessous.

La dimension organisationnelle : Elle permet de définir la structure de l'organisation en termes de rôles et d'interactions entre les agents jouant ces rôles. Elle est composée d'une structure sociale et d'une structure d'interaction. La structure sociale spécifie les rôles que jouent les agents ainsi que les relations entre ces rôles. Elle est définie sous la forme d'un contrat social, mettant en relation un ensemble d'agents et une structure de rôles représentée par un arbre de dépendances entre les rôles. Chaque rôle est défini par un identifiant, un ensemble d'objectifs à atteindre, ainsi qu'un ensemble de normes le concernant. La structure d'interaction permet quant à elle de définir les cadres d'interaction qui décrivent comment les agents doivent intéragir entre eux en fonction de leur rôle. Ainsi un ensemble de *scènes* sont définies et liées les unes aux autres par des transitions. Chaque scène est décrite par ses acteurs (les rôles), ses résultats attendus et les normes régulant les interactions. La structure d'interaction est par la suite instanciée par des contrats d'interaction qui définissent les liens de communication réels entre les agents. Ces contrats sont basés sur des diagrammes d'états/transitions. Ces diagrammes permettent de contrôler l'exécution des normes. En effet, la violation d'une norme est détectée lorsque les interactions en cours mènent à un état d'exception. Un tel état définit également la réaction appropriée par rapport à l'exception (une sanction ou une correction). Ainsi chaque agent est lié par contrat à la plateforme en fonction du rôle qu'il joue. Ce contrat modélise les spécifications de son rôle. Si le comportement de l'agent ne respecte pas ces spécifications, il viole alors les termes de son contrat et peut être sanctionné.

La dimension normative : Elle permet d'exprimer le contexte institutionnel en définissant les normes utilisées pour spécifier les rôles ou les scènes. Ces normes sont d'abord définies avec le langage *CNorms* [65]. Ce langage utilise une logique déontique, temporelle, relative et

conditionnelle. Elles sont ensuite transformées en règles de comportement décrites dans la logique PDL [27]. Ces règles définissent de manière descriptive le comportement que doivent avoir les agents afin de respecter les normes les concernant.

En plus de ces deux dimensions, la plateforme OMNI est également composée d'une dimension ontologique. Cette dernière dimension augmente la généralité de la plateforme en permettant d'utiliser une ontologie spécifique pour chaque domaine d'application. La figure 2.3 illustre les liens entre les différentes dimensions.

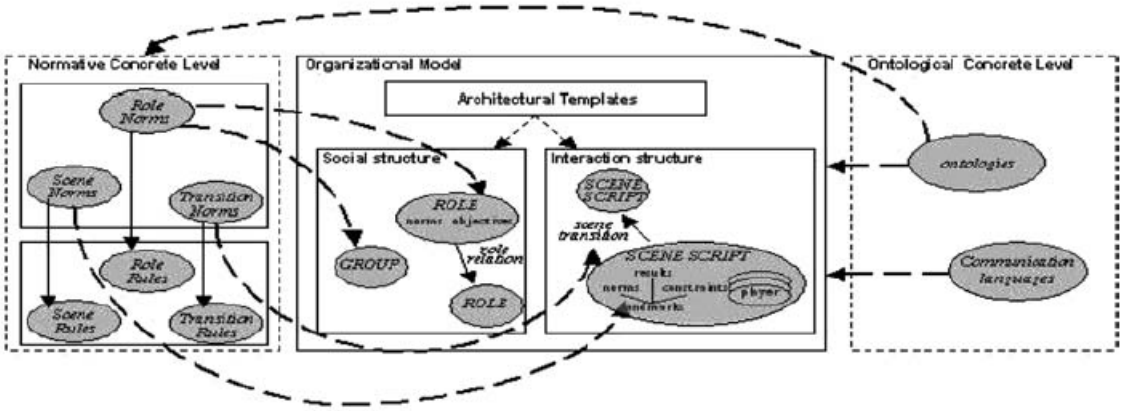


FIGURE 2.3 – Liens entre les différentes dimensions de la plateforme OMNI

La plateforme OMNI permet donc de définir les rôles que les agents devront jouer ainsi que les protocoles qu'ils devront suivre, et ceci sans avoir à faire d'hypothèses sur leur architecture interne. L'intégration de la dimension ontologique permet aux agents de comprendre l'organisation de la plateforme. L'approche OMNI considère également des agents institutionnels dont le rôle est de contrôler le respect des normes en fonction des comportements des autres agents, appelés agents externes.

Plateforme AMELI

AMELI [23] est une plateforme logicielle créée pour exécuter les institutions électroniques spécifiées avec le langage ISLANDER.

ISLANDER [22] est un langage basé sur XML. Il permet de définir des institutions électroniques sous la forme de systèmes dialogiques permettant l'échange de messages. Une institution définie avec le langage ISLANDER est composée d'un **cadre d'exécution dialogique**, d'une **structure performative** et de **normes**. Le **cadre d'exécution dialogique** définit l'ensemble des messages échangeables ainsi que les différents rôles des agents de l'institution et les liens

hiérarchiques entre ces rôles. La **structure performative** , similairement à l'approche OMNI, spécifie les interactions entre les agents en définissant des *scènes* liées entre elles par des transitions. ISLANDER permet également de définir des *protocoles* que devront suivre les agents dans une scène. Un protocole spécifie quel agent peut envoyer quel messages aux différents états de la conversation. Enfin les **normes** sont définies par des formules logiques exprimées par une conjonction de trois formules représentant respectivement une obligation, sa condition d'activation (la cause de l'obligation, c'est-à-dire l'envoi d'un certain message dans une certaine scène) et sa condition de satisfaisabilité. Elles permettent de représenter les conséquences des actions des agents sous la forme d'obligations.

Dans leurs travaux, Esteva et al. décomposent l'architecture d'une institution électronique en trois niveaux : le niveau de communication, le niveau social et le niveau des agents externes. La figure 2.4 illustre cette décomposition. Le niveau de communication désigne les moyens de transmission des messages entre les agents et le niveau des agents externes correspond aux agents autonomes faisant partie de l'institution. AMELI représente le niveau social. Il a pour but d'assurer le bon comportement des agents externes en veillant à ce qu'ils respectent les spécifications de l'institution. Ainsi la plateforme AMELI peut être vue comme une surcouche du système multi-agents. Elle joue le rôle d'intermédiaire entre les agents membres de l'institution via la couche de communication.

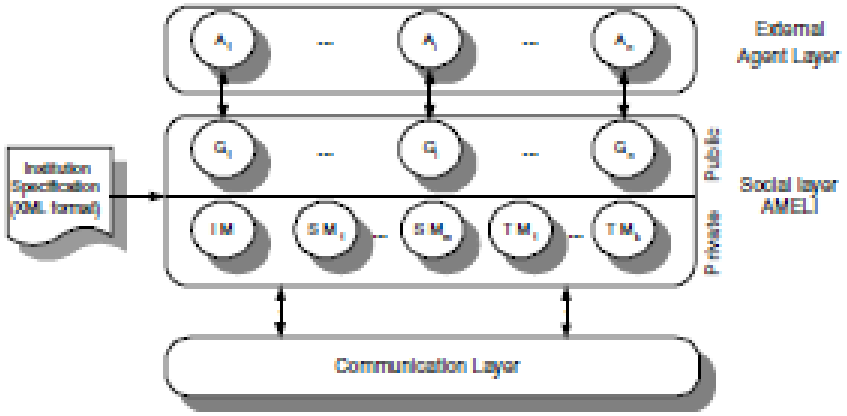


FIGURE 2.4 – Architecture de la plateforme AMELI

AMELI est constituée de quatres types d'agents :

- *Institution Manager* : Il y en a un par institution. Il est en charge de démarrer l'institution et d'autoriser les agents à entrer dans l'institution.
- *Transition Manager* : Il est en charge de gérer les transitions entre les scènes en contrôlant le comportement des agents.
- *SceneManager* : Il y en a un par scène. Il est responsable de l'exécution de la scène en s'assurant que les agents suivent le protocole défini.

– *Governor* : Il y en a un par agent externe. Il sert d'interface entre l'agent externe et l'institution et assure la médiation des interactions entre cet agent et les autres agents de la plateforme.

Le principe d'AMELI est donc de modéliser une institution en tant que surcouche sociale du système multi-agents. Chaque agent « externe » participant dans l'institution intéragit uniquement avec son agent *governor* correspondant. Ce dernier s'assure alors de fournir les informations nécessaires à l'agent externe. La particularité de la plateforme AMELI est qu'elle supervise les spécifications de l'institution et fait en sorte que les agents externes ne violent pas ces spécifications en contrôlant que leurs actions respectent les protocoles dans les scènes ainsi que les transitions entre les scènes.

Dans [38], les auteurs étendent ces travaux en présentant un mécanisme pour la résolution de conflits. Ils se basent sur des algorithmes d'unification du premier order et de satisfaction de contraintes pour détecter et résoudre les conflits tels que l'obligation et l'interdiction simultanées d'une action.

Travaux d'Artikis, Sergot et Pitt

Dans [1], Artikis et al. s'intéressent à la spécification de sociétés d'agents ouvertes, c'est-à-dire des système multi-agents dont les agents membres ont été développés de manières indépendantes et dont les charactéristiques internes ne sont pas publiquement connues. L'idée pour concevoir un tel système est alors de pouvoir contrôler que les interactions entre les agents respectent les spécifications du système malgré l'éventuelle inconsistance de leurs objectifs respectifs. Ils proposent donc de représenter ces spécifications sous la forme de normes qui régiront le comportement des agents. Pour cela ils introduisent le concept de *contrainte sociale*. Ils distinguent quatres niveaux de contraintes sociales :

– les capacités physiques : ce que les agents sont capables de faire ;
– les pouvoirs institutionnels : ce qu'impliquent les actions des agents ;
– les permission, obligations et interdictions : ce que les agents ont le droit de faire ;
– les sanctions et les politiques d'application des « lois » : conséquences du non-respects des obligations ou des interdictions.s

Leurs travaux se basent donc sur la notion de pouvoir institutionnel [37]. Cette notion permet de représenter les effets d'une action dans le contexte d'une institution. Ainsi suite à l'exécution d'une action, des faits institutionnels tels qu'une obligation ou une interdiction peuvent être

déduits, sous certaines conditions, en plus des effets réels de l'action. Ceci permet ainsi de faire évoluer l'*état social* du système au cours des interactions.

Ils définissent également la notion de *rôle social*. Un rôle social est défini par un ensemble de conditions que doit satisfaire un agent pour pouvoir jouer ce rôle, ainsi qu'une ensemble de contraintes que tout agent jouant ce rôle doit respecter.

Cette approche permet donc de définir le comportement attendu d'un ensemble d'agents, au sein d'un système multi-agents, au travers des actions qu'ils effectuent. Le modèle qu'ils proposent est indépendant du langage de modélisation utilisé. Dans [2], les auteurs proposent deux solutions pour la formalisation et l'exécution de leur modèle : par le langage de description d'actions $C+$ [35], ou en utilisant la logique du calcul d'évènements (*Event Calculus* [39]). Ils analysent ainsi les différences entre ces formalismes, chacun ayant ses avantages et ses inconvénients.

Les trois approches que nous venons de décrire ont pour objectif de spécifier les règles de comportements auxquels peuvent être soumis des agents au sein d'un système multi-agents. Les notions de rôles et de normes permettent de représenter l'organisation du système et les contraintes qui en découlent. Dans la section suivante, nous détaillons une approche spécifique au B2B.

2.3.2 Une institution pour le B2B

Dans [9], Lopes Cardoso et Oliveira définissent une institution électronique pour le B2B. Leur objectif est de proposer un environnement ouvert permettant de spécifier les normes relatives aux relations d'affaires. Le but d'un tel environnement est de pouvoir réguler le comportement d'agents hétérogènes agissant pour le compte des entreprises afin d'assurer le bon déroulement des interactions dans le cadre d'échanges B2B.

Pour cela, ils formalisent plusieurs concepts dans le métalangage EBNF (*Extended Backus-Naur Form*) :

– Rôle : un rôle est associé à un agent. Un agent peut jouer un ou plusieurs rôles. A chaque rôle est rattaché un ensemble de règles auxquelles sont soumis les agents jouant ce rôle.

$$\begin{aligned} < AgentRole > &::= agentRole(< Agent >, < Role >) \\ < Agent > &::= agent_identifier \\ < Role > &::= role_identifier \end{aligned}$$

– Fait brut/fait institutionnel : les faits bruts correspondent aux actions de communication effectuées par les agents au sein de la plateforme. Les faits institutionnels sont des faits considérés vrais dans le contexte d'une certaine institution.

$$
\begin{aligned}
< BruteFact > &::= illocution(< Sender >, < Content >) \\
< Sender > &::= < Agent > \\
< Content > &::= atomic_formula
\end{aligned}
$$

$$
\begin{aligned}
< InstitutionalFact > &::= [\text{“}[\text{“} < Ctx > \text{“}]\text{“}]ifact(< IFact >, < Timestamp >) \\
< Ctx > &::= context_identifier \\
< IFact > &::= atomic_formula \\
< Timestamp > &::= temporal_reference
\end{aligned}
$$

– Obligation : Les obligations et les faits institutionnels déterminent la *réalité institutionnelle*. Ce concept déontique d'obligation est complété par les notions de satisfaction et de violation.

$$
\begin{aligned}
< Obligation > &::= [\text{“}[\text{“} < Ctx > \text{“}]\text{“}]obligation(< Agent >, < IFact >, < Deadline >) \\
< Fulfillment > &::= [\text{“}[\text{“} < Ctx > \text{“}]\text{“}]fulfilled(< Agent >, < IFact >, < Timestamp >) \\
< Violation > &::= [\text{“}[\text{“} < Ctx > \text{“}]\text{“}]violated(< Agent >, < IFact >, < Timestamp >) \\
< Deadline > &::= < Timestamp > [\text{“} + \text{“} time_units]
\end{aligned}
$$

– Règle constitutive/règle institutionnelle : les règles constitutives font le lien entre les faits bruts et les faits institutionnels. Elles déterminent l'implication des faits institutionnels à partir des faits bruts. Les règles institutionnelles permettent de définir les interdépendances entres les éléments de la réalité institutionnelle (faits institutionnels, obligation...).

$$
\begin{aligned}
< ConstitutiveRule > &::= < BruteFacts > \text{“} \wedge \text{“} < AgentRole > \text{“} \rightarrow \text{“} < InstitutionalFacts > \\
< BruteFacts > &::= < BruteFact > \text{“} \wedge \text{“} < BruteFact > \\
< InstitutionalFacts > &::= < InstitutionalFact > \text{“} \wedge \text{“} < InstitutionalFact >
\end{aligned}
$$

$$
\begin{aligned}
< InstitutionalRule > &::= [\text{“}[\text{“} < Ctx > \text{“}]\text{“}] < Antecedent > \text{“} \rightarrow \text{“} < Consequent > \\
< Antecedent > &::= < Prem > \text{“} \wedge \text{“} < Prem > \text{“} \wedge \text{“} \neg < Prem > \\
< Consequent > &:= < Conc > \text{“} \wedge \text{“} < Conc > \\
< Prem > &::= < InstitutionalFact > | < Obligation > | < Fulfillment > | \\
&\quad < Violation > | < RelCondition > \\
< Conc > &::= < InstitutionalFact > | < Fulfillment > | < Violation > | \\
&\quad institutional_procedure \\
< RelCondition > &::= < Operand >< RelOperator >< Operand > \\
< Operand > &::= variable|expression \\
< RelOperator > &::= \text{“} < \text{“}|\text{“} > \text{“}|\text{“} \leq \text{“}|\text{“} \geq \text{“}|\text{“} = \text{“}
\end{aligned}
$$

– Normes : les normes sont des règles spécifiant les obligations auxquelles sont soumis les agents dans certaines situations. Elles représentent le comportement attendu des agents.

$$
\begin{aligned}
< Norm > &::= [\text{“}[\text{“} < Ctx > \text{“}]\text{“}] < Situation > \text{“} \rightarrow \text{“} < Prescription > \\
< Situation > &::= < Cond > \text{“} \wedge \text{“} < Cond > \text{“} \wedge \text{“} \neg < Cond > \\
< Prescription > &::= < Obligation > \text{“} \wedge \text{“} < Obligation > \\
< Cond > &::= < InstitutionalFact > | < Fulfillment > | < Violation > | \\
&\quad < RelCondition > | < ContextualInfo > \\
< ContexttualInfo > &::= \text{“}[\text{“} < Ctx > \text{“}]\text{“} \ atomic_formula
\end{aligned}
$$

Lopes Cardoso et Oliveira distinguent trois types de normes. Les normes *institutionnelles*, les normes *constitutionnelles* et les normes *opérationnelles* (ou *contractuelles*). Les premières désignent les normes prédéfinies dans l'institution (applicables à l'ensemble des agents ou rattachées à des rôles). Les normes constitutionnelles sont des normes créées dans le cadre d'un groupe d'agents afin d'établir une coopération entre eux. Enfin, les normes opérationnelles représentent les engagements contractuels entre deux agents. Ces normes sont un moyen de spécifier les règles définies dans un contrat entre les agents en déterminant les obligations qu'ils doivent satisfaire en fonction des actions qui sont effectuées. Des règles définissent les conditions de satisfaction et de violation d'une obligation, permettant ainsi de contrôler le bon comportement des agents :

$$
\begin{aligned}
[?Ctx]\ ifact(?IFact, ?T) \wedge obligation(?Agent, ?IFact, ?Deadline) \wedge ?T < ?Deadline \\
\rightarrow fulfilled(?Agent, ?IFact, ?T)
\end{aligned}
$$

$$
\begin{aligned}
[?Ctx]\ time(?Deadline) \wedge obligation(?Agent, ?IFact, ?Deadline) \wedge \neg fulfilled(?Agent, ?IFact, ?) \\
\rightarrow violated(?Agent, ?IFact, ?Deadline)
\end{aligned}
$$

En cas de violation, l'agent auquel se rapporte l'obligation est alors considéré comme responsable, d'autres règles pouvant par la suite entraîner des sanctions.

Dans [10], Lopes Cardoso et Oliveira complètent le notion d'obligation en y ajoutant ce qu'ils appellent la contrepartie, représentant l'agent bénéficiaire de l'obligation. De plus, il considèrent la fenêtre de temps dans laquelle l'obligation doit être satisfaite afin de prendre en compte les cas où la satisfaction anticipée d'une obligation est indésirable (pouvant par exemple entraîner des coûts de stockage). Ainsi une obligation est modélisée par un titulaire (l'agent qui est obligé), une contrepartie (l'agent envers qui le titulaire est obligé), un fait institutionnel (ce dont est obligé le titulaire) et une date de début et de fin. Sous la forme EBNF ont obtient alors l'expression suivante :

$$
\begin{aligned}
< Obligation > ::= [\text{“}[\text{“} < Ctx > \text{“}]\text{“}] obligation(< Agent >, < Agent >, < IFact >, \\
< Liveline >, < Deadline >)
\end{aligned}
$$

L'intérêt des travaux de Lopes Cardoso est Oliveira est qu'ils proposent un environnement permettant de modéliser les normes auxquelles sont soumises des entreprises dans leurs relations d'affaires, et ainsi de spécifier les contrats entre les partenaires commerciaux. De plus, leur approche est non-intrusive, c'est-à-dire qu'elle laisse leur entière autonomie aux agents représentant les entreprises. En effet, leur environnement n'impose pas un comportement spécifique aux agents en leur contraignant de respecter les engagements engendrés au cours des échanges. Leur approche consiste à fournir un cadre formel pour représenter le contrat définissant ces engagements afin de suivre l'évolution de la relation d'affaires. Ils laissent ainsi les agents adapter leur comportement en fonction de l'état courant des échanges et de leurs intentions.

2.4 Analyse

Engagements et responsabilités sont deux notions essentielles dans les relations d'affaires. Les approches présentées précédemment formalisent ces notions afin de prendre en compte la sémantique métier des échanges lors de la modélisation des processus d'affaires. Elles proposent de représenter l'ensemble des SI par un système multi-agents où chaque agent représente le SI d'une entreprise. En effet, les SI des entreprises peuvent être vus comme des entités autonomes capables de communiquer entre elles au sein d'un environnement commun (p. ex. : internet), ce qui est la définition d'un agent.

L'approche utilisant le modèle d'engagement social tente de s'affranchir des contraintes statiques liées à la description syntaxique des processus d'affaires en proposant une représentation se basant sur les engagements d'affaires impliqués au cours des échanges. L'état d'un échange est alors déterminé par l'ensemble des engagements courants. Cela permet ainsi de contrôler les interactions en détectant l'évolution des engagements (en particulier la satisfaction ou de la violation d'un engagement).

Les approches basées sur la notion d'institution électronique quant à elles, et en particulier l'approche de Lopes Cardoso et Oliviera (cf. section 2.3.2), offrent un cadre normatif aux agents leur permettant de spécifier les contrats qui les lient. Ainsi lors d'échanges entre deux partenaires, la plateforme peut contrôler les échanges en interprétant les obligations impliquées par l'envoi des documents d'affaires selon les règles définies dans le contrat préalablement établi.

Ces approches ont pour objectif de permettre aux entreprises de développer des agents autonomes ayant un comportement correspondant à leur besoins métiers. Ces agents peuvent ainsi dialoguer avec les agents représentant les entreprises partenaires en étant capables d'interpréter la sémantique des messages. Leur raisonnement se base sur la sémantique des échanges plutôt

que sur la syntaxe, ce qui apporte plus de flexibilité aux échanges. De plus, elles fournissent un mécanisme de contrôle des interactions permettant de détecter le mauvais comportement d'un partenaire, c'est-à-dire le non respect des engagements engendrés par les documents d'affaires échangés.

Une autre approche connue est celle de Minsky. Dans [43], Minsky et Ungureanu se concentre sur la coordination d'agents hétérogènes dans un système ouvert. Leur approche consiste à représenter explicitement les spécifications du système sous la forme d'une *loi* définissant les effets impliqués par les évènements (en particulier les interactions) qui se produisent au sein du système. Un *état de contrôle* est alors déterminé pour chaque agent, désignant l'état individuel de l'agent en fonction de la *loi*. Dans le domaine du B2B, la difficulté est que si les partenaires ne se sont pas préalablement mis d'accord sur des spécifications communes, alors chaque agent dispose potentiellement d'une loi spécifique. Dans [44], Minsky et Ungureanu s'intéressent à ce problme d'interopérabilité en proposant d'introduire ce qu'ils appellent un *portail*. Un portail permet de faire la correspondance entre deux lois différentes. Ce concept permet alors de prendre en compte la politique des autres agents dans la mise à jour de l'*état de contrôle* d'un agent afin d'éviter tout mauvais comportement. Tout comme les travaux d'Esteva et al. sur la plateforme AMELI [23] (cf. section 2.3.1), l'approche de Minsky et Ungureanu est une approche régulatrice. C'est-à-dire qu'elle a pour but d'assurer le bon comportement des agents au sein du système. Ainsi il ne prennent pas en compte la possibilité que le comportement d'un agent ne respecte pas ses spécifications et ne propose pas de méthode pour résoudre les éventuels problèmes de compatibilité pouvant survenir entre des lois différentes.

Bien que très intéressantes, ces approches sont freinées par le fait qu'elles ne peuvent être appliquées aux SI d'aujourd'hui. Ceci résulte du fait qu'elles remettent en cause les méthodes d'échanges actuelles en obligeant les entreprises à changer leur façon d'opérer. En effet, les entreprises doivent alors interfacer leur SI à la plateforme multi-agents et développer des agents capables de prendre en charge de manière autonome les tâches initialement accomplies par le SI, ce qui est très lourd à mettre en place.

Synthèse

Les technologies actuelles pour décrire les processus d'affaires des entreprises (p. ex. : BPEL [68]) ne permettent de définir que la syntaxe des échanges sans prendre en compte leur sémantique. Cette limitation est un frein à l'interopérabilité des processus car le moindre écart entre les descriptions de deux processus entraîne leur incompatibilité. L'ajout d'informations sémantiques permet de dépasser cette limite en définissant les interactions à un haut niveau d'abstraction, offrant ainsi la possibilité de mettre en relation deux services n'ayant pas exactement la même syntaxe. Les travaux que nous venons de présenter se basent donc sur une représentation sémantique des interactions entre processus d'affaires. Dans le domaine des architectures orientées services (cf. Chapitre 1), les approches se basent sur les nouvelles technologies du web sémantiques afin de proposer un système de médiation raisonnant sur la description sémantique des services. Dans le domaine de l'intelligence artificielle (cf. Chapitre 2), les différents travaux proposent de modéliser les interactions à un plus haut niveau d'abstraction en considérant le point de vue métier. En effet, dans le domaine du B2B, l'échange de documents d'affaires entraîne des engagements entre les partenaires. Ainsi ils formalisent ces engagements et définissent un cadre permettant de contrôler le déroulement des échanges en suivant l'évolution des engagements en cours.

Les projets européens *CONTRACT* et *ALIVE* allient ces deux domaines de recherche. Le projet *CONTRACT* [51] propose un environnement d'exécution des services web permettant de représenter les contrats entre les différents partenaires. Ces contrats spécifient les interdépendances entre leurs différents services en termes de modalités déontiques (obligations, interdictions et permissions), afin de permettre le contrôle au niveau business du comportement de l'ensemble des services par rapport aux spécifications des contrats. Le projet *ALIVE* [50], quant à lui, a pour but d'améliorer la flexibilité des processus en coordonnant l'exécution des services web. Pour cela un environnement multi-agents est mis en place en tant que surcouche des services web. Cet environnement est composé de trois niveaux : un niveau service, un niveau de coordination et un niveau organisationnel. Le premier niveau sert d'interface avec les services web et contient une description de l'ensemble de ces services. Le second niveau est constitué d'un ensemble d'agents, représentant les différents acteurs, ainsi qu'un ensemble de tâches pouvant être exéctuées. Enfin,

le dernier niveau, basé sur la plateforme OMNI (cf. Section 2.3.1), fournit une description à haut niveau du système en termes d'objectifs et de normes. Les agents peuvent ainsi planifier l'exécution des services web en fonction des objectifs à atteindre, des services disponibles et des normes en vigueur.

D'autres travaux tentent également d'améliorer l'interopérabilité entre les processus d'affaires des entreprises. L'association INTEROP-VLab [29], le laboratoire virtuel international pour l'interopérabilité d'entreprise créé en 2007 suite à la fin du projet INTEROP-NoE [49], a pour but d'entretenir la communauté européenne de recherche sur les problèmes d'interopérabilité d'entreprise et de structurer les travaux dans la perspective d'homogénéiser les résultats. Parmi ces travaux, certains ont pour objectif de proposer un système permettant de modéliser un processus d'affaires commun. Dans [53], les auteurs présentent une méthode permettant d'analyser l'interopérabilité entre les processus d'affaires d'entreprises en simulant des échanges dans un système multi-agents. L'approche de Shishkov et al. [58] a pour objectif de faciliter la modélisation de processus d'affaires interopérables en utilisant un système multi-agents pour analyser l'interopérabilité entre les processus au cours de leur modélisation. Ces travaux se focalisent sur des moyens d'optimiser la phase de conception dans les échanges B2B.

Les travaux actuels sur l'automatisation des échanges B2B se consacrent donc soit à faciliter la phase de conception permettant de modéliser des processus d'affaires rigides qui soient interopérables, soit à proposer de nouvelles solutions permettant de définir des SI plus flexibles et donc plus interopérables. L'objet de cette thèse est légèrement différent. Nous nous intéressons à l'interopérabilité entre les SI actuels durant la phase d'exécution. C'est-à-dire que notre intention est de pouvoir s'affranchir de la phase de conception en permettant la résolution dynamique des problèmes d'interopérabilité entre les processus d'affaires des entreprises sans impacter leurs SI. Dans ce but, nous devons donc adapter notre solution aux SI actuels. Pour cela, nous devons nous confronter à deux difficultés : la rigidité des processus d'affaires actuels et l'absence de sémantique dans leur description.

Dans la partie suivante, nous exposons l'approche que nous avons choisie afin de répondre à cette problématique. Nous décrivons ensuite les solutions que nous proposons afin de mettre en oeuvre l'approche exposée.

Deuxième partie

Contribution

Introduction

Notre approche

L'objectif de notre travail est de proposer un modèle opérationnel de gestion dynamique des exceptions dans le cadre d'échanges B2B sans accord mutuel préalable. L'expression « sans accord mutuel préalable » signifie que les partenaires n'ont pas préalablement établi de contrat d'interchange (lors de la phase de conception) et ne se sont donc pas mis d'accord sur une chorégraphie commune leur permettant d'effectuer leurs échanges. Nous devons donc concevoir un système permettant de résoudre, au cours de la phase d'exécution, les exceptions survenant lors d'échanges B2B entre partenaires ayant des processus d'affaires hétérogènes.

Avant tout, nous allons commencer par définir ce qu'est une exception. Nous distinguons trois catégories d'exceptions : (1) celles dues à un problème au niveau de l'exécution d'un processus d'affaires, (2) celles liées aux problèmes d'interopérabilité entre les SI des entreprises et enfin (3) celles liées à la transaction d'affaires sous-jacentes aux échanges. Les premières désignent les exceptions classiques survenant au cours des échanges suite au comportement anormal d'un partenaire, c'est-à-dire lorsqu'un partenaire exécute une action qui n'a initialement pas été prévue dans son processus d'affaires. Elles peuvent-être causées soit par une erreur humaine soit par une erreur lors de l'implémentation du processus d'affaires au sein du SI de l'entreprise. Les exceptions liées aux problèmes d'interopérabilités sont quant à elles dues au fait que les processus d'affaires des partenaires soient hétérogènes et donc éventuellement incompatibles. Les types d'exceptions liées à ces problèmes d'interopérabilité sont présentés à la section 1.1. Enfin, il y a les exceptions liées à la transaction d'affaires, que nous appelons exceptions *métier*. En effet, comme il a été exposé à la section 2.2, l'échange de documents d'affaires implique des engagements *métier* entre les partenaires (p. ex. : engagement de livrer un produit). Par conséquent, la satisfaction de ces engagements doit également être prise en compte pour assurer le bon déroulement des échanges. Les exceptions métier représentent donc la violation des engagements métier impliqués par les échanges de documents.

Un système de gestion des exceptions dans les échanges B2B doit gérer ces trois catégories d'exceptions. Nous allons maintenant préciser en quoi consiste la gestion d'une exception, et notamment ce que cela nécessite.

La gestion d'une exception est constituée de trois phases. Tout d'abord il y a la phase de détection. En effet, une exception doit d'abord avoir été détectée afin de pouvoir être considérée par le système lors de sa survenue. Dans les échanges B2B, la détection d'une exception nécessite de surveiller les échanges et d'être capable de les interpréter. Ensuite, une fois que l'exception a été détectée, elle doit être identifiée, c'est-à-dire que le système doit savoir ce qui a déclenché l'exception (Quelle action ? Quel acteur ?). A partir de là, on peut alors entreprendre la troisième phase qui consiste à résoudre l'exception. Le problème à ce stade est que la résolution d'une exception peut-être une tâche complexe, voire irréalisable. Il n'est pas raisonnable d'avoir un système laissant les échanges arriver dans un état d'exception pour essayer de les résoudre par la suite. Un mécanisme d'anticipation est donc indispensable afin d'éviter cette situation lorsque cela est possible.

Nous devons donc concevoir un système capable de détecter et d'identifier les exceptions au cours d'échanges B2B dans le but de les résoudre. Pour cela, nous proposons de mettre en oeuvre une plateforme de médiation servant d'intermédiaire entre les SI des partenaires B2B (cf. Figure 5). Cette plateforme doit être capable d'interpréter les interactions avec les SI et de raisonner en fonction de cette interprétation. Ce système doit également permettre l'anticipation des exceptions pouvant survenir. Afin de pouvoir anticiper le comportement des SI, la plateforme doit connaître leurs capacités d'interaction, décrites par la chorégraphie des échanges exposée par le SI. Ces capacités d'interaction correspondent à l'ensemble des échanges possibles avec le SI.

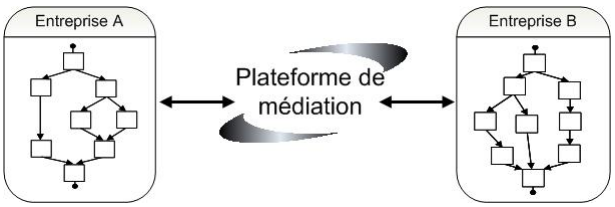


FIGURE 5 – Plateforme de médiation jouant le rôle d'intermédiaire entre les SI des entreprises

La section suivante présente l'architecture de la plateforme que nous proposons, basée sur un système multi-agent, ainsi que les bases logiques sur lesquelles nous nous appuyons afin de représenter l'information.

La plateforme de médiation

Cette section décrit les bases de la plateforme de médiation que nous avons mise en oeuvre au cours de nos travaux de recherche. Nous présentons dans un premier temps la composition de la plateforme, qui est basée sur un système multi-agents. Nous exposerons ensuite les bases de la logique que nous avons utilisée pour la représentation de l'information au sein de cette plateforme.

La composition de la plateforme

Comme nous l'avons dit précédemment, nous avons basé notre plateforme de médiation sur un système multi-agents. Comme cela a été exposé au chapître 2, un système multi-agent est un système composé de plusieurs agents. Ces agents sont des entités autonomes dotées de capacités de communication et de raisonnement leur permettant d'interagir entre elles afin d'évoluer au sein d'un environnement commun dans le but de satisfaire leurs objectifs. Le parallèle avec les échanges B2B est évident. En effet, un échange B2B consiste en un ensemble de partenaires communiquant entre eux dans le but de réaliser une transaction commerciale.

Les méthodes actuelles permettant l'automatisation des échanges, consistent en l'exécution, par l'ensemble des partenaires, d'une chorégraphie commune préalablement définie. Le comportement de l'ensemble des partenaires est alors contraint par cette chorégraphie. Notre objectif est de permettre l'automatisation des échanges sans avoir besoin de définir au préalable une chorégraphie commune. Dans ce cas, il n'y aura plus de contraintes prédéfinies sur le comportement des partenaires, leur laissant ainsi une entière autonomie. C'est justement cette autonomie qui ouvre la porte aux problèmes d'interopérabilité car elle peut mener à des comportements incompatibles. Et c'est également la raison pour laquelle nous avons orienté notre réflexion vers le domaine des systèmes multi-agents. La modélisation de notre plateforme par un système multi-agent (cf. Figure 6) permet dès lors de représenter chaque partenaire comme une entité autonome (un agent) au sein de la plateforme. Un agent supplémentaire, appelé agent « médiateur », servira d'intermédiaire entre ces agents représentant les partenaires, appelés agents « entreprise ». L'idée ici est que chaque agent « entreprise » joue le rôle de l'entreprise qu'il représente en transmettant les documents entre la plateforme et le SI de l'entreprise. Chaque document envoyé par l'entreprise est alors transféré par l'agent « entreprise » à l'agent « médiateur » et chaque document envoyé par l'agent « médiateur » à un agent « entreprise » est transféré au SI de l'entreprise que ce dernier représente (que nous désignerons désormais par « son » entreprise). L'interface entre l'entreprise et l'agent « entreprise », symbolisée par le rectangle accolé à l'agent « entreprise » sur la figure, représente l'interface technique entre le SI de

l'entreprise et la plateforme permettant à l'agent « entreprise » de communiquer avec le SI de son entreprise. Nous discutons de cette interface à la section 8.3.2.

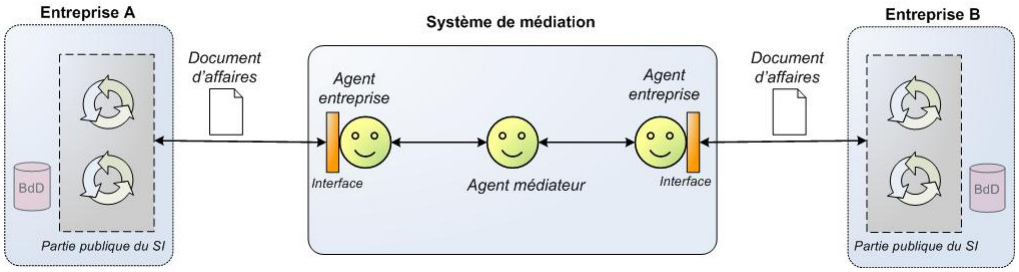


FIGURE 6 – Plateforme de médiation multi-agents

La logique

Nous avons utilisé la logique FIPA-SL [26] pour représenter les informations au sein de notre plateforme multi-agents. FIPA-SL est un langage logique initialement utilisé pour formaliser la sémantique du langage FIPA-ACL [25], en définissant les comportements d'un agent en termes de ses attitudes mentales et des actions qu'il peut observer ou réaliser. Il couvre la logique des prédicats du premier ordre et sa syntaxe est préfixée.

La logique FIPA-SL permet notamment de modéliser les notions de croyance et d'intention (comparables à d'autres présentes dans des approches bien connues de type « BDI » [52]). L'opérateur modal B formalise les croyances de l'agent. $(B\ i\ p)$ signifie que l'agent i croit que la propriété p est vraie. L'opérateur modal I formalise quant à lui les intentions d'un agent. $(I\ i\ p)$ signifie que l'agent i a l'intention que la propriété p devienne vraie. FIPA-SL définit également l'opérateur *done*, qui permet de décrire l'occurrence des actions. $(done\ a\ p)$ signifie que l'action a a été réalisée avant quoi la propriété p était vraie. Ces opérateurs s'interprètent dans une structure de mondes possibles de Kripke.

Une action FIPA-SL est décrite par l'expression $(action\ acteur\ acte)$ où *acteur* représente l'agent qui a effectué l'action et *act* est un terme désignant l'instance de l'action effectuée. Par exemple, l'expression ci-dessous signifie que l'agent i a informé l'agent j que la formule "*sunny*" est vraie.

$$(done\ (action\ i\ (inform\ j\ "sunny"))\ true)$$

Il existe deux pincipaux types d'expressions en FIPA-SL : les termes et les formules. Les termes représentent les objets du domaine (instances, actions ...). Les formules, quant à elles, sont des expressions booléennes qui représentent les faits (vrais ou faux).

À partir de cette base logique, nous présentons dans la section suivante la représentation de l'information au sein de notre plateforme de médiation.

Dans cette partie, nous présentons l'ensemble de nos contributions permettant la conception de notre plateforme de médiation. Nous décrivons tout d'abord le modèle que nous avons défini pour représenter les actions d'interactions. Ce modèle nous fournit les bases pour spécifier un comportement dynamique permettant de contrôler les échanges et de détecter la survenue d'exceptions. Ce comportement dynamique ne permettant pas de prévoir le comportement des partenaires et donc d'éventuelles exceptions futures, nous proposons ensuite un mécanisme, basé sur la théorie des jeux, utilisé comme moyen d'anticipation des exceptions. Enfin, à partir de ce mécanisme, nous définissons la notion d'interopérabilité de médiation.

3

Interprétation des interactions

Dans la section précédente, nous avons décrit la structure de notre plateforme qui est un système multi-agents composé d'agents interagissant entre eux en s'envoyant des documents d'affaires. Nous nous intéressons ici à l'interprétation de ces interactions, représentant les interactions entre les systèmes d'information des entreprises. Pour cela, nous devons représenter la sémantique impliquée par ces interactions, c'est-à-dire par l'envoi des documents d'affaires. Avant de présenter la modélisation de cette sémantique, nous montrons l'intérêt de mettre en place un contexte institutionnel au sein de notre plateforme. Ensuite nous décrivons comment les échanges de documents d'affaires sont définis dans ce contexte institutionnel. En particulier, afin de prendre en compte les trois types d'exceptions présentés précédemment, nous proposons de modéliser à la fois la sémantique métier impliquée par les documents d'affaires et la sémantique liée à la rigidité des chorégraphies définies par les entreprises.

3.1 Contexte institutionnel

Notre objectif est de modéliser les échanges de documents d'affaires au sein de notre plateforme afin de pouvoir les interpréter dynamiquement. Le but est de contrôler l'état courant de l'échange afin d'assurer son bon déroulement. Cet état dépend des documents échangés. En effet, les documents ont une certaine sémantique et leur interprétation génère des effets. L'état d'un échange entre des partenaires est donc caractérisé par l'ensemble des effets impliqués par les documents qu'ils se sont échangés. Nous devons alors représenter les effets que provoque l'envoi d'un document d'affaires sur l'état de l'échange en cours.

Dans l'approche classique, ces effets sont décrits dans le contrat d'interchange, préalablement

créé par les partenaires lors de la phase de conception. Ainsi, les partenaires interprètent l'envoi d'un document tel que cela est défini dans le contrat d'interchange. Il en résulte une interprétation uniforme de la part de l'ensemble des partenaires.

Dans notre cas, les échanges se font sans accord mutuel préalable. Les partenaires ne se sont donc pas préalablement mis d'accord sur une définition commune des effets engendrés par l'envoi des documents d'affaires. Ces échanges seront donc interprétés par chaque entreprise selon son propre point de vue. Un même document échangé peut alors être interprété de manières différentes par chaque partenaire. Il en résulte un état spécifique pour chacun d'entre eux, ce qui risque de mener à une incohérence entre leurs différents états pouvant être à l'origine d'une exception. Il est donc nécessaire de prendre en compte le point de vue de chaque partenaire pour contrôler les échanges.

Nous proposons donc de modéliser les échanges entre les agents de la plateforme dans un cadre institutionnel. Cette approche offre aux agents un cadre formel et explicite permettant de définir des règles institutionnelles décrivant l'évolution de l'environnement en fonction de leur comportement. Nous nous basons ici sur le concept de fait institutionnel. Un fait institutionnel, par opposition à un fait brut, désigne une propriété qui est considérée vraie dans le contexte d'une certaine institution, c'est-à-dire, par rapport aux normes en vigueur dans cette institution. Un exemple classique pour illustrer ce concept est l'exemple du mariage. En effet, deux personnes mariées en France, sont considérées mariées dans la contexte de la constitution française. Dans un autre pays, ayant une constitution différente, leur mariage peut ne pas être reconnu.

Notre approche consiste alors à représenter les interactions entre un agent « entreprise » et l'agent « médiateur » dans le contexte d'une institution spécifique représentant l'entreprise relative à l'agent. L'état de chaque institution spécifique sera alors représenté par un ensemble de faits institutionnels relatifs à l'institution. Ceci permet d'interpréter l'échange d'un document d'affaires avec le système d'information d'une entreprise dans le contexte d'une institution spécifique, représentant le point de vue de l'entreprise.

Demolombe et Louis [17] ont proposé un langage logique permettant notamment de formaliser les notions liées à l'environnement institutionnel dans lequel évolue un agent. L'opérateur D formalise les faits institutionnels. $D_{inst}p$ signifie que p est reconnu vrai pour l'institution $inst$. Ces faits sont donc valables uniquement dans le contexte de l'institution. Les conséquences normatives sont représentées par l'opérateur $\Rightarrow_{inst}$. $p \Rightarrow_{inst} q$ signifie que par rapport aux normes en vigueur dans l'institution $inst$, $p \Rightarrow q$. Si p est vrai, q sera alors reconnu comme un fait institutionnel dans le contexte de l'institution $inst$. Cet opérateur est plus connu sous le nom de *count-as*. Il a été formalisé par Sergot et Jones [37]. Enfin, l'opérateur *power* formalise les pouvoirs institutionnels. $power(i,\ inst,\ cond,\ proc,\ n)$ signifie que l'agent i a le pouvoir, dans le

contexte de l'institution *inst*, de faire en sorte que n soit vrai en réalisant *proc* et si les conditions *cond* sont satisfaites. C'est une abbréviation de la formule $(cond \wedge done(i,\ proc,\ \top)) \Rightarrow_{inst} n$. Demolombe et Louis ont également défini l'opérateur déontique O afin de modéliser la notion d'obligation d'être impersonnelle. Op signifie qu'il est obligatoire que p soit vrai. Il permet également de modéliser les notions d'interdiction ($O(\neg p)$) et de permission ($\neg O(\neg p)$).

A partir de ces concepts institutionnels, nous proposons de modéliser les interactions correspondant aux échanges de documents d'affaires spécifiés dans la chorégraphie du système d'information d'une entreprise. Les caractéristiques de ces interactions doivent être définies dans le contexte de l'institution relative à l'entreprise. Pour cela, chaque interaction est modélisée par une action ayant une dimension institutionnelle, appelée action institutionnelle.

Le concept d'action institutionnelle, présenté dans [71], désigne une action dont l'observation et l'interprétation sont relatives au fonctionnement d'une (ou plusieurs) institution(s). Une action institutionnelle est caractérisée par sa précondition institutionnelle et son effet institutionnel. La précondition institutionnelle décrit les propriétés devant être satisfaites pour que l'exécution de l'action soit permise dans le contexte de l'institution. Une telle précondition est à différencier de la précondition habituelle de faisabilité d'une action. En effet, rien n'empêche un agent d'exécuter l'action institutionnelle même si sa précondition institutionnelle n'est pas satisfaite. Cependant, dans ce cas, l'agent violerait les règles de l'institution et l'effet institutionnel de l'action ne serait pas reconnu. De plus, l'agent pourrait alors s'exposer à des sanctions. L'effet institutionnel décrit les effets de l'action dans le contexte d'une certaine institution. En particulier, il permet de définir les faits institutionnels engendrés par l'exécution de l'action (si sa précondition institutionnelle est satisfaite). Par exemple, dans le cas du mariage, l'action de dire « je vous déclare mari et femme » est une action institutionnelle ayant la précondition et l'effet institutionnels suivant :

- précondition institutionnelle :
  - l'auteur de l'action est titulaire de la fonction de maire ;
  - les deux personnes concernées ont manisfesté leur accord ;
  - les deux personnes satisfont les conditions requises pour se marier (célibataires, de sexes opposés, sans lien de parenté...).
- effet institutionnel : les deux personnes sont reconnues en tant que mari et femme dans le cadre de la constitution française.

Une même action physique peut donner lieu à des interprétations différentes dans des institutions différentes. Dans notre contexte applicatif, ce concept d'action institutionnelle permet de modéliser le fait que l'envoi d'un document d'affaires est interprété différemment par des entreprises différentes, c'est-à-dire au niveau de leur sous-institution correspondante.

Formellement, dans le cadre de la logique proposée par Demolombe et Louis, une action institutionnelle est modélisée par la notion de pouvoir institutionnel. Une action *act* ayant pour précondition institutionnelle *cond* et pour effet institutionnel E dans le contexte de l'institution *inst* est modélisée par la formule $power(i,\ inst,\ cond,\ act,\ E)$. Ici, i désigne l'acteur de l'action. D'après la logique proposée par Demolombe et Louis, nous obtiendrons donc $D_{inst}E$ à partir de l'observation de $done(i,\ act,\ \top) \wedge cond$. De manière informelle, cela signifie que lorsqu'un agent i exécute l'action *act*, si la précondition institutionnelle *cond* est satisfaite, l'effet institutionnel E sera interprété en tant que fait institutionnel dans le contexte de l"institution *inst*.

Carole Adam et al. [72] ont étendu le langage logique FIPA-SL, présenté dans la section précédente, afin de prendre en compte ces opérateurs déontiques et institutionnels. Cette extension est appelée le langage xSL. Ce langage nous permet donc de modéliser les actions institutionnelles représentant les échanges de documents d'affaires entre les entreprises. L'expression xSL formalisant une action institutionnelle est la suivante :

(*power Acteur Institution Condition Action Effet*)

La figure 3.1 illustre la nouvelle architecture de la plateforme, intégrant l'aspect institutionnel. Elle est composée d'une institution globale représentant l'état globale de la plateforme. Cette institution est elle-même composée d'institutions, que nous appelons *sous-institutions*. Une *sous-institution* est relative à un agent « entreprise ». Elle représente l'état des échanges selon le point de vue de son entreprise.

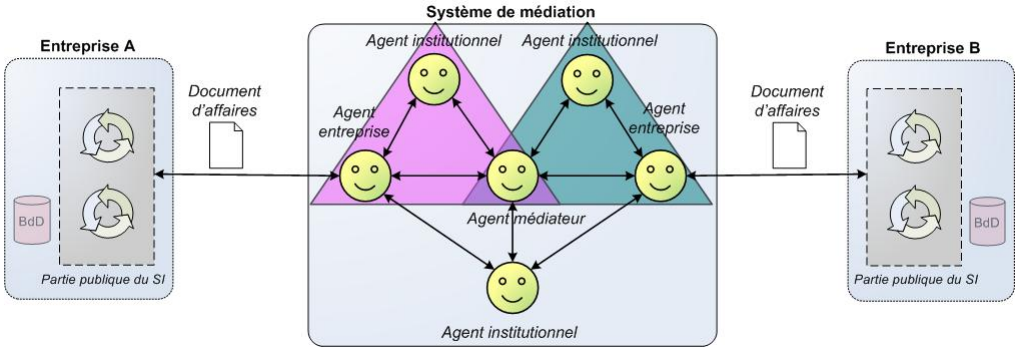


FIGURE 3.1 – Architecture institutionnelle de la plateforme de médiation

Chaque institution est représentée par un agent institutionnel qui est le dépositaire des règles relatives à son institution. Son rôle est d'observer toutes les actions effectuées au sein de la plateforme afin d'inférer les faits institutionnels à partir de ces règles. L'état d'une institution est déterminé par la base de croyance de l'agent institutionnel la représentant.

Les agents « entreprise » appartiennent à exactement deux institutions : l'institution globale et l'institution relative à leur entreprise. L'agent « médiateur » appartient à toutes les institutions, c'est-à-dire qu'il a la connaissance des règles institutionnelles de toutes les institutions afin d'interpréter ses interactions avec l'ensemble des agents « entreprise » selon le point de vue de leur entreprise.

La structure institutionnelle que nous venons de présenter offre un cadre permettant de définir la sémantique des interactions du point de vue de chaque entreprise. Cette distinction entre les différents points de vue des partenaires est essentielle pour déterminer les éventuels problèmes d'interopérabilité entre leurs systèmes d'informations. Dans la suite, nous présentons comment modéliser les informations impliquées par les documents d'affaires échangés.

3.2 La sémantique des documents d'affaires

Au cours d'échanges B2B, les interactions, consistant en l'échange de documents d'affaires, engendrent des effets sur l'état de l'échange. Ces effets dépendent de la sémantique des documents échangés. Dans la section précédente, nous avons proposé de définir ces interactions dans un contexte institutionnel propre à chaque entreprise, en les modélisant par des actions institutionnelles. Ces actions institutionnelles offre un cadre pour définir la sémantique impliquée par ces documents selon le point de vue de chaque entreprise (représenté par la sous-institution correspondante). Nous décrivons ici la modélisation que nous proposons pour représenter cette sémantique.

Nous considérons deux types d'informations : les informations métiers et les informations liées à la chorégraphie. Dans la suite nous présentons ces deux types d'informations ainsi que leur formalisation logique. A l'aide de cette formalisation, nous montrons ensuite comment représenter chaque envoi d'un document d'affaires, spécifié dans une chorégraphie du système d'information d'une entreprise, par une action institutionnelle spécifique.

3.2.1 Les engagements métier

Comme nous l'avons vu dans la section 2, la notion d'engagement est une notion primordiale dans les échanges B2B. En effet, lorsque des partenaires s'échangent des documents d'affaires, il en découle généralement des engagements au niveau métier. Typiquement, lors d'échanges B2B entre un client et un fournisseur, l'envoi d'un bon de commande de la part du client l'engage envers le fournisseur à payer pour les produits qu'il a commandé. L'opérateur déontique O, dans

la logique de Demolombe et Louis [17], permet de représenter la notion d'obligation. Ainsi, pour reprendre notre exemple, la formule $O(done(i,\ paiement,\ \top))$ traduit le fait qu'il est obligatoire que l'agent i ait exécuté l'action *paiement*. Le principal problème de cette représentation est qu'elle n'exprime pas explicitement la notion de responsabilité. En effet, il n'est pas indiqué dans la formule quel agent est responsable en cas de non respect de l'obligation (c'est-à-dire qui est obligé). Ainsi, en cas de non respect de cette obligation, on ne peut que se retourner vers l'acteur de l'action. De plus, il n'est pas précisé non plus quel est le bénéficiaire de l'obligation, c'est-à-dire envers qui l'agent i est engagé à effectuer le paiement.

Dans le domaine du B2B, il est important de représenter cette notion de responsabilité. En effet, il est possible au cours d'échanges, de déléguer une action à un tiers. Dans le cas de la livraison de produits par exemple, un fournisseur peut s'engager envers un client sur le fait qu'un livreur livre les produits. Dans ce cas, le fournisseur prend la responsabilité de la livraison, bien que cette action doit être exécutée par le livreur. Il est également nécessaire de représenter envers qui un agent est responsable afin qu'en cas de non respect de l'engagement, le préjudice soit reconnu pour le bénéficiaire de l'engagement, et qu'il puisse ainsi par la suite en obtenir réparation.

Pour ces raisons, nous avons donc choisi d'utiliser la notion d'engagement social (cf. Section 2.2) pour représenter les engagements métier. En effet, un engagement social , en plus d'être caractérisé par l'objet de l'engagement et une éventuelle condition, définit deux agents, le débiteur et le créditeur, représentant respectivement l'agent engagé et celui envers qui il est engagé.

Différentes modélisations ont été proposées dans la littérature. Dans la section 2.2, nous avons détaillé le modèle d'engagement social proposé par Singh [60] afin d'exposer l'utilisation de ce modèle pour représenter les processus d'affaires des entreprises en termes de manipulations d'engagements métier. Pour notre représentation, nous nous sommes également inspirés des travaux de Fornara et Colombetti [33]. Leur objectif est de modéliser les communications entre agents par des engagements. Pour cela, ils définissent un engagement comme un objet caractérisé par cinq attributs : un identifiant (unique), le débiteur (l'agent qui est engagé), le créditeur (l'agent envers qui le créditeur est engagé), l'objet de l'engagement (sur quoi est engagé le débiteur), la condition de l'engagement (condition devant être satisfaite afin de rendre l'engagement actif) et enfin l'état de l'engagement (cf. ci-dessous).

Une particularité de leur modèle est de représenter de manière explicite l'état d'un engagement. Ainsi un engagement peut se retrouver dans un des cinq états suivants : *unset* (état abstrait utilisé pour représenter une proposition d'engagement), *pending* (ou *en attente*), *active* (ou $actif$), $fulfilled$ (ou $satisfait$), *violated* (ou *violé*), *cancelled* (ou *annulé*). L'état *pending* représente un engagement dont ni la condition, ni l'objet ne sont satisfaits. L'état *active* re-

présente un engagement dont la condition est satisfaite mais dont l'objet ne l'est pas encore. L'état $fulfilled$ représente un engagement dont l'objet est satisfait. L'état *violated* représente un engagement ($actif$) dont l'objet ne peut pas être satisfait. Enfin l'état *cancelled* représente un engagement dont la condition ne peut pas être satisfaite (ce qui annule alors l'engagement).

Ils définissent également deux opérations : l'opération de création d'un engagement ($mc()$) et l'opération de changement d'état ($sc()$). Ces opérations permettent de manipuler les engagements de manière explicite.

Ils ont ensuite défini le cycle de vie d'un engagement (cf. Figure 3.2), en indiquant les règles de transition entre les états. Ces règles sont présentées dans le tableau 3.1, où a, b, P et Q représentent respectivement le débiteur, le créditeur, l'objet et la condition de l'engagement (la condition est écrite T lorsqu'elle est vraie).

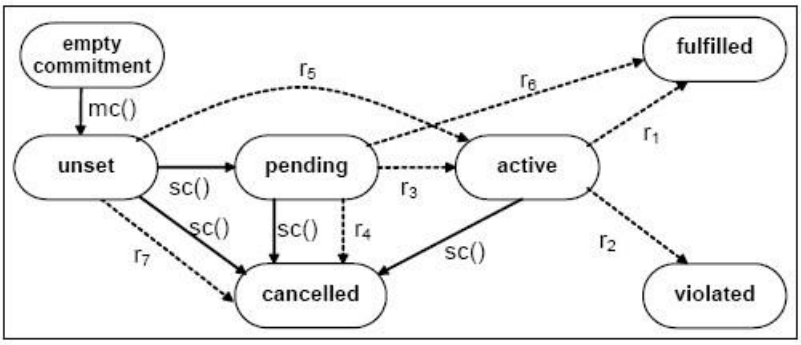


FIGURE 3.2 – Cycle de vie d'un engagement

event	action	rule
$P.truth_value() = 1$	$C_i(a,a,b,P\|T) \rightarrow C_i(f,a,b,P\|T)$	1
$P.truth_value() = 0$	$C_i(a,a,b,P\|T) \rightarrow C_i(v,a,b,P\|T)$	2
$Q.truth_value() = 1$	$C_i(p,a,b,P\|Q) \rightarrow C_i(a,a,b,P\|T)$	3
$Q.truth_value() = 0$	$C_i(p,a,b,P\|Q) \rightarrow C_i(c,a,b,P\|Q)$	4
$P.truth_value() = 1$	$C_i(u,a,b,P\|T) \rightarrow C_i(a,a,b,P\|T)$	5
$P.truth_value() = 1$	$C_i(p,a,b,P\|Q) \rightarrow C_i(f,a,b,P\|Q)$	6
$curr_time > t$	$C_i(u,a,b,P\|Q,t) \rightarrow C_i(c,a,b,P\|Q)$	7
legend: $u = unset$, $p = pending$, $c = cancelled$, $a = active$, $f = fulfilled$, $v = violated$		

TABLE 3.1 – Règles de transition d'état d'un engagement

Note : la règle 7 fait référence à l'attribut t d'un engagement. Cet attribut représente la date limite avant laquelle une proposition d'engagement (état *unset*) doit avoir été acceptée ou annulée. Au delà de cette date, l'engagement est automatiquement annulé.

La représentation explicite de l'état d'un engagement permet de garder une trace de son évolution. De plus, la définition des règles de transition entre états permet la mise à jour automatique de l'état courant d'un engagement. Ceci a un double intérêt : (a) les agents peuvent raisonner sur les engagements directement en fonction de leur état courant, (b) la violation d'un engagement est détectée automatiquement (par la transition de l'état courant vers l'état *violated*).

Notre objectif ici est de modéliser les engagements métier impliqués par l'envoi de documents d'affaires, dans le but de détecter la survenue éventuelle d'une exception métier au cours d'échanges entre des partenaires. C'est pourquoi nous avons choisi d'utiliser un modèle d'engagement social représentant explicitement l'état d'un engagement. Ainsi la survenue d'une exception pourra être détectée de maniére automatique par la violation d'un engagement (le changement de son état).

Nous définissons donc un engagement par un quintuplet (*state*, *debtor*, *creditor*, *content*, *condition*) où *state* désigne l'état courant de l'engagement, *debtor* et *creditor* représentent respectivement le débiteur et le créditeur de l'engagement, et *content* et *condition* repésentent respectivement l'objet et la condition de l'engagement. Nous avons étendu la logique xSL avec ce concept d'engagement. Nous avons nommé cette logique xSL^{com}. Ainsi dans la logique xSL^{com}, un engagement est formalisé par l'expression (*commitment state debtor creditor content condition*) où *commitment* est une constante permettant d'identifier l'expression comme étant un engagement, *state*, *debtor* et *creditor* sont des termes FIPA-SL et *content* et *condition* sont des formules FIPA-SL (cf. section II pour la définition d'un terme et d'une formule FIPA-SL).

Concernant les états dans lesquels peut être un engagement, nous reprenons les états définis dans [33] excepté l'état abstrait *unset*. Un engagement peut donc être dans l'état *pending*, *active*, $fulfilled$, *violated* ou *cancelled*. Nous ne considérons pas l'état *unset* car il représente un état intermédiaire précédant l'existence réelle d'un engagement qui nécessite alors une action explicite de changement d'état pour devenir concret. Or, notre objectif ici est de représenter les engagements courants sous-jacents aux échanges. Une fois les engagements créés, leur état évolue automatiquement en fonction des valeurs de leur objet et de leur condition. Ceci permet ainsi de suivre l'évolution de ces engagements de manière dynamique.

Pour cela, nous avons formalisé les règles de transition automatique entre états (règles 1, 2, 3, 4 et 6 présentées dans le tableau 3.1). La règle 3 définit la transition entre les deux états intermédiaires (de l'état *pending* à l'état *active*). Les règles 1, 2, 4 et 6 définissent quant à elles les transitions d'un état intermédiaire vers un état final (états $fulfilled$, *violated* ou *cancelled*).

Règle 1 : $(commitment\ active\ debtor\ creditor\ content\ condition) \wedge content$
$\rightarrow (commitment\ fulfilled\ debtor\ creditor\ content\ condition)$

Règle 2 : $(commitment\ active\ debtor\ creditor\ content\ condition) \wedge (not\ content)$
$\rightarrow (commitment\ violated\ debtor\ creditor\ content\ condition)$

Règle 3 : $(commitment\ pending\ debtor\ creditor\ content\ condition) \wedge condition$
$\rightarrow (commitment\ active\ debtor\ creditor\ content\ condition)$

Règle 4 : $(commitment\ pending\ debtor\ creditor\ content\ condition) \wedge (not\ condition)$
$\rightarrow (commitment\ cancelled\ debtor\ creditor\ content\ condition)$

Règle 6 : $(commitment\ pending\ debtor\ creditor\ content\ condition) \wedge content$
$\rightarrow (commitment\ fulfilled\ debtor\ creditor\ content\ condition)$

L'intégration de ces règles au sein de la plateforme permet alors la mise à jour automatique et dynamique de l'état des engagements courant. La survenue d'une exception sera donc détectée automatiquement.

À partir de cette formalisation d'un engagement social, nous pouvons donc maintenant représenter un engagement métier dans la logique xSL^{com}. Ainsi l'engagement d'une entreprise *Client* à payer les produits (formule *PayProducts*) à l'entreprise *Fournisseur* si cette dernière les a livrés (formule *DeliverProducts*) se traduit par l'expression xSL^{com} suivante :

$$(commitment\ \ pending\ \ Client\ \ Fournisseur\ \ PayProducts\ \ DeliverProducts)$$

Dans la suite, l'ensemble des expressions logiques présentées est exprimé dans la logique xSL^{com} qui est donc la logique xSL étendue avec le concept d'engagement social présenté ci-dessus.

3.2.2 Les informations liées à la chorégraphie

Pour contrôler les échanges entre les systèmes d'information des entreprises, notre plateforme de médiation doit avoir la connaissance des contraintes liées à leurs chorégraphies. Nous devons donc représenter ces contraintes pour qu'elles soient interprétées au cours des échanges afin (1) de

détecter les éventuelles exceptions pouvant provenir d'un comportement anormal des systèmes d'information et (2) d'avoir un comportement conforme aux spécifications de leur chorégraphie respective et ainsi ne pas provoquer d'exceptions.

Nous avons identifié deux types de contraintes : les documents requis et les documents non-attendus. Les documents requis sont les documents dont l'envoi est nécessaire à la poursuite des échanges. Ne pas envoyer un tel document provoque une exception bloquant le processus d'affaires de l'entreprise. Les documents non-attendus quant à eux sont des documents qui, n'étant pas prévus dans la chorégraphie, provoque une exception s'ils sont envoyés au système d'information de l'entreprise.

Dans la suite, l'envoi d'un document *Doc* par un agent $Agt1$ à un agent $Agt2$ est modélisé par le terme xSL$^{\text{com}}$ suivant :

$$(send\ Agt1\ Agt2\ Doc)$$

Documents requis : Comme nous venons de l'exposer, ne pas envoyer un document requis provoque une exception. Afin d'éviter cette situation, nous devons représenter le fait que l'envoi d'un document soit obligatoire. De façon similaire aux engagements métiers, nous proposons de modéliser cette obligation avec le concept d'engagement social. Ainsi, l'envoi obligatoire d'un document *Doc* d'un agent Agt_1 à un agent Agt_2 est formalisé par l'expression xSL$^{\text{com}}$ suivante :

$$\begin{array}{l}(commitment\ active \\ \qquad\qquad Agt_1 \\ \qquad\qquad Agt_2 \\ \qquad\qquad (done\ (action\ Agt_1\ (send\ Agt_1\ Agt_2\ Doc))\ true) \\ \qquad\qquad true)\end{array}$$

L'expression ci-dessus signifie bien que l'agent Agt_1 doit envoyé le document *Doc* à l'agent Agt_2. Cependant, au cours d'un échange, lorsqu'un document est requis à un moment donné, il doit être le prochain message échangé. Or, si Agt_1 envoie d'autres documents puis envoie *Doc*, l'engagement ci-dessus sera satisfait bien que Agt_1 n'est pas respecté la contrainte. Nous devons donc également exprimer le fait qu'un document soit requis à la prochaine étape de l'échange entre les deux agents. Pour cela, nous avons introduit une nouvelle modalité au langage xSL$^{\text{com}}$, inspirée de l'opérateur modal *next* dans la logique temporelle LTL [46]. L'opérateur X_{inst} permet

de représenter l'étape suivante d'un échange. Ainsi la formule $(X_{inst}\ (done\ act\ true))$ est vraie si la prochaine action effectuée dans le contexte de l'institution *inst* est l'action *act*. Si c'est une autre action, alors la formule sera fausse.

L'envoi obligatoire d'un document *Doc* d'un agent Agt_1 à un agent Agt_2 à l'étape suivante d'un échange dans le cadre de l'institution *inst* est donc formalisé par l'expression xSL^{com} suivante :

$$
\begin{array}{l}
(commitment\ active \\
\qquad Agt_1 \\
\qquad Agt_2 \\
\qquad (X_{inst}\ (done\ (action\ Agt_1\ (send\ Agt_1\ Agt_2\ Doc))\ true)) \\
\qquad true)
\end{array}
$$

Dans certains cas, il se peut que l'envoi d'un document parmi plusieurs possibles soit requis. Par exemple, suite à la réception d'un devis, si une entreprise cliente représentée par l'agent Agt_C doit envoyer à son fournisseur, représenté par l'agent Agt_F, soit une confirmation de sa commande ($confirmation$), soit une annulation ($cancellation$), dans le contexte de l'institution *inst*, alors l'objet de l'engagement sera une disjonction entre ces deux actions. On obtient alors l'expression xSL^{com} suivante :

$$
\begin{array}{l}
(commitment\ active \\
\qquad Agt_C \\
\qquad Agt_F \\
\qquad (or\ (X_{inst}\ (done\ (action\ Agt_C\ (send\ Agt_C\ Agt_F\ confirmation))\ true)) \\
\qquad\quad (X_{inst}\ (done\ (action\ Agt_C\ (send\ Agt_C\ Agt_F\ cancellation))\ true))) \\
\qquad true)
\end{array}
$$

Documents non-attendus : Les documents non-attendus désignent tous les documents que le système d'information de l'entreprise ne peut pas recevoir à une étape donnée d'un échange B2B. L'ensemble de ces documents ne peut être défini explicitement. Nous proposons donc de modéliser l'ensemble des documents attendus.

Les documents attendus sont les documents dont l'envoi est permis à une étape donnée d'un

échange B2B. Ils représentent à la fois les documents requis et les documents optionnels. En logique, le concept de permission s'exprime classiquement par la négation de l'interdiction (cf. Section 3.1). La permission qu'un agent Agt_1 envoie un document Doc à un agent Agt_2, à l'étape suivante d'un échange, dans le contexte de l'institution $inst$, est donc formalisée par l'expression xSL^{com} suivante (on parle également d'action autorisée) :

$$(not\ (O\ (not\ (X_{inst}\ (done\ (action\ Agt_1\ (send\ Agt_1\ Agt_2\ Doc))\ true)))))$$

La représentations du monde d'un agent désigne les connaissances de l'agent sur son environnement. Ce sont des connaissances partielles. L'interprétation de ces connaissances respecte donc l'hypothèse du monde ouvert. L'hypothèse du monde ouvert signifie que les informations présentes (positives ou négatives) décrivent les connaissances de l'agent, et les informations absentes représentent l'ignorance de l'agent. Ainsi, les documents attendus selon un agent sont les documents dont l'envoi est explicitement permis dans sa base de connaissance. À partir de là, nous considérons alors tous les autres documents comme des documents non-attendus. L'ensemble des documents non-attendus correspond donc à l'ensemble des documents dont l'envoi n'est pas explicitement permis.

Afin de contrôler l'exécution des documents non-attendus, nous proposons de définir un engagement social global (c'est-à-dire valable pour tous les échanges). Cet engagement est formalisé par l'expression xSL^{com} suivante :

$$\begin{array}{l}(commitment\ active\\ \qquad ??Agt_1\\ \qquad ??Agt_2\\ \qquad (or\ (not\ (X_{inst}\ (done\ (action\ ??Agt_1\ (send\ ??Agt_1\ ??Agt_2\ Doc))\ true))))\\ \qquad\quad (not\ (O\ (not\ (X_{inst}\ (done\ (action\ ??Agt_1\ (send\ ??Agt_1\ ??Agt_2\ Doc))\ true)))))\\ \qquad true)\end{array}$$

Ici $??Agt_1$ et $??Agt_2$ sont des variables représentant n'importe quel agent. Ainsi l'expression ci-dessus signifie que tout agent Agt_1 est engagé envers tout agent Agt_2 sur le fait suivant : soit l'action est autorisée à l'étape suivante de l'échange, soit elle n'est pas exécutée. Intuitivement, cela veut dire que tous les agents sont engagés à ne pas exécuter une action si elle n'est pas autorisée. Ainsi, si un agent exécute une action non-autorisée, cet engagement sera violé et l'exception sera donc détectée.

Nous venons de présenter la formalisation des informations nous permettant de définir la sémantique des échanges B2B au sein de notre plateforme. La section suivante précise comment définir cette sémantique à partir des spécifications d'une chorégraphie.

3.3 Modélisation d'une chorégraphie

Dans cette section, nous allons décrire comment modéliser les interactions entre les systèmes d'informations des entreprises et notre plateforme. Ces interactions sont représentées au sein de notre plateforme par des actions entre les agents « entreprise » et l'agent « médiateur ». Comme nous en avons discuté à la section 3.1, ces actions sont modélisées par des actions institutionnelles. Chaque action institutionnelle correspond à l'échange d'un document d'affaires entre un agent « entreprise » et l'agent « médiateur », interprété dans le contexte de l'entreprise de l'agent « entreprise ». Sa précondition et son effet institutionnels sont définis à partir de la représentation des informations présentée précédemment, selon le point de vue de l'entreprise.

Comme nous l'avons vu, il y a deux types d'informations à prendre en compte : les informations métier et les informations liées à la chorégraphie. Les informations métier décrivent les engagements métier engendrés par l'envoi de certains documents d'affaires. Ce sont des informations abstraites non définies dans le processus d'affaires de l'entreprise. Ces informations doivent donc être définies directement par l'entreprise. Ainsi pour chaque échange d'un document d'affaires, c'est-à-dire pour chaque action institutionnelle, les engagements métier engendrés par ce document (le cas échéant) doivent être déterminés. Ces engagements seront alors modélisés dans l'effet institutionnel de l'action. Les informations liées à la chorégraphie, quant à elles, représentent les capacités d'interaction du système d'information de l'entreprise, déterminées par les spécifications de la chorégraphies. Elles s'obtiennent donc directement à partir de la chorégraphie. Nous allons maintenant formaliser la définition d'une action institutionnelle correspondant à l'échange d'un document défini dans une chorégraphie.

Une action institutionnelle est modélisée par un pouvoir institutionnel, décrit par l'expression xSL^{com} suivante (cf. Section 3.1) :

(*power Acteur Institution Condition Action Effet*)

Soit Z une entreprise, I_z la sous-institution relative à Z, Agt_z l'agent « entreprise » correspondant à Z et Agt_M l'agent « médiateur ». Une action institutionnelle représentant l'envoi d'un document d'affaires *Doc* par le système d'information de Z est modélisée au sein de la

plateforme par une expression de la forme suivante :

$$(power\ Agt_z\ I_z\ Condition\ (send\ Agt_z\ Agt_M\ Doc)\ Effet)$$

De manière symétrique, l'envoi d'un document *Doc* au système d'information de Z est modélisé par une expression de la forme suivante :

$$(power\ Agt_M\ I_z\ Condition\ (send\ Agt_M\ Agt_Z\ Doc)\ Effet)$$

On suppose ici que chaque document représenté dans la formalisation d'une action est un modèle de document conforme au standard utilisé par le système d'information de l'entreprise. Dans le cadre de nos travaux, nous faisons l'hypothèse que l'ensemble des partenaires utilisent le même standard afin de se concentrer uniquement sur les problèmes d'interopérabilité au niveau des processus. Nous discutons plus en détail du modèle d'un document d'affaires dans le section 3.4.

La précondition institutionnelle d'une action modélisant l'envoi d'un document d'affaires détermine les situations dans lesquelles l'envoi de ce document n'engendrera pas d'exception. Il y a deux cas à distinguer : (a) l'acteur de l'action est l'agent « entreprise », (b) l'acteur de l'action est l'agent « médiateur ».

(a) Le rôle de l'agent « entreprise » est de faire l'interface entre la plateforme et le système d'information de son entreprise. Ainsi l'envoi d'un document d'affaires par l'entreprise déclenche automatiquement l'exécution de l'action correspondante par l'agent « entreprise » au sein de la plateforme. En effet, le but de la plateforme est de contrôler les échanges sans impacter le comportement des systèmes d'information des entreprises. Elle n'a donc aucune maîtrise sur le comportement de ces derniers. Le comportement de l'agent « entreprise » devant refléter celui du système d'information de son entreprise, ses actions ne doivent pas être conditionnées par une quelconque condition.

(b) L'agent « médiateur » a pour but d'agir en tant qu'intermédiaire entre les agents « entreprise » et donc entre les systèmes d'information des entreprises. Afin de ne pas engendrer d'exceptions, son comportement doit être en conformité avec les spécifications de leur chorégraphie respective. La condition pour que l'agent « médiateur » puisse exécuter une action correspondant à l'envoi d'un document d'affaires à un agent « entreprise » est donc que la reception de ce document soit prévue dans la chorégraphie de l'entreprise de l'agent. La précondition insti-

tutionnelle d'une action est donc qu'elle soit autorisée. Comme il a été discuté dans la section précédente, nous représentons ce concept par l'expression xSL$^{\text{com}}$ modélisant la permission. Ainsi la précondition d'une action (*send* Agt_1 Agt_2 *Doc*), modélisant l'envoi d'un document *Doc* d'un agent Agt_1 à un agent Agt_2, dans le contexte de la sous-institution *inst* est représentée par la formule suivante :

$$(not\ (O\ (not\ (X_{inst}\ (done\ (action\ Agt_1\ (send\ Agt_1\ Agt_2\ Doc))\ true)))))$$

En plus de cette précondition sur l'autorisation d'envoyer un document d'affaires, nous devons également considérer la connaissance de l'agent « médiateur ». En effet, même si l'envoi d'un document spécifique est autorisé, l'agent « médiateur » doit connaître le contenu de ce dernier afin de pouvoir l'envoyer. Il ne doit pas envoyer un document vierge ni le remplir aléatoirement. Le document doit avoir été préalablement envoyé par l'entreprise partenaire puis transmis par l'agent « entreprise » correspondant. Ainsi la précondition institutionnelle est également composée de l'expression xSL$^{\text{com}}$ signifiant que l'agent de l'entreprise partenaire doit avoir effectué l'action d'envoyer le document à l'agent « médiateur ».

Soit (*send* Agt_M Agt_z *Doc*) une action de l'agent « médiateur » Agt_M modélisant l'envoi du document *Doc* à l'agent Agt_Z, et $Inst_Z$ la sous-institution relative à l'agent Agt_Z. Soit Agt_Y l'agent « entreprise » représentant l'entreprise à l'origine du document. La précondition institutionnelle de cette action est modélisée par l'expression xSL$^{\text{com}}$ suivante :

$$\begin{aligned}(and\ &(not\ (O\ (not\ (X_{Inst_Z}\ (done\ (action\ Agt_M\ (send\ Agt_M\ Agt_z\ Doc))\ true)))))\\ &(done\ (action\ Agt_Y\ (send\ Agt_Y\ Agt_M\ Doc))\ true)\end{aligned} \tag{3.1}$$

L'effet institutionnel d'une action modélisant l'envoi d'un document définit l'ensemble des conséquences qu'a ce document sur l'état de l'échange. Il représente l'interprétation du document selon le point de vue de l'entreprise.

Comme il a été discuté précédemment, l'effet institutionnel est composé des éventuels engagements métier impliqués par l'envoi du document. Par exemple, l'envoi d'une facture d'un fournisseur à destination d'un client peut engager ce dernier à payer le montant de la facture. Dans notre approche, les envois de documents sont représentés par des interactions entre les agents « entreprise » et l'agent « médiateur » . Ainsi un document d'affaires d'une entreprise Z à destination d'une entreprise Y sera traduit par l'envoi du document par l'agent « entre-

prise » de Z à l'agent « médiateur ». Nous devons donc identifier le destinataire du document pour pouvoir représenter un engagement métier entre les partenaires commerciaux que sont l'expéditeur et le destinataire. Cette information est indiquée dans le document lui-même. En effet un document d'affaires est composée d'une collection d'informations telles que l'identité des partenaires, la référence de la transaction (p. ex. : n° de la commande)... Les engagements métier entre les partenaires s'expriment donc par des engagements sociaux entre l'entreprise expéditrice et l'entreprise destinataire, tel qu'indiqué dans le document d'affaires concerné. Par exemple, dans un bon de commande au standard UBL [63], les partenaires sont identifiés par les attributs « Customer party » et « Supplier Party » représentant respectivement le client et le fournisseur.

À ces effets sur les engagements métier des partenaires s'ajoutent les conséquences qu'a l'envoi du document sur la chorégraphie des échanges. En effet, l'envoi d'un certain document à une certaine étape de la chorégraphie a un impact sur l'échange en cours. Cela modifie l'ensemble des actions autorisées à l'étape suivante de l'échange entre l'agent « médiateur » et l'agent « entreprise », et peut également entraîner l'obligation d'envoyer un type de document spécifique en retour. L'effet institutionnel d'une action est donc composé des nouveaux engagements métier impliqués par l'envoi du document, de l'ensemble des action autorisées entre l'agent « médiateur » et l'agent « entreprise » à l'étape suivante de l'échange, et de l'engagement sur l'envoi obligatoire d'un document en retour (le cas échéant). Il ne peut y avoir qu'un seul engagement sur l'envoi obligatoire d'un document. En effet, il n'est pas possible que deux documents soient requis au même moment de l'échange. La cooccurennce de deux engagements sur des documents différents entraînerait obligatoirement la violation de l'un d'entre eux. Une telle situation est donc incohérente. Cependant, comme il a été discuté à la section 3.2.1, il est possible que l'envoi d'un document parmi plusieurs possibles soit requis. Nous avons montré que dans ce cas il n'y a qu'un engagment, dont l'objet est modélisé par une disjonction entre les envois de ces différents documents. L'effet institutionnel de l'action ci-dessus est donc modélisé par une expression xSL$^{\text{com}}$ de la forme :

$$
\begin{array}{ll}
(\mathit{and} & \\
\quad \left.\begin{array}{l} (\mathit{not}\ (O\ (\mathit{not}\ (X_{I_z}(\mathit{done}\ \mathit{action}\ \mathit{true})))) \\ \quad\quad \vdots \end{array}\right\} & \text{Ensemble des actions autorisées} \\
\quad (\mathit{commitment}\ \mathit{active}\ \mathit{Agt}_M\ \mathit{Agt}_Z\ \mathit{action}(s)\ \mathit{true}) \rightarrow & \text{Action(s) obligatoire(s)} \\
\quad \left.\begin{array}{l} (\mathit{commitment}\ \mathit{pending}\ Y\ Z\ \mathit{content}1\ \mathit{condition}1) \\ (\mathit{commitment}\ \mathit{pending}\ Z\ Y\ \mathit{content}2\ \mathit{condition}2) \\ \quad\quad \vdots \end{array}\right\} & \text{Ensemble des engagements métier} \\
) &
\end{array}
$$

Afin d'illustrer cette représentation des échanges de documents d'affaires en actions institutionnelles au sein de notre plateforme nous appliquons dans la section suivante notre modélisation à un exemple de chorégraphie.

3.4 Exemple illustratif

Dans cette partie, nous présentons la modélisation de l'ensemble des actions institutionnelles relatives à une chorégraphie. Comme exemple de chorégraphie, nous nous sommes basés sur le processus d'affaires présenté dans l'introduction (cf. Figure 1). La figure 3.3 illustre cette chorégraphie, représentant les échanges entre une entreprise *Buyer* et une entreprise *Seller* dans le cadre d'un processus de commande. On suppose ici qu'elle correspond à la chorégraphie de l'entreprise *Buyer*. Pour chaque type de document pouvant être échangé entre l'entreprise *Buyer* et la plateforme, nous allons définir les effets de l'action institutionnelle correspondante. Nous appelons Agt_B l'agent « entreprise » représentant l'entreprise *Buyer*, I_B son institution, Agt_S l'agent « entreprise » représentant l'entreprise *Seller* et Agt_M l'agent « médiateur ».

Comme il a été discuté ci-dessus, une action de l'agent « entreprise » n'a pas de précondition institutionnelle et une action de l'agent « médiateur » a toujours la même précondition (cf. expression 3.1). Nous ne détaillerons donc que les effets institutionnelles de chaque action.

Envoi du bon de commande (Order message) : Le première action définie par la chorégraphie est l'envoi d'un bon de commande par l'entreprise. Nous modélisons donc ici les effets institutionnelles de l'action ($send\ Agt_B\ Agt_M\ OrderMessage$) (définissant l'envoi d'un bon de commande de l'agent « entreprise » à l'agent « médiateur »). Suite à cette action, l'envoi d'une réponse simple ou d'une réponse détaillée est autorisé (expressions 3.2 et 3.3). De plus, l'envoi d'un de ces deux types de document est requis pour poursuivre l'échange (expression 3.4). Enfin, on peut supposer par exemple que l'interprétation par l'entreprise *Buyer* de l'envoi d'un bon de commande engendre deux engagements métiers conditionnés par l'envoi d'une réponse positive à l'étape suivante de l'échange : les engagements respectifs pour l'entreprise *Buyer* et *Seller* de payer et de livrer les produits (expressions 3.5 et 3.6). Les actions de payer pour les produits commandés et de les livrer sont repésentées respectivement par les expressions $PayProducts$ et $DeliverProducts$. Elles représentent les actions réelles de paiement et de livraison. L'idée ici est de faire intervenir des évènements du monde réel permettant de déterminer de manière sûre la bonne exécution de ces actions (grâce à l'Internet des objets [2] par exemple). Ceci n'a pas été

2. L'Internet des objets a pour but de prolonger l'Internet au monde réel en fixant des étiquettes munies de codes ou d'URLs aux objets ou aux lieux. Ces étiquettes pourront être lues par un dispositif qui transmettra des informations relatives à ces objets et lieux dans un réseau informatique.

développé durant ces travaux de thèse.

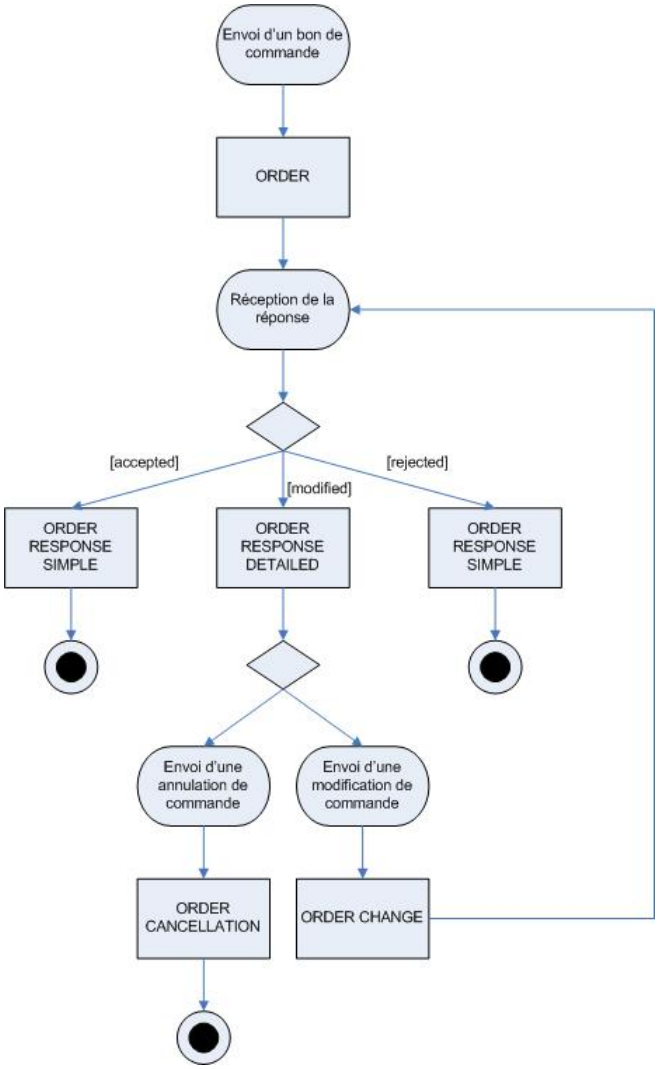


FIGURE 3.3 – Chorégraphie d'un processus de commande

L'information qu'une réponse soit positive ou négative est indiquée dans le document OrderResponseSimple. En UBL, elle est donnée par la valeur de l'attribut « Accepted Indicator » qui peut être égal à $true$ (dans le cas d'une réponse positive) ou $false$ (dans le cas d'une réponse négative). Nous représentons ici une réponse positive par l'expression $OrderResponseSimple^{+}$.

$$(not\ (O\ (not\ (X_{I_B}\ (send\ Agt_M\ Agt_B\ OrderResponseSimple)))))) \tag{3.2}$$

$$(not\ (O\ (not\ (X_{I_B}\ (send\ Agt_M\ Agt_B\ OrderResponseDetailed)))))) \tag{3.3}$$

$$\begin{aligned}
&(commitment\ active \\
&\qquad Agt_M \\
&\qquad Agt_B \\
&\qquad (or\ (X_{I_B}\ (done\ (action\ Agt_M\ (send\ Agt_M\ Agt_B\ OrderResponseSimple)) \\
&\qquad\qquad true)) \\
&\qquad\quad (X_{I_B}\ (done\ (action\ Agt_M\ (send\ Agt_M\ Agt_B\ OrderResponseDetailed)) \\
&\qquad\qquad true)) \\
&\qquad true)
\end{aligned} \tag{3.4}$$

$$\begin{aligned}
&(commitment\ pending \\
&\qquad Agt_B \\
&\qquad Agt_S \\
&\qquad PayProducts \\
&\qquad (X_{I_B}\ (done\ (action\ Agt_M\ (send\ Agt_M\ Agt_B\ OrderResponseSimple^+)) \\
&\qquad\qquad true)))
\end{aligned} \tag{3.5}$$

$$\begin{aligned}
&(commitment\ pending \\
&\qquad Agt_S \\
&\qquad Agt_B \\
&\qquad DeliverProducts \\
&\qquad (X_{I_B}\ (done\ (action\ Agt_M\ (send\ Agt_M\ Agt_B\ OrderResponseSimple^+)) \\
&\qquad\qquad true)))
\end{aligned} \tag{3.6}$$

Envoi d'une réponse simple (Order Response Simple) : L'envoi d'une réponse simple par l'agent « médiateur » (quelle soit positive ou négative) met fin à l'échange en cours. L'action institutionnelle correspondante n'a donc pas d'effet institutionnel. Cependant, elle a quand même des effets sur l'état de l'échange, bien que ces effets soient indirects. En effet, en cas de réponse positive, la formule $(X_{I_B}\ (done\ (action\ Agt_M\ (send\ Agt_M\ Agt_B\ OrderResponse\text{-}Simple^+))))$ deviendra vraie, ce qui activera les engagements métier créés suite à l'envoi du bon de commande. En cas de réponse négative, cette formule deviendra alors fausse, ce qui provoquera l'annulation de ces engagements métier=.

Nous pouvons remarquer que cette action n'est autorisée que suite à l'envoi d'un bon de commande. L'acteur de l'action étant l'agent « médiateur », elle a une précondition institutionnelle indiquant qu'elle ne doit s'exécuter que si elle est autorisée. Dans notre exemple, ce sera

donc uniquement le cas suite à l'interprétation de l'action correspondant à l'envoi d'un bon de commande définie ci-dessus.

Envoi d'un réponse détaillée (Order Response Detailed) : Contrairement à l'envoi d'une réponse simple, l'envoi d'une réponse détaillée ne met pas fin à l'échange. Au contraire, une réponse détaillée permet de proposer une modification de la commande. C'est pourquoi, dans notre exemple, après avoir reçu une réponse détaillée, l'entreprise *Buyer* doit envoyer en retour soit une modification soit une annulation de la commande. Les expressions 3.7 et 3.8 indiquent que ces actions sont autorisées et l'expression 3.9 modélise l'engagement décrivant le fait que l'exécution d'une de ces actions est obligatoire.

$$(not\ (O\ (not\ (X_{I_B}\ (send\ Agt_B\ Agt_M\ OrderChange)))))) \tag{3.7}$$

$$(not\ (O\ (not\ (X_{I_B}\ (send\ Agt_B\ Agt_M\ OrderCancellation)))))) \tag{3.8}$$

$$\begin{aligned}
&(commitment\ active \\
&\qquad Agt_B \\
&\qquad Agt_M \\
&\qquad (or\ (X_{I_B}\ (done\ (action\ Agt_B\ (send\ Agt_B\ Agt_M\ OrderChange)) \\
&\qquad\qquad\qquad true)) \\
&\qquad\quad (X_{I_B}\ (done\ (action\ Agt_B\ (send\ Agt_B\ Agt_M\ OrderCancellation)) \\
&\qquad\qquad\qquad true)) \\
&\qquad true)
\end{aligned} \tag{3.9}$$

Envoi d'une annulation de commande (Order Cancellation) : L'envoi d'une annulation de commande met fin à l'échange en cours. Cet envoi n'est permis dans la chorégraphie qu'à la suite d'une réponse détaillée. À ce stade de l'échange, les engagements métier impliqués par l'envoi du bon de commande ont été annulés et plus aucun échange de document n'est possible. L'action correspondante n'a donc aucun effet explicite.

Envoi d'une modification de commande (Order Change) : Tout comme l'envoi d'une annulation de commande, l'envoi d'une modification n'est permis qu'à la suite d'une réponse détaillée. Ce document a pour but de proposer une nouvelle commande pour laquelle l'entreprise attend une réponse. Cette action renvoie l'échange dans le même état que suite à l'envoi d'un bon de commande. Ses effets sont donc les mêmes.

État initial : Nous venons de décrire la modélisation des actions relatives à la chorégraphie illustrée par la figure 3.3. Pour chaque action, nous avons notamment défini les actions autorisées à l'étape suivante de l'échange. Cependant, les actions autorisées à l'état initial de l'échange n'ont pas été définies. Nous devons donc déterminer l'état initial de l'échange.

Dans notre exemple, l'état initial de l'échange définit que l'action d'envoyer un bon de commande est autorisée (expression 3.10). En outre, il est également composé de l'engagement global que nous avons défini à la section 3.2.2 (expression 3.11).

$$(not\ (O\ (not\ (X_{I_B}\ (send\ Agt_B\ Agt_M\ OrderMessage)))))) \tag{3.10}$$

$$\begin{aligned}
&(commitment\ active\\
&\qquad ??Agt_1\\
&\qquad ??Agt_2\\
&\qquad (or\ (not\ (X_{inst}\ (done\ (action\ ??Agt_1\ (send\ ??Agt_1\ ??Agt_2\ Doc))\ true))))\\
&\qquad\quad (not\ (O\ (not\ (X_{inst}\ (done\ (action\ ??Agt_1\ (send\ ??Agt_1\ ??Agt_2\ Doc))\ true)))))\\
&\qquad true)
\end{aligned} \tag{3.11}$$

3.5 Synthèse

Dans cette section, nous avons décrit le modèle que nous proposons pour représenter la sémantique impliquée par les interactions au cours d'échanges B2B. Ce modèle permet de modéliser à la fois le niveau métier en exprimant les engagements métier engendrés par l'envoi de certains documents, et le niveau processus en représentant les contraintes sur le déroulement des échanges tel que spécifié dans la chorégraphie. Ces informations sont définies dans un cadre institutionnel en modélisant chaque échange d'un document d'affaires entre une entreprise et la plateforme de médiation par une action institutionnelle. Les effets d'une action institutionnelle sont interprétés dans le contexte d'une institution représentant le point de vue de l'entreprise. Ceci nous permet de prendre en compte les différents points de vue de chaque entreprise afin de détecter une éventuelle incohérence entre les états d'entreprises partenaires.

Ce modèle est basé sur le concept d'engagement social. Ce concept permet de représenter de manière explicite l'état courant d'un engagement et de définir son cycle de vie. La transition d'un engagement vers l'état de violation révèle la survenue d'une exception. La modélisation des transitions définies dans le cycle de vie d'un engagement permet à la plateforme de détecter les exceptions de manière automatique.

Dans la section suivante, nous présentons le comportement que nous avons défini pour la plateforme. Nous nous basons sur le modèle présenté dans cette section afin de définir des règles de raisonnement et de comportement permettant à la plateforme de suivre l'évolution des échanges en cours et d'en déterminer les actions à effectuer.

4

Comportement de la plateforme

Dans l'introduction de cette partie, nous avons présenté la représentation des échanges B2B au sein de notre plateforme. Ces échanges sont définis sous la forme d'actions institutionnelles modélisant les envois de documents d'affaires entre les systèmes d'information des entreprises. Les effets de ces actions retranscrivent alors l'interprétation de l'envoi de ces documents selon le point de vue des entreprises. Cette représentation des échanges B2B permet de suivre l'évolution des échanges entre les partenaires B2B. Dans cette section, nous exposons les principes de raisonnement des agents permettant de déterminer leur comportement vis-à-vis de ces actions. Nous définissons notamment les règles de raisonnement à l'origine du comportement de l'agent « médiateur ». Ces règles permettent de déterminer ses critères de décisions concernant l'exécution d'une actions.

4.1 Interprétation institutionnelle d'une action

Pour commencer, nous présentons les algorithmes décrivant l'interprétation institutionnelle (par un agent) de l'exécution d'une action (Algorithmes 1 et 2).

Lorsqu'un agent observe l'exécution d'une action, il recherche les pouvoirs institutionnels associés à cette action dans sa base de connaissance. Il interprète l'effet institutionnel d'un pouvoir uniquement si sa précondition institutionnelle est satisfaite. Dans ce cas, l'exécution de l'action est également reconnue comme fait institutionnel de l'institution.

Algorithme 1: Interprétation institutionnelle de l'exécution d'une action

```
1  while true do
2  |  if (done (action ??Agt ??Act)) then
3  |  |  L_P = SearchInstitutionalPowers((action ??Agt ??Act));
4  |  |  forall the (power ??Agt ??Inst ??Cond_Act ??Act ??Eff_Act) ∈ L_P do
5  |  |  |  if ??Cond_Act then
6  |  |  |  |  (D_??Inst ??Eff_Act) and (D_??Inst (done (action ??Agt ??Act) true));
7  |  |  |  end
8  |  |  end
9  |  end
10 end
```

Algorithme 2: Récupération des pouvoirs institutionnels (méthode SearchInstitutionalPowers)

Data : une action (*action Acteur Action*) et la base de connaissance de l'agent KB
Result : Liste L_P des pouvoirs institutionnels de *Acteur* en exécutant *Action* selon KB

```
1 L_P = new List(Powers);
2 forall the (power ??Agt ??Inst ??Cond_Act ??Act ??Eff_Act) ∈ KB do
3 |  if ??Agt == Acteur and ??Act == Action then
4 |  |  L_P.add((power ??Agt ??Inst ??Cond_Act ??Act ??Eff_Act));
5 |  end
6 end
```

Note : Chaque agent n'a connaissance que des pouvoirs relatifs aux institutions auxquelles il appartient.

4.1.1 Mise à jour des engagements courants

L'interprétation des actions permet ainsi aux agents de mettre à jour leur base de connaissance. Suite à cette mise à jour, l'état des engagemens courants peut avoir changé et doit donc ègalement être mis à jour.

L'algorithme 3 présente la mise à jour de l'état des engagements courants. Il implémente les règles de transition définissant le cycle de vie d'un engagement présentées à la section 3.2.1.

4.2 Comportement des agents « entreprise »

Nous décrivons maintenant le raisonnement des agents « entreprise ».

Algorithme 3: Mise à jour de l'état des engagements

```
1  while true do
2      forall the (commimtment ??state ??debtor ??creditor ??content ??condition) do
3          switch ??state do
4              case PENDING
5                  if ??content then                                   /* Règle 6 */
6                      ??state ← FULFILLED;
7                  else
8                      if ??condition then                             /* Règle 3 */
9                          ??state ← ACTIVE;
10                     else                                            /* Règle 4 */
11                         if ¬??condition then
12                             ??state ← CANCELLED;
13                         end
14                     end
15                 end
16                 break;
17             endsw
18             case ACTIVE                                  /* ??condition == true */
19                 if ??content then                                   /* Règle 1 */
20                     ??state ← FULFILLED;
21                 else
22                     if ¬??content then                              /* Règle 2 */
23                         ??state ← VIOLATED;
24                     end
25                 end
26                 break;
27             endsw
28         endsw
29     end
30 end
```

Comme nous l'avons exposé à la section II, chaque agent « entreprise » est interfacé directement avec le système d'information de son entreprise. Son rôle est de transférer les documents d'affaires entre le système d'information de l'entreprise et l'agent « médiateur » (A_M). Lorsqu'un document d'affaires est envoyé par le système d'information d'une entreprise, l'agent représentant cette entreprise transmet le document à l'agent « médiateur » en effectuant l'action institutionnelle correspondante, définie selon notre modèle. De manière symétrique, suite à l'envoi d'un document d'affaires par l'agent « médiateur » à un agent « entreprise », ce dernier transmet le document au système d'information de son entreprise. Chaque agent « entreprise » a donc le même comportement décrit par l'algorithme 4.

Comme il a été discuté à la section 3.3, l'exécution d'une action par l'agent « entreprise » n'est

pas soumise à une condition. En effet, son comportement modélise le comportement du système d'information de l'enterprise. Ainsi chaque envoi de document de ce dernier déclenche automatiquement l'exécution de l'action correspondante par l'agent « entreprise ». Ceci permet de contrôler le bon comportement du système d'information de l'entreprise au sein de la plateforme par le biais du contrôle du comportement de l'agent « entreprise ». De même, chaque envoi de document de la part de l'agent « médiateur » à un agent « entreprise » est reconnu dans la sous-institution relative à l'entreprise uniquement s'il est conforme aux spécifications du système d'information de son entreprise. En effet, selon notre modélisation des actions d'interactions, la précondition institutionnelle d'une action de l'agent « médiateur » est qu'elle soit permise, c'est-à-dire attendue par le système d'information de l'entreprise concernée. L'agent « entreprise » n'a donc qu'à transférer ce document au système d'information de son entreprise.

Algorithme 4: Raisonnement de l'agent « entreprise » A_E

```
1 while true do
2 |   if receivefromIS(Doc) then
3 |   |   exec((action A_E (send A_E A_M Doc)));
4 |   end
5 |   if (D_{Inst_E} (done (action A_M (send A_M A_E Doc)) true)) then
6 |   |   sendToIS(Doc);
7 |   end
8 end
```

Les méthodes $receiveFromIS$ et $sendToIS$ représentent l'interface avec le système d'information de l'entreprise. Elles expriment respectivement la réception et l'envoi d'un document provenant/à destination du système d'information de l'entreprise de l'agent. La méthode *exec* traduit l'exécution effective d'une action. Ainsi *exec*((*action Agt Act*) exécute le code de l'action *Act* et implique la formule (*done* (*action Agt Act*)) (qui permettra alors l'interprétation institutionnelle de l'action).

4.3 Comportement de l'agent « médiateur »

L'état interne d'un agent est déterminé par ses croyances et ses intentions. L'exécution d'une action par l'agent « médiateur » est provoquée par l'intention de cet agent d'exécuter l'action. Nous avons donc défini des règles qui génèrent les intentions de l'agent « médiateur » en fonction de l'état de son environnement (correspondant à l'état des échanges).

Nous présentons dans un premier temps les règles de raisonnement de base, concernant l'interprétation des intentions dans un contexte institutionnel. Nous décrivons ensuite les régles spécifiques liées au comportement de l'agent « médiateur ». Ces dernières s'appuient sur les

engagements courants.

4.3.1 Règles sur les intentions

Lorsque l'agent « médiateur » (Agt_M) a l'intention qu'une formule soit vraie dans une certaine institution (c'est-à-dire qu'elle soit reconnue comme fait institutionnel de l'institution), ou qu'une action institutionnelle spécifique soit effectuée, et que cette intention n'est pas satisfaite, il doit chercher les moyens dont il dispose pour la satisfaire. Les règles ci-dessous décrivent l'interprétation de l'agent par rapport à ses intentions.

R_i1 - Lorsque l'agent « médiateur » a l'intention qu'une action institutionnelle, dont il est l'acteur, soit exécutée, si la précondition institutionnelle de cette action est satisfaite, alors il l'exécute afin de satisfaire son intention.

R_i2 - Lorsque l'agent « médiateur » a l'intention qu'une action institutionnelle *Act*, dont il est l'acteur, soit exécutée mais que sa précondition institutionnelle ($Cond_{Act}$) n'est pas vérifiée, alors il adopte l'intention de satisfaire cette précondition.

$$(I\ Agt_M\ (D_{inst}\ (done\ (action\ Agt_M\ Act)\ true))) \wedge (not\ (B\ Agt_M\ Cond_{Act})) \rightarrow (I\ Agt_M\ Cond_{Act})$$

Cette règle permet à l'agent de ne pas rester passif en attendant de pouvoir exécuter une action.

R_i3 - Lorsque l'agent « médiateur » a l'intention qu'une formule F soit vraie dans une institution *inst*, il recherche dans sa base de connaissance une action institutionnelle *Act* dont l'exécution implique F dans *inst* (sous la précondition institutionnelle $Cond_{Act}$). S'il trouve une telle action, il adopte alors l'intention qu'elle soit exécutée.

$$(I\ Agt_M\ (D_{inst}\ F)) \wedge (Cond_{Act} \wedge (done\ Act\ true) \Rightarrow_{inst} F) \rightarrow (I\ Agt_M\ (D_{inst}\ (done\ Act\ true)))$$

R_i4 - Dans le cas où l'agent « médiateur » a l'intention qu'une formule F soit vraie dans une institution *inst* mais qu'il ne trouve aucune action institutionnelle dont l'exécution implique F, il adopte alors l'intention qu'un agent tiers $??Agt_X$ soit engagé à satisfaire F (sous une condition $??Cond$).

$$(I\ Agt_M\ (D_{inst}\ F)) \wedge (not\ (exists\ (??Act\ ??Agt) \\ (??Cond_{Act} \wedge (done\ ??Act\ true) \Rightarrow_{inst} F))) \\ \rightarrow (I\ Agt_M\ (D_{inst}\ (commitment\ pending\ ??Agt_X\ Agt_M\ F\ ??Cond)))$$

Cette règle traduit la notion de délégation. Elle exprime le fait qu'un agent, ne pouvant satisfaire une de ses intentions, va confier la résolution de ce problème à un agent tiers (en l'engageant à la satisfaire). Elle s'applique en particulier dans le cas où l'agent « médiateur » a l'intention qu'une action *Act* soit exécutée mais qu'il n'est pas l'acteur de l'action. Soit Agt_{Act} l'acteur de l'action. Dans ce cas en effet, l'agent « médiateur » cherchera à satisfaire la formule $(done\ (action\ Agt_{Act}\ Act)\ true)$. Il ne pourra alors trouver aucune action institutionnelle ayant pour effet le fait que l'agent Agt_{Act} ait effectué l'action *Act*. Il cherchera alors une action institutionnelle créant l'engagement que l'agent Agt_{Act} exécute l'action *Act*.

Les règles ci-dessus définissent le raisonnement de base de l'agent « médiateur » par rapport à ses intentions. Afin de déterminer son comportement au cours des échanges, nous avons défini les règles à l'origine de ces intentions. Ces règles sont présentées ci-dessous.

4.3.2 Règles sur les engagements courants

Le rôle de l'agent « médiateur » est d'éviter la survenue d'exceptions durant les échanges B2B. Ces exceptions sont modélisées par la violation des engagements. L'objectif de l'agent « médiateur » est donc de satisfaire les engagements courants.

Nous proposons de déterminer le comportement de l'agent « médiateur » en fonction des engagements courants. Nous avons défini deux règles dépendant du rôle de l'agent dans l'engagement concerné (débiteur ou créditeur) et de l'état courant de l'engagement.

R_c1 - Si l'agent « médiateur » est le débiteur d'un engagement dans l'institution *inst* et que l'engagement est dans l'état *active* (c'est-à-dire que la condition est vérifiée), alors l'agent adopte l'intention de satisfaire l'objet de l'engagement.

$$(D_{inst}\ (commitment\ active\ Agt_M\ Agt_X\ Content\ Condition)) \\ \rightarrow (I\ Agt_M\ (D_{inst}\ Content))$$

R_c2 - Si l'agent « médiateur » est le créditeur d'un engagement dans l'institution *inst* et que l'engagement est dans l'état *pending* (c'est-à-dire que sa condition n'est pas vérifiée), alors l'agent adopte l'intention de satisfaire la condition de l'engagement.

$$(D_{inst}\ (commitment\ pending\ Agt_X\ Agt_M\ Content\ Condition))$$
$$\rightarrow (I\ Agt_M\ (D_{inst}\ Condition))$$

Ces règles décrivent le fait que l'agent « médiateur » a toujours l'intention que les engagements courants soient satisfaits. Ainsi lorsqu'il est le débiteur, si l'engagement est dans l'état *active*, il va tenter de satisfaire son objet afin de ne pas violer l'engagement et ainsi ne pas générer d'exception (règle R_c1). À l'opposé, s'il est le créditeur d'un engagement dans l'état *pending*, il va vouloir satisfaire la condition afin d'activer l'engagement et ainsi obliger le débiteur à satisfaire l'objet de l'engagement (règle R_c2).

4.3.3 Algorithme de raisonnement de l'agent « médiateur »

Le comportement de l'agent « médiateur » (A_M) est caractérisé par les règles spécifiées ci-dessus en fonction de ses connaissances (représentées dans sa base de connaissance) et de l'interprétation des actions exécutées au cours des échanges. L'algorithme 7 représente le raisonnement qui détermine ce comportement. Nous détaillons ici cet algorithme.

Les lignes 3 à 11 formalisent les règles R_i1 et R_i2. Ainsi, lorsque l'agent « médiateur » a l'intention d'avoir exécuté une action *Act* dans le contexte d'une institution *Inst*, il récupère la précondition institutionnelle du pouvoir relatif à l'exécution de *Act* dans *Inst* (méthode *GetInstitutionalCondition*, cf. Algorithme 5) et vérifie dans sa base de connaissance s'il croit qu'elle est satisfaite. Si tel est le cas (règle R_i1) alors il l'exécute (méthode *exec*). Sinon (règle R_i2), il adopte l'intention de la satisfaire.

Les lignes 12 à 20, quant à elles, formalisent les règles R_i3 et R_i4. Elle considèrent l'intention de satisfaire une formule F dans le contexte d'une institution *Inst*. Il est à noter que cette intention est générique et peut également se rapporter à l'intention d'avoir exécuté une action. C'est pourquoi il est important de considérer ce cas après celui décrit précédemment. Ainsi, ces règles ne s'appliqueront pas si la formule décrit le fait que l'agent ait effectué une action car une des règles précédentes aura été appliquée. Afin de satisfaire son intention, l'agent « médiateur » va donc rechercher un action institutionnelle ayant pour effet F dans *Inst* (méthode *SearchInstitutionalAction*, cf. Algorithme 6). S'il trouve une telle action (règle R_i3), alors il

adopte l'intention de l'exécuter. Sinon (règle R_i4), il adopte l'intention d'engager un agent tiers à la satisfaire.

Enfin, les lignes 21 à 24 et 25 à 28 formalisent respectivement les règles R_c1 et R_c2.

Algorithme 5: Récupération de la précondition institutionnelle (méthode GetInstitutionalCondition)

Data : une action A, une institution I et une base de connaissance KB
Result : une formule correspondant à la précondition institutionnelle de Act dans le contexte de $Inst$ selon KB

```
1 forall the (power ??Agt ??Inst ??Cond ??Act ??Eff) ∈ KB do
2 |   if ??Inst == I and ??Act == A then
3 |   |   return ??Cond;
4 |   end
5 end
6 return null;
```

Algorithme 6: Recherche d'une action institutionnelle (méthode SearchInstitutionalAction)

Data : une formule F, une institution I et une base de connaissance KB
Result : une action Act ayant pour effet institutionnel F dans le contexte de $Inst$ selon KB

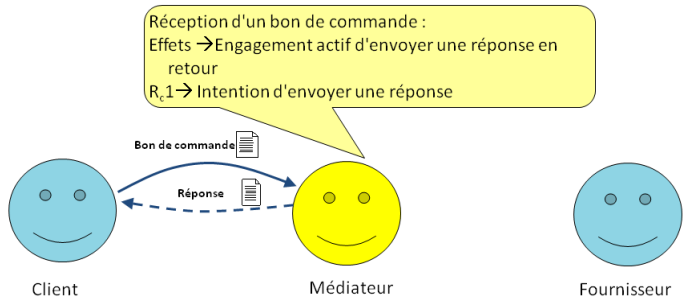
```
1 forall the (power ??Agt ??Inst ??Cond ??Act ??Eff) ∈ KB do
2 |   if ??Inst == I and ??Eff == F then
3 |   |   return ??Act;
4 |   end
5 end
6 return null;
```

Algorithme 7: Raisonnement de l'agent « médiateur »

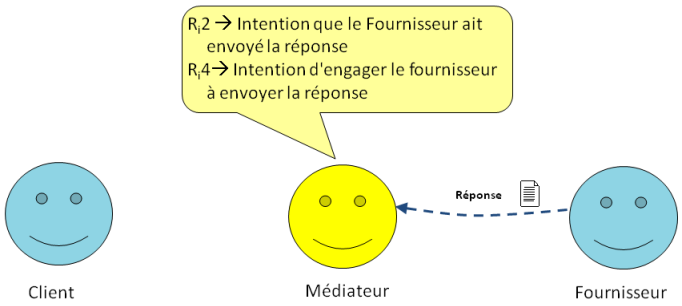
```
1  while true do
2      switch intention do
3          case (I Agt_M (D_Inst (done (action Agt_M Act) true))
4              Cond_Act = GetInstitutionalCondition(Act,Inst);
5              if (B Agt_M Cond_Act) ; then                          /* R_i1 */
6                  exec((action Agt_M Act)) ;
7              else                                                  /* R_i2 */
8                  (I Agt_M Cond_Act) ;
9              end
10             break;
11         endsw
12         case (I Agt_M (D_Inst F))
13             Act = SearchInstitutionalAction(F,Inst);
14             if Act ≠ null ; then                                  /* R_i3 */
15                 (I Agt_M (D_Inst (done Act true))) ;
16             else                                                  /* R_i4 */
17                 (I Agt_M (D_Inst (commitment pending ??Agt_X Agt_M F ??Cond))) ;
18             end
19             break;
20         endsw
21         case (D_Inst (commitment active Agt_M ??Agt_X ??Content ??Condition)) /* R_c1
           */
22             (I Agt_M (D_Inst ??Content));
23             break;
24         endsw
25         case (D_Inst (commitment pending ??Agt_X Agt_M ??Content ??Condition))
           /* R_c2 */
26             (I Agt_M (D_Inst ??Condition));
27             break;
28         endsw
29     endsw
30 end
```

4.4 Illustration

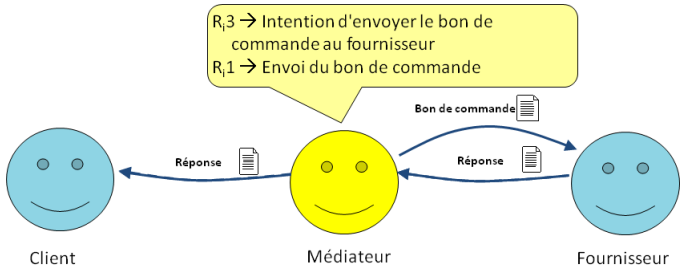
Pour plus de clarté, nous illustrons maintenant le raisonnement de l'agent « médiateur » en reprenant l'exemple de la section 3.4. Nous décrivons le comportement de l'agent « médiateur » durant les premières étapes de la chorégraphie présentée à la figure 3.3. Nous supposons que les deux partenaires, représentés ici par l'agent *Client* et l'agent *Fournisseur* agissent en accord avec la chorégraphie. Nous détaillons le raisonnement de l'agent *Médiateur* lors de la réception d'un bon de commande. La figure 4.1 expose ce raisonnement que nous avons découpé en trois étapes.



(a) Étape 1



(b) Étape 2



(c) Étape 3

FIGURE 4.1 – Exemple de comportement

Étape 1 : Lors de l'envoi d'un bon de commande (à destination de l'agent *Fournisseur*) de l'agent *Client* à l'agent *Médiateur*, l'interprétation des effets de cette action va créer un engagement de l'agent *Médiateur* envers l'agent *Client* à lui envoyer une réponse en retour. L'agent *Médiateur* étant le débiteur de cet engagement et ce dernier étant dans l'état *active* (précondition == true), l'agent va alors adopter l'intention de satisfaire

l'objet de l'engagement (Règle R_c1), c'est-à-dire d'envoyer une réponse à l'agent *Client* (symboliser par la flèche en pointillés sur la figure).

Étape 2 : Pour pouvoir envoyer la réponse de l'agent *Fournisseur* à l'agent *Client*, ce document doit avoir préalablement été envoyé par l'agent *Fournisseur* (précondition de l'action). L'agent *Médiateur* adopte donc l'intention que l'agent *Fournisseur* lui envoie sa réponse (Règle R_i2). Afin de satisfaire son intention, l'agent *Médiateur* va alors adopter l'intention d'engager l'agent *Fournisseur* sur l'envoi de sa réponse (Règle R_i4).

Étape 3 : L'envoi du bon de commande à l'agent *Fournisseur* engage ce dernier à envoyé une réponse en retour. L'agent *Médiateur* adopte donc l'intention d'envoyer le bon de commande à l'agent *Fournisseur* (Règle R_i3). Les préconditions de cette action étant satisfaites, l'agent *Médiateur* envoie le bon de commande à l'agent *Fournisseur* (Règle R_i1). L'agent *Fournisseur* envoie alors sa réponse en retour, ce qui permet à l'agent *Médiateur* de la transmettre à l'agent *Client* et ainsi de satisfaire son engagement initial.

Nous venons de voir comment l'ensemble des règles définies dans cette section détermine le comportement de l'agent « médiateur ». Dans la section suivante nous discutons sur les capacités de médiation de notre plateforme qui découlent de ce comportement.

4.5 Capacités de médiation de la plateforme

L'utilisation du modèle d'engagement social pour représenter les échanges B2B au sein de notre plateforme permet de détecter la survenue d'une exception. La plateforme de médiation est ainsi capable de déceler un comportement anormal de la part d'un partenaire B2B. Mais au delà de permettre la détection des exceptions, cette représentation a pour but principal de modéliser les capacités des systèmes d'information afin de se conformer aux spécificités de leur chorégraphie.

Les chorégraphies de chaque système d'information étant potentiellement incompatibles, la plateforme doit pouvoir se comporter de manière à outrepasser les problèmes d'interopérabilité. Nous entrons alors dans le domaine de la médiation de processus. Dans la section 1.1, nous avons présenté en quoi consiste la médiation de processus. En particulier, nous avons exposé les différents types de problème d'interopérabilité pouvant se rencontrer entre deux chorégraphies : (a) les documents non-attendus, (b) les documents requis non envoyés, (c) les ordonnancements différents de documents. La figure 4.2 illustre ces types de problème. Nous allons maintenant voir comment notre représentation des échanges B2B associée aux règles de raisonnement définies dans la section précédente apportent une réponse à ces problèmes.

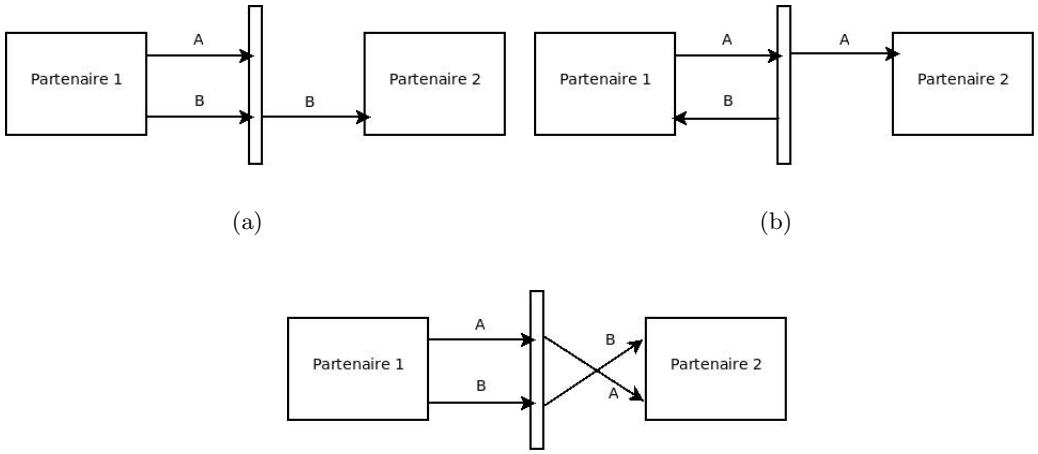


FIGURE 4.2 – Problèmes d'interopérabilité entre chorégraphies

(a) Lorsqu'un document B est envoyé par une entreprise $P1$ à une entreprise $P2$ mais que le système d'information de $P2$ ne prévoit pas de recevoir un tel document, cela peut engendrer une exception.

La résolution de ce problème est inhérent à la représentation des échanges dans notre plateforme de médiation. En effet, l'agent « médiateur » est l'intermédiaire entre les partenaires. Ainsi c'est lui qui recevra le document envoyé par $P1$. Cependant il ne pourra le transmettre à $P2$ qu'à la condition que cette action soit permise par la chorégrpahie du système d'information de $P2$. Ainsi aucun document non-attendu ne pourra être transmis.

(b) Dans le cas de documents requis non-envoyés, nous distinguons deux cas :

1. Dans [16], les auteurs considèrent la transmission d'accusés de réception suite à l'envoi d'un document A d'une entreprise $P1$ à une entreprise $P2$. Si cet accusé de réception n'est pas envoyé par $P2$, cela met $P1$ dans un état de blocage et engendre donc une exception. La représentation des échanges tel que nous l'avons décrite ne permet pas de résoudre ce problème car l'envoi de l'accusé de réception sera conditionné par l'envoi préalable de ce document par $P2$. Cependant, ce cas peut être facilement traité en remplaçant cette précondition par une formule représentant le fait que l'agent « médiateur » ait transmis le document A à $P2$. Ainsi, l'agent « médiateur » pourra envoyer un accusé de réception une fois le document A transmis, même si la chorégraphie de $P2$ ne prévoit pas l'envoi d'un accusé de réception. Il faut cependant faire attention car la plateforme prend alors la responsabilité que $P2$ ait bien reçu le document A.
2. Lorsque les documents requis ne sont pas des accusés de réception, la génération d'un tel

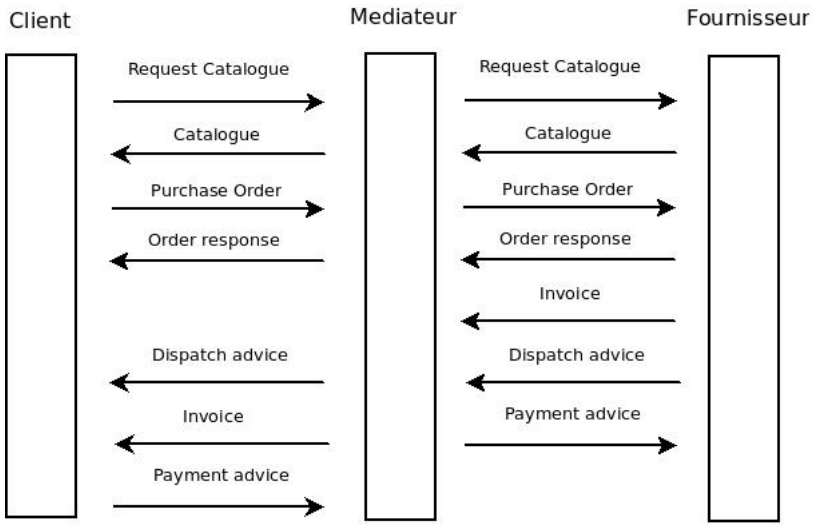


FIGURE 4.3 – Exemple de problème d'ordonnancement (1)

document ne peut être faite que par le partenaire concerné. Un tel problème ne peut donc pas être résolu.

(c) Lorsque les ordonnancements des documents, tel que défini dans les chorégraphies des partenaires, diffèrent, la plateforme doit pouvoir changer l'ordre de ces documents afin de se conformer aux spécificités de chaque chorégraphie. Comme dans le cas (a), la plateforme résoudra le problème automatiquement lorsque cela est possible. En effet, l'agent « médiateur » n'ayant l'autorisation d'envoyer des documents à un partenaire que lorsque cela est permis par le système d'information de ce dernier, il ne pourra pas transmettre les documents dans le mauvais ordre. Supposons par exemple qu'au cours d'un échange B2B de commande-livraison, l'agent « médiateur » reçoive d'abord la facture (*Invoice*) puis l'avis de livraison (*Dispatch advice*) de la part du fournisseur alors que ces documents sont attendus dans l'ordre inverse par le partenaire client. Le scénario de la figure 4.3 décrit ce problème. L'action de transmettre la facture au client ne sera alors permise qu'à la suite de l'envoi de l'avis de livraison selon la représentation des échanges dans notre plateforme. L'agent « médiateur » transmettra donc en premier l'avis de livraison, après l'avoir reçu, puis la facture.

Cette situation n'est malheureusement qu'un cas particulier car les documents concernés sont envoyés par le même partenaire. Supposons par exemple que la chorégraphie du client prévoit d'envoyer l'avis de paiement uniquement après avoir reçu l'avis de livraison alors que le fournisseur attend de recevoir l'avis de paiement pour envoyer l'avis de livraison (cf. Figure 4.4). Dans ce cas, nous nous retrouvons dans un état de blocage où chaque partenaire attend une action de l'autre partenaire pour poursuivre l'échange et où l'agent « médiateur » n'a aucune possibilité pour le débloquer.

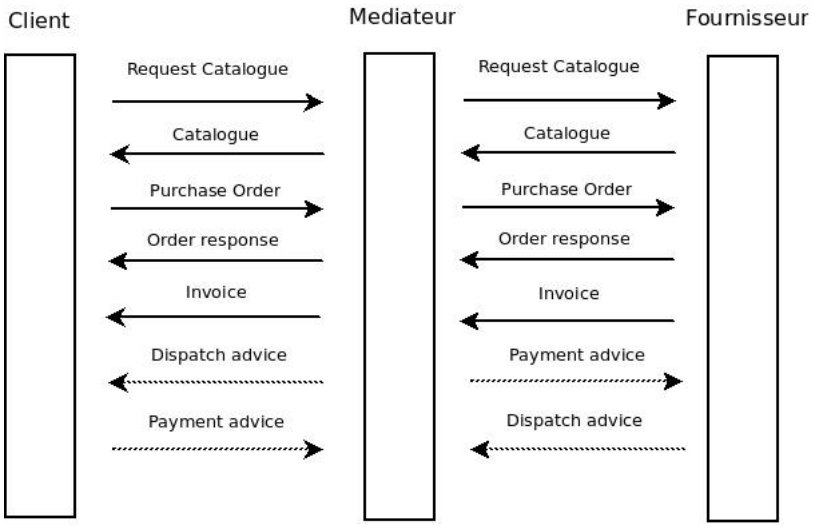


FIGURE 4.4 – Exemple de problème d'ordonnancement (2)

De manière générale, la représentation des échanges au sein de notre plateforme, associée aux règles de raisonnement que nous avons définies, produit le comportement de l'agent « médiateur » tel qu'il respecte les contraintes liées aux systèmes d'information des entreprises. Ainsi, il ne pourra pas envoyer de document si ce n'est pas permis et il cherchera tous les moyens possibles pour envoyer un document requis. Cependant, comme nous venons de le voir, bien que certains problèmes d'interopérabilité puissent être résolus au cours des échanges, il y a toujours des cas où l'incompatibilité entre les chorégraphies des partenaires peut engendrer des exceptions qui ne peuvent pas être résolues. Afin d'éviter ces situations irrécupérables, la plateforme doit alors être capable d'anticiper les problèmes afin d'adapter son comportement en conséquence, dans le but de les éviter.

Dans la section suivante, nous proposons donc un mécanisme d'anticipation basé sur la théorie des jeux. Les échanges entre les partenaires via la plateforme sont représentés sous la forme d'un jeu, que nous appelons *jeu de médiation*. L'arbre de jeu correspondant est construit à partir des actions définies dans notre plateforme pour représenter les interactions avec les partenaires B2B. Il décrit l'ensemble des scénarios possibles au cours des échanges entre les partenaires via la plateforme.

5

Jeu de médiation

Les règles de raisonnement que nous avons définies dans la section précédente permettent de déterminer de manière dynamique le comportement de l'agent « médiateur » au cours d'échanges B2B. Néanmoins, cet aspect dynamique ne permet pas à l'agent d'anticiper les problèmes d'interopérabilité pouvant survenir et, le cas échéant, il est parfois impossible de les résoudre « à la volée ». Par conséquent un mécanisme d'anticipation est nécessaire afin de pouvoir raisonner sur les problèmes potentiels qui pourront se rencontrer au cours d'échanges entre des partenaires.

Dans le but de concevoir un tel mécanisme, nos travaux se sont d'abord concentrés sur le domaine de la planification. L'objectif était de planifier les actions de l'agent « médiateur » de façon à calculer un scénario pour lequel les échanges se déroulent sans problème, c'est-à-dire sans provoquer la violation d'un engagement. Cependant, le déroulement des échanges ne dépend pas uniquement des actions de l'agent « médiateur », mais également des choix des partenaires parmi les documents qu'ils peuvent envoyer. Dès lors, réussir à calculer un tel plan ne permet pas d'affirmer que les échanges se dérouleront effectivement de la sorte au moment de leur exécution. C'est pourquoi nos recherches se sont orientées vers le domaine de la théorie des jeux. Nous présentons dans cette section les résultats de nos recherche. Nous décrivons brièvement dans un premier temps le domaine de la théorie des jeux avant de définir ce que nous avons appelé un *jeu de médiation*.

5.1 La théorie des jeux

La théorie des jeux [45][70] est un formalisme qui a pour but d'étudier les interactions entre agents. Le principe consiste à représenter les situations où les agents sont conduits à faire des

choix parmi un certain nombre d'actions possibles, par un jeu où chaque agent est appelé un *joueur*. Un gain (positif ou négatif) propre à chaque joueur, appelé son *utilité*, est associé à chaque situation. Le but d'un joueur est alors de prendre la meilleure décision en fonction des actions des autres joueurs afin de maximiser son utilité.

On distingue différents types de jeux :

- les jeux coopératifs / non-coopératifs. Un jeu est dit coopératif si les joueurs peuvent se consulter afin de trouver un accord. Sinon il est dit non-coopératif.
- les jeux à somme nulle / non-nulle. Un jeu à somme nulle est un jeu où la somme des utilités de chaque joueur est nulle. Les gains d'un joueur correspondent alors aux pertes d'un autre joueur. A l'opposé, dans un jeu à somme non-nulle, certaines issues peuvent être profitables à l'ensemble des joueurs.
- les jeux à information complète / incomplète. Un jeu à information complète désigne un jeu où chaque agent connaît l'ensemble des actions possibles ainsi que les gains des autres joueurs en plus des siens. Sinon, il est dit à information incomplète.

Un jeu est défini par un ensemble de joueurs, leurs utilités respectives, le déroulement du jeu (si les joueurs jouent simultanément ou chacun leur tour, quel joueur joue en premier ...), et les choix (les actions) qui s'offrent aux joueurs en fonction de la situation. A partir de ces données, on peut alors représenter le jeu sous forme extensive. Un jeu sous forme extensive est un jeu représenté sous la forme d'un arbre où les noeuds représentent les choix parmi les actions possibles et les branches représentent une alternative (une action). À chaque noeud est associé le joueur qui doit jouer. Les gains de chaque joueur sont associés aux feuilles de l'arbre (noeuds teminaux). L'arbre de jeu décrit alors l'ensemble des scénarios pouvant se dérouler au cours d'une partie.

Voici un exemple de jeu :

Soit deux joueurs $J1$ et $J2$. Le jeu se déroule en deux tours. Le joueur $J1$ joue en premier puis le joueur $J2$ joue. Chaque joueur à le choix entre deux actions A et B. La figure 5.1 illustre la représentation du jeu sous forme extensive. Les utilités des joueurs sont indiquées dans les feuilles de l'arbre.

Raisonnement des joueurs : L'objectif de chaque joueur est de maximiser son utilité. Ainsi, dans notre exemple, si le joueur $J1$ joue A, alors le joueur $J2$ va jouer B de sorte que son gain soit égal à 1 (plutôt que 0) ; le gain du joueur $J1$ sera alors également égal à 1. Selon le même raisonnement, si le joueur $J1$ joue B, alors le joueur $J2$ va jouer A ; le gain du joueur $J1$

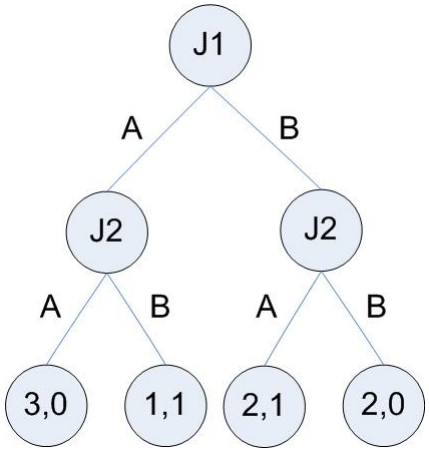


FIGURE 5.1 – Exemple de jeu sous forme extensive

sera alors égal à 2. Dès lors, si le joueur $J1$ suppose que le joueur $J2$ choisit sa meilleure réponse à la seconde étape, il choisit B à la première.

Pour prendre en compte le fait que nous ne pouvons pas prévoir le comportement qu'auront les partenaires au cours d'un échange B2B, nous proposons donc de modéliser ces échanges sous la forme d'un jeu. Dans un premier temps, nous avons considéré les interactions entre la plateforme et une entreprise. Nous avons ensuite défini le *jeu de médiation* comme l'union des jeux correspondant aux interactions entre chaque entreprise et la plateforme.

5.2 Échanges entre une entreprise et la plateforme : jeu à deux joueurs

Afin de modéliser les échanges B2B via la plateforme de médiation sous la forme d'un jeu, nous commençons par représenter les interactions entre une entreprise E et la plateforme sous la forme d'un jeu à deux joueurs. Chaque état du jeu correspond alors à l'état de l'échange dans le contexte de la sous-institution relative à E.

Joueurs : Les deux joueurs sont l'agent « entreprise » représentant E et l'agent « médiateur ».

Déroulement du jeu : Un joueur a le droit de jouer plusieurs fois de suite (lorsqu'il en a la possibilité). Un première approche pour représenter ceci est de considérer que les joueurs jouent chacun leur tour et que chaque joueur dispose d'une action « ne rien faire ». Ainsi, le choix de cette action à un tour du jeu permet au joueur du tour précédent de jouer une deuxième

fois de suite. Le problème de cette approche est de déterminer à quel moment un joueur peut estimer que l'autre joueur choisit de ne rien faire. La seconde approche, que nous avons adoptée, consiste alors à stipuler qu'à chaque tour, les deux joueurs sont susceptibles de jouer (en fonction de leurs actions possibles). Cependant, une étape d'un échange B2B se traduisant par l'envoi d'un document d'affaires, une seule action est considérée à chaque tour. Ainsi, un tour consiste à choisir une action parmi l'ensemble des actions possibles des deux joueurs.

Détermination des choix : Les actions possibles sont déterminées à partir des actions institutionnelles correspondant à la représentation des capacités d'interactions du système d'information de E, et de l'état courant de l'échange (dans le contexte de la sous-institution relative à E). Ainsi, l'ensemble des actions possibles à un tour du jeu correspond à l'ensemble des actions permises à l'état courant de l'échange.

Note : L'envoi d'un document d'affaires par l'agent « médiateur » est conditionné par le fait que le document ait été préalablement transmis à l'agent par l'entreprise à l'origine du document. Nous ne considérons pas cette précondition dans la détermination des choix. En effet, l'entreprise concernée ne faisant pas partie de la sous-institution relative à E (et donc des joueurs de ce jeu), nous ne prenons pas en compte ses actions.

Fin du jeu : Le jeu se termine lorsque plus aucune action n'est permise pour les joueurs. Cependant, dans certain cas, il se peut que le jeu boucle. Afin d'éviter les arbres de jeu infinis, nous déterminons alors les noeuds équivalents.

Définition. *Deux noeuds sont dits « équivalents » si les ensembles d'actions possibles à partir de ces noeuds sont les mêmes.*

Un noeud N_2 équivalent à un noeud N_1 est donc à l'origine du même sous-arbre que le noeud N_1. Par conséquent, afin de ne pas générer d'arbre infini, lorsqu'un nouveau noeud est équivalent à un de ses noeuds « père », il est considéré comme un noeud terminal.

Calcul de l'utilité des joueurs : L'objectif de notre approche est de garantir le bon déroulement des échanges pour l'entreprise en prévenant la survenue des exceptions (c'est-à-dire la violation des engagements). L'utilité des agents au terme de l'échange est donc calculée à partir des engagements courants. À chaque engagement est associé un gain en fonction de son état courant. Ainsi, par exemple, un agent aura un gain négatif pour un engagement violé. L'utilité d'un agent est alors égale à la somme des gains associés à chaque engagement. Nous

avons déterminé ces gains *a priori* pour chaque état. Leur valeur est présentée dans le tableau 5.1.

État de l'engagement	Gain associé
$pending$	-1
$active$	-2
$fulfilled$	$+1$
$violated$	-2
$cancelled$	-1

TABLE 5.1 – Correspondance entre l'état courant d'un engagement et le gain associé

Justification des gains : L'état $fulfilled$ se retrouve naturellement associé à un gain positif. A l'opposé, un gain négatif est associé à l'état *violated*. Notre priorité étant d'éviter la survenue d'exceptions, la valeur absolue associée à un engagement violé est supérieure à celle associée à un engagement satisfait. Dans le cas de l'état *active*, le gain est identique à celui associé à l'état *violated*. En effet, cette évaluation étant faite au terme de l'échange, un engagement actif ne pourra plus être satisfait. Il est donc considéré comme violé. Nous avons également associé un gain négatif à l'état *cancelled*. Ceci a pour but de représenter le fait qu'un partenaire n'a pas d'intérêt à s'engager dans des échanges B2B menant à l'annulation des engagements engendrés. Au contraire, cela entraînerait une perte de temps non souhaitable. Cependant, un engagement annulé n'entraînant pas d'exception, la valeur absolue de son gain est faible. Enfin, dans le cas de l'état *pending*, s'il reste un engagement en attente d'activation à la fin de l'échange, celui-ci ne pourra jamais être activé et est donc considéré comme annulé. Le gain associé est donc le même que pour l'état *cancelled*.

Ce jeu est un jeu non-coopératif à somme non-nulle et à information complète. En effet, les agents ne se concertent pas pour décider de leurs actions, ils ont la même utilité et connaissent chacun les possibilités de l'autre joueur (correspondant aux capacités d'interaction du système d'information de l'entreprise E). Son arbre de jeu modélise l'ensemble des scénarios d'échanges possibles avec le système d'information de E. Il correspond donc à la chorégraphie du système d'information de E.

5.2.1 Exemple illustratif :

Afin d'illustrer la formalisation des échanges entre un agent « entreprise » et l'agent « médiateur » sous la forme d'un jeu, nous représentons ici l'arbre de jeu correspondant à la chorégraphie présenté à la figure 5.2. Cette chorégraphie décrit le processus de commande-paiement-livraison d'une entreprise cliente. La figure 5.3 expose l'arbre de jeu. Ce dernier modélise les échanges entre l'agent « entreprise » Agt_E (représentant l'entreprise cliente) et l'agent « médiateur » Agt_M.

Nous pouvons remarquer qu'il correspond exactement à la chorégraphie.

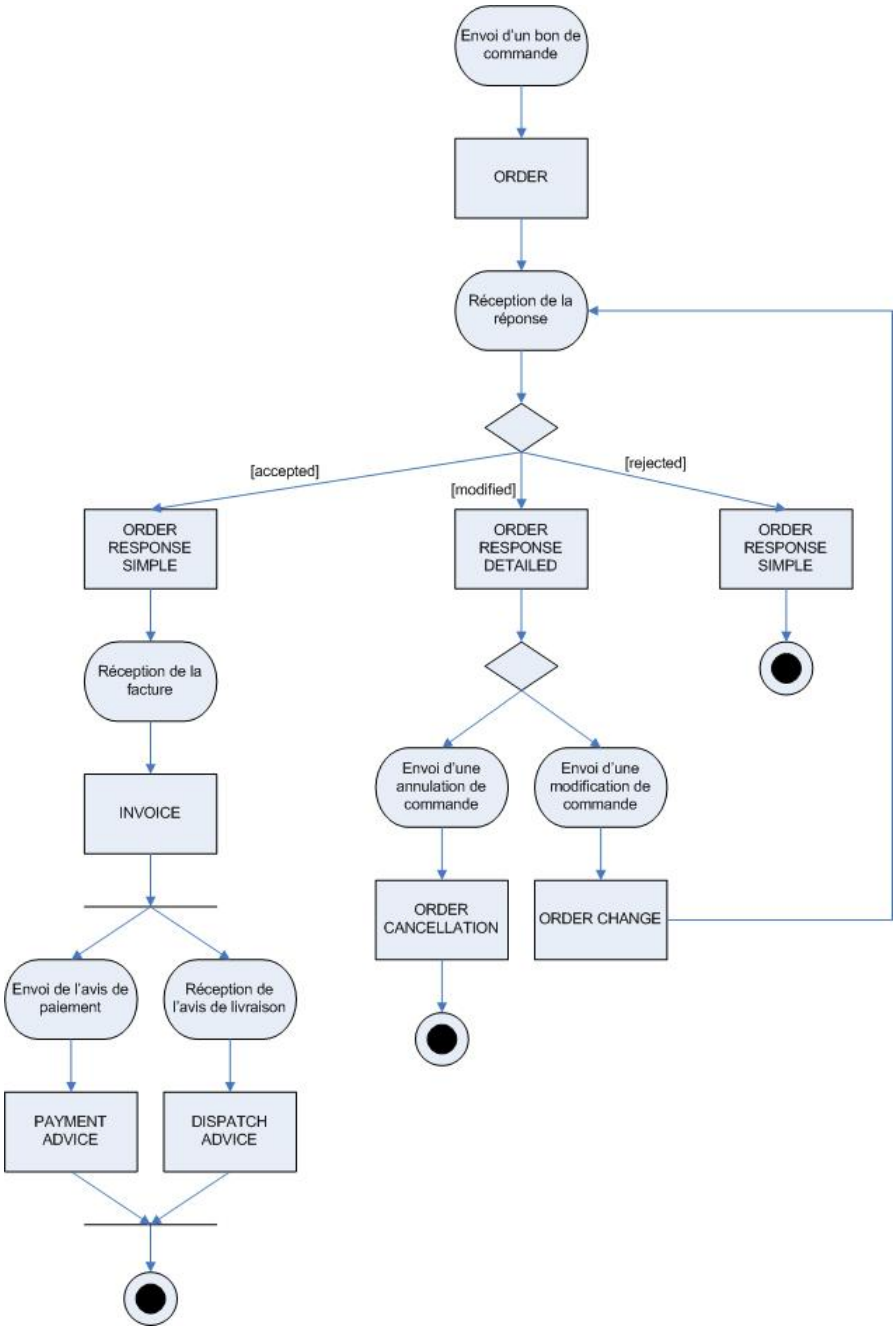


FIGURE 5.2 – Chorégraphie spécifiant un processus de commande-paiement-livraison

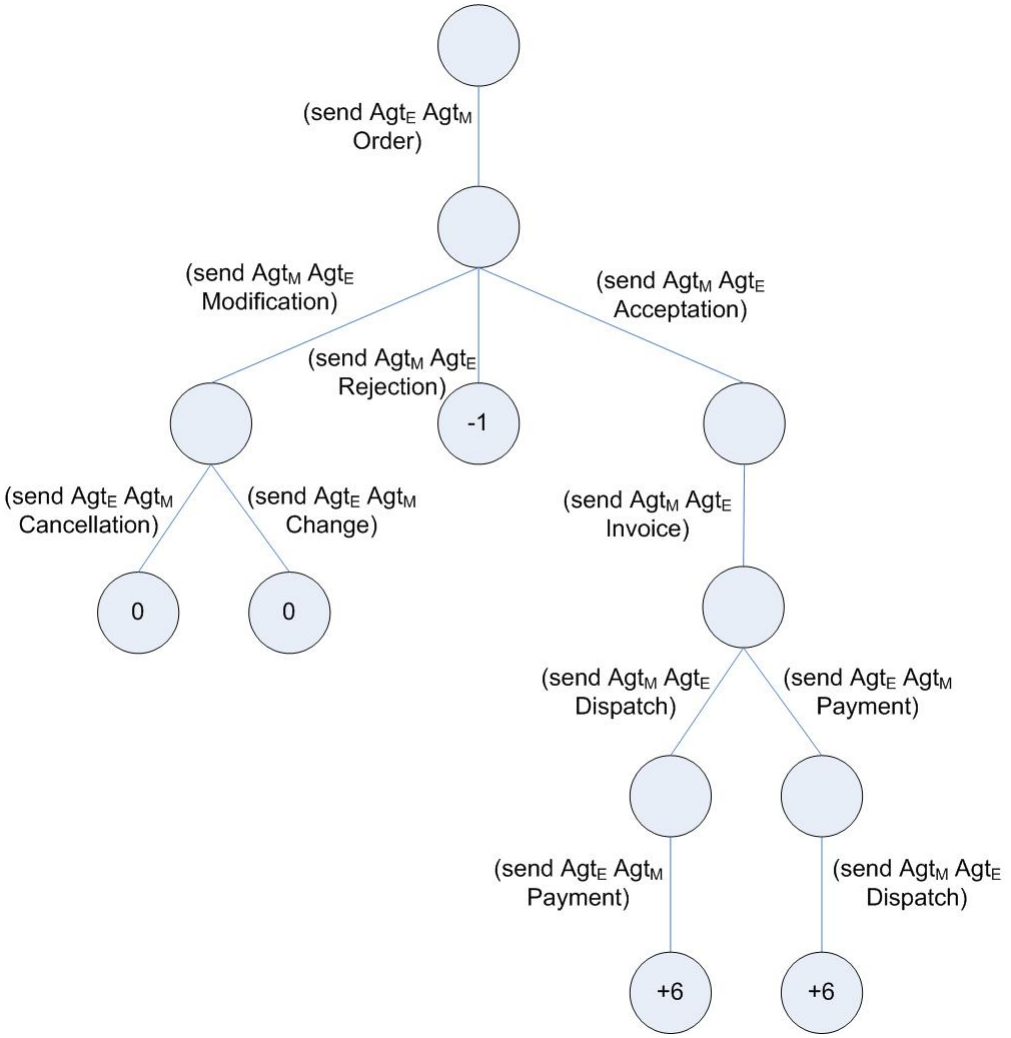


FIGURE 5.3 – Représentation des échanges entre un agent « entreprise » Agt_E et l'agent « médiateur » Agt_M par un jeu sous forme extensive

À partir de cette représentation des interactions entre chaque agent « entreprise » et l'agent « médiateur » sous la forme d'un jeu à deux joueurs, nous formalisons, dans la section suivante, les échanges entre les agents entreprises via la plateforme.

5.3 Échanges entre les entreprises via la plateforme : jeu de médiation

Les échanges entre les agents « entreprise », via la plateforme, sont représentés sous la forme d'un jeu à n joueurs où les joueurs sont l'agent « médiateur » et les agents « entreprise » partenaires. Nous appelons ce jeu un *jeu de médiation*. Il modélise l'union des jeux à deux joueurs

entre chaque agent « entreprise » et l'agent « médiateur ».

Joueurs : L'ensemble $\mathbb{J}$ des joueurs d'un jeu de médiation est :

$$\mathbb{J} = \{Agt_1,\ \ldots,\ Agt_{n-1},\ Agt_M\}$$

où les Agt_i représentent les agents « entreprise » et Agt_M représente l'agent « médiateur ».

Déroulement du jeu : Comme nous l'avons vu précédemment, chaque joueur est susceptible de jouer à chaque étape du jeu, mais une étape du jeu consiste uniquement en l'exécution d'une action par un joueur. Ainsi, à chaque étape, une action parmis celles possibles est choisie.

Détermination des choix : Les actions possibles à chaque étape sont déterminées par l'ensemble des actions institutionnelles des joueurs (définies selon notre modèle) dont les préconditions sont satisfaites.

L'ensemble $\mathbb{E}_{Agt}$ des actions possibles pour un agent Agt à une étape du jeu est déterminé par la formule suivante :

$$\mathbb{E}_{Agt} = \{??Act \mid ((power\ Agt\ ??Inst\ ??Cond\ ??Act\ ??Eff) \wedge ??Cond)\}$$

L'ensemble $\mathbb{E}$ des choix disponibles est alors :

$$\mathbb{E} = \bigcup_{Agt \in \mathbb{J}} \mathbb{E}_{Agt}$$

La fin du jeu : Le jeu est terminé lorsqu'il n'y a plus d'actions possibles, ou lorsque le nouveau noeud est *équivalent* à un de ses noeuds « père » (cf. section précédente).

Calcul de l'utilité des joueurs : L'utilité d'un agent « entreprise » est calculée à partir de l'état de la sous-institution relative à son entreprise. Ainsi, le point de vue des entreprises est considéré dans le calcul des utilités. Le rôle de l'agent dans les engagements est également pris en compte pour déterminer le gain associé. En effet, un partenaire B2B privilégiera toujours son intérêt personnel. Ainsi un agent « entreprise » créditeur d'un engagement, étant le bénéficiaire de cet engagement, a un gain plus important (en valeur absolue) que le débiteur. Nous fixons le

rapport entre les gains du créditeur et du débiteur à deux (les gains du débiteur correspondant aux gains déterminés dans la section précédente). Le tableau 5.2 présente ces gains.

État de l'engagement	Gain associé pour le créditeur	Gain associé pour le débiteur
pending	-2	-1
active	-4	-2
$fulfilled$	$+2$	$+1$
violated	-4	-2
cancelled	-2	-1

TABLE 5.2 – Correspondance entre l'état courant d'un engagement et le gain associé

Soit $\mathcal{E}_{N_t}$ un état terminal du jeu. L'utilité associée, U_{Agt_i}, d'un agent « entreprise » Agt_i est donc égale à la somme de ses gains dans $\mathcal{E}_{N_t}$. Soit $Inst_i$ la sous-institution relative à l'entreprise de Agt_i et $G_{Agt_i}(C)$ le gain de Agt_i pour l'engagement C selon le tableau 5.2 (par défaut, ce gain est égal à 0). On a :

$$U_{Agt_i} = \sum (G_{Agt_i}(C) \mid (D_{Inst_i}\ C) \in \mathcal{E}_{N_t})$$

Nous distinguons les utilités des agents « entreprise » et celle de l'agent « médiateur ». En effet, l'agent « médiateur » fait partie de l'ensemble des sous-institutions relatives à chaque entreprise. Son utilité correspond donc à la somme des utilités correspondant à ces sous-institution, c'est-à-dire des utilités des agents « entreprises ». Néanmoins, son objectif n'est pas uniquement de maximiser les utilités de tous les agents « entreprises ». Son rôle de médiation implique qu'il doit également condidérer l'équité entre ces utilités.

Définition. *Un scénario d'échange Sc_1 est dit plus équitable qu'un scénario Sc_2 si l'écart-type entre les utilités individuelles des partenaires B2B au terme de Sc_1 est inférieur à celui au terme de Sc_2. On note $Sc_1 >_{eq} Sc_2$.*

L'utilité de l'agent « médiateur » est alors définie par la somme des utilités des agents « entreprise » à laquelle on soustrait l'écart-type. Cette utilité est appelée *utilité de médiation.*

Définition. *Soit U_{Agt_i} l'utilité de l'agent entreprise Agt_i associé à un état terminal. Soit U_M l'utilité de médiation associée à ce même état. On a alors :*

$$U_M = \Big(\sum_{i=1}^{i=(n-1)} U_{Agt_i} \Big) - \sigma(U_{Agt_i}) = \Big(\sum_{i=1}^{i=(n-1)} U_{Agt_i} \Big) - \sqrt{\frac{\sum_{i=1}^{i=(n-1)} (U_{Agt_i} - \bar{U})^2}{(n-1)}}$$

où $\bar{U}$ est la moyenne des utilités des agents entreprise ($\bar{U} = \frac{\sum_{i=1}^{i=(n-1)} U_{Agt_i}}{(n-1)}$)

5.4 Arbre d'un jeu de médiation

Nous venons de définir ce qu'est un jeu de médiation. Nous allons maintenant présenter la construction de l'arbre de jeu d'un jeu de médiation et discuter sur l'utilisation de cet arbre pour déterminer le comportement de l'agent médiateur.

État courant du jeu de médiation : Chaque noeud de l'arbre de jeu modélise l'état du jeu de médiation à l'étape courante. Un noeud N est défini par un quadruplet ($\mathcal{E}_N$, $\mathbb{A}_N$, $V(N)$, $\mathbb{F}_N$) où $\mathcal{E}_N$ représente l'état courant de l'échange (c'est-à-dire l'état de l'institution globale, regroupant les états de chaque sous-institution), $\mathbb{A}_N$ l'ensemble des actions possibles à partir de N, $V(N)$ la valeur associée au noeud et $\mathbb{F}_N$ l'ensemble de ses noeuds fils. $\mathbb{F}_N$ est décrit par l'ensemble des couples (Act_i, N_i) où N_i est le noeud fils de N obtenu en exécutant l'action Act_i (avec $Act_i \in \mathbb{A}_N$).

Construction de l'arbre : Nous exposons ici la construction de l'arbre. Elle est décrite par les algorithmes 8, 9, 10.

Initialement, les paramètres de la méthode *GameTree* sont la racine de l'arbre représentée par le noeud $N_R = (\mathcal{E}_{N_R}, \mathbb{A}_{N_R}, V(N_R), \mathbb{F}_{N_R})$ et $\mathbb{P}_N = \emptyset$ (correspondant à l'ensemble des noeuds pères de N_R), avec $\mathcal{E}_{N_R}$ l'état initial de l'échange, $\mathbb{A}_{N_R} = \emptyset$, $V(N_R) = 0$ et $\mathbb{F}_{N_R} = \emptyset$.

5.5 Exemple illustratif

Nous présentons ici l'arbre de jeu d'un jeu de médiation représentant les échanges via la plateforme entre un client et un fournisseur au cours d'un processus de commande-paiement-livraison. Pour la chorégraphie de l'entreprise jouant le rôle de client, nous reprenons l'exemple présenté à la figure 5.2. La chorégraphie de l'entreprise jouant le rôle de fournisseur, quant à elle, est décrite par la figure 5.4. La figure 5.5 expose l'arbre de jeu correspondant. Cet arbre modélise les échanges entre les deux agents « entreprise » Agt_1 et Agt_2 et l'agent « médiateur » Agt_M, avec Agt_1 l'agent représentant le client et Agt_2 celui représentant le fournisseur.

Algorithme 8: Construction de l'arbre de jeu (méthode GameTree)

Data : - un noeud $N = (\mathcal{E}_N,\ \mathbb{A}_N,\ V(N),\ \mathbb{F}_N)$
- l'ensemble $\mathbb{P}_N$ des noeuds pères de N dans l'arbre
- l'ensemble $\mathbb{J}$ des joueurs

```
1  /* détermination des actions possibles */
2  A_N = GetChoices(E_N,J);
3
4  if A_N == ∅ then                                   /* N est une feuille */
5  |  break;
6  else
7  |  /* recherche si N est équivalent à un de ses pères */
8  |  forall the N_P = (E_{N_P}, A_{N_P}, V(N_P), F_{N_P}) ∈ P_N do
9  |  |  if A_N == A_{N_P} then                       /* N est une feuille */
10 |  |  |  break;
11 |  |  end
12 |  end
13 |  /* construction des noeuds fils */
14 |  forall the choice ∈ A_N do
15 |  |  N_F = ChildNode(N, choice);
16 |  |  ajouter (choice, N_F) à F_N;
17 |  |  P_{N_F} = {P_N, N};
18 |  |  GameTree(N_F, P_{N_F});
19 |  end
20 |  eval(N);
21 end
```

Algorithme 9: Détermination des actions possibles (méthode GetChoices)

Data : - l'état du noeud courant $\mathcal{E}$
- l'ensemble $\mathbb{J}$ des joueurs

Result : l'ensemble $\mathbb{A}$ des actions possibles dans l'état $\mathcal{E}$

```
1  A = ∅;
2  forall the Agt_i ∈ J do
3  |  forall the (power Agt_i ??Inst ??Cond ??Act ??Eff) ∈ E do
4  |  |  if ??Cond ∈ E then
5  |  |  |  ajouter ??Act à A;
6  |  |  end
7  |  end
8  end
9  return A;
```

Algorithme 10: Construction du noeud fils (méthode ChildNode)

Data : - un noeud $N = (\mathcal{E}_N,\ \mathbb{A}_N,\ V(N),\ \mathbb{F}_N)$
- une action (*action Agt Act*)

Result : le noeud fils N_1 de N obtenu en exécutant (*action Agt Act*)

/* suppression des actions permises */
$\mathcal{E}_{N_1} = \mathcal{E}_N - \{Act_i = (not\ (O\ (X_{??inst}\ (done\ ??Act\ true)))) \mid Act_i \in \mathcal{E}_N\}$;
)
/* interpretation de l'exécution de l'action */
forall the (*power Agt* $??Inst\ ??Cond_{Act}$ *Act* $??Eff_{Act}$) $\in \mathcal{E}_N$ **do**
  if $??Cond_{Act}$ **then**
    $(D_{??Inst}\ ??Eff_{Act})$ **and** $(D_{??Inst}\ (done\ (action\ Agt\ Act)\ true))$;
  end
end

/* mise à jour de l'état des engagements courants */
forall the (*commitment* ??*state* ??*debtor* ??*creditor* ??*content* ??*condition*) $\in \mathcal{E}_N$ **do**
  (cf. algorithme 3)
end

/* initialisation de $\mathbb{A}_{N_1}$, $V(N_1)$ et $\mathbb{F}_{N_1}$ /*
$\mathbb{A}_{N_1} = \emptyset$; $V(N_1) = 0$; $\mathbb{F}_{N_1} = \emptyset$;

return $N_1 = (\mathcal{E}_{N_1},\ \mathbb{A}_{N_1},\ V(N_1),\ \mathbb{F}_{N_1})$;

5.6 Discussion

Dans cette section, nous avons défini ce que nous avons appelé un jeu de médiation. Un jeu de médiation permet de représenter les échanges entre les partenaires B2B via la plateforme de médiation sous la forme d'un arbre de jeu. Cet arbre de jeu permet de calculer, au préalable, l'ensemble des scénarios pouvant se dérouler au cours d'échanges entre les systèmes d'information des partenaires (selon les spécifications de leurs chorégraphies).

La modélisation des échanges B2B par un jeu de médiation permet donc de raisonner sur l'ensemble des scénarios possibles entre les partenaires. L'utilité de médiation évalue ces scénarios afin de guider le comportement de l'agent médiateur. Le but de l'agent « médiateur » est alors de faire les meilleurs choix en anticipant les comportements des partenaires menant à une utilité de médiation faible. Cependant, cette utilité n'est calculée qu'aux feuilles de l'arbre. Afin de déterminer le comportement de l'agent « médiateur », nous devons donc définir son raisonnement par rapport à l'arbre de jeu.

Raisonnement de l'agent « médiateur » : Le raisonnement de l'agent « médiateur » consiste

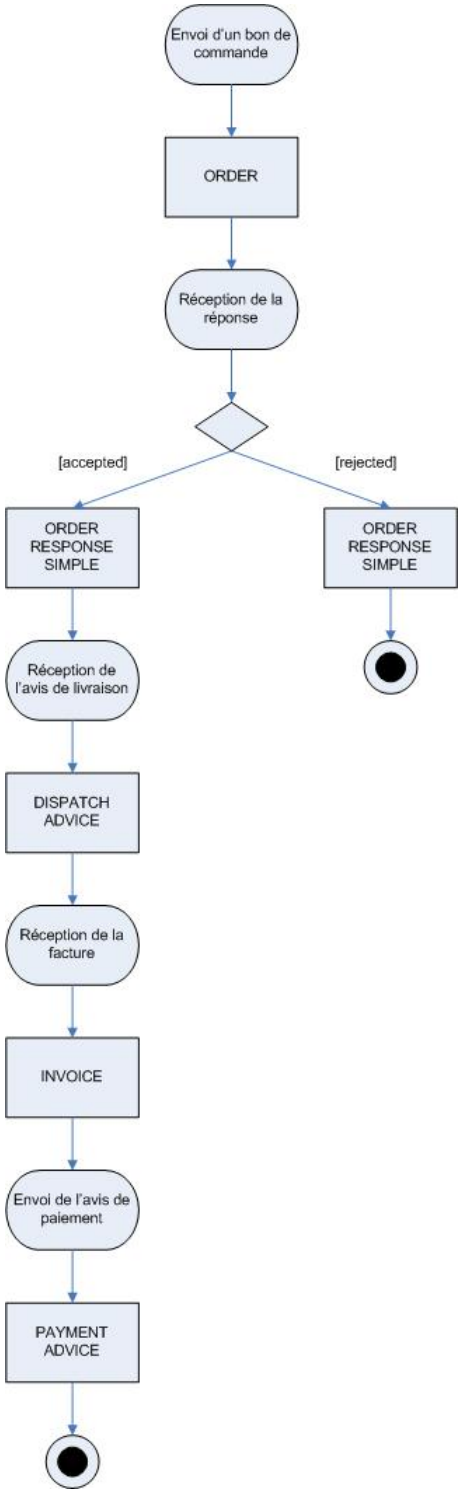


FIGURE 5.4 – Chorégraphie spécifiant un processus de commande-paiement-livraison

à déterminer ses choix au cours du jeu. Son objectif est de maximiser l'utilité de médiation. Afin de déterminer les meilleurs choix à chaque étape du jeu, nous devons donc évaluer l'ensemble des noeuds de l'arbre. La valeur associée à un noeud est appelée *utilité espérée*.

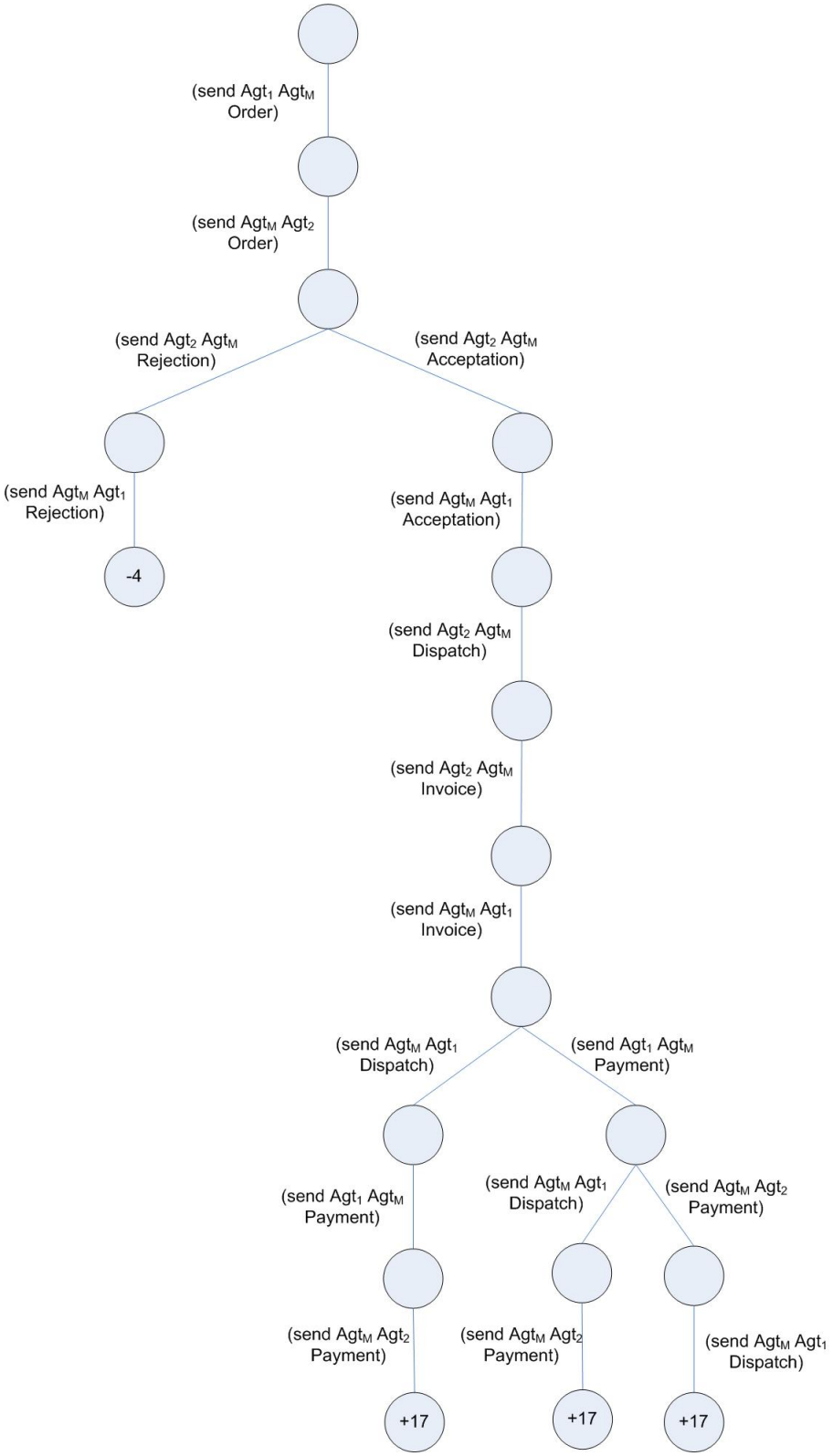


FIGURE 5.5 – Exemple d'un arbre de jeu d'un jeu de médiation

La particularité de l'arbre de jeu est, qu'à chaque étape, le joueur n'est pas fixé. En effet, les actions possibles à partir de chaque noeud de l'arbre sont potentiellement des actions de joueurs différents. Cette caractéristique du jeu de médiation empêche de calculer le meilleur choix pour l'agent « médiateur » de manière déterministe. Nous envisageons donc une approche probabiliste.

Malheureusement, nous n'avons pas eu le temps d'étudier ce problème de manière approfondie. Nous présentons néanmoins ci-dessous nos pistes de réflexion sur le calcul de l'utilité espérée associée à un noeud de l'abre.

Soit N un noeud de l'arbre. Soit U_M l'utilité espérée maximum de l'agent « médiateur » (parmi celles associées aux noeuds obtenus à partir de N en choisissant une action de l'agent « médiateur »). En d'autres termes, U_M est l'utilité espérée maximum si l'agent « médiateur » joue. L'idée est de déterminer l'utilité espérée de l'agent « médiateur » associée à N. Cette utilité dépend non seulement de U_M, mais également de la probabilité qu'une action d'un agent « entreprise » menant à un noeud d'utilité espérée supérieure à U_M se produise. Cette probabilité pourra alors être utilisée pour déterminer si l'agent « médiateur » doit agir ou non. Intuitivement, l'utilité espérée de l'agent « médiateur » associée à un noeud dépend de la probabilité qu'un agent « entreprise » exécute une meilleure action que la meilleure action de l'agent « médiateur ». Si cette probabilité est faible, l'agent « médiateur » aura alors intérêt à jouer et l'utilité espérée associée sera alors importante. Nous devons donc définir la probabilité qu'un agent « entreprise » exécute une certaine action. Une méthode possible de calcul consiste à considérer que cette probabilité dépend de l'utilité espérée individuelle de l'agent « entreprise ». En d'autres termes, un agent « entreprise » aura une pobabilité plus importante d'exécuter l'action A par rapport à l'action B si son utilité espérée associée au noeud obtenu en exécutant A est supérieure à celle associée au noeud obtenu en exécutant B.

Après avoir présenté le principe d'un jeu de médiation, nous définissons dans la section suivante le concept d'interopérabilité de médiation. Ce concept nous permet de déterminer l'interopérabilité entre les systèmes d'information de partenaires B2B en considérant les échanges via la plateforme de médiation.

6

Interopérabilité entre deux systèmes d'informations

L'interopérabilité entre deux systèmes d'information désigne leur capacité à intéragir entre eux. En particulier, deux systèmes d'information sont interopérables au niveau des processus si leurs chorégraphies définissent le même ordonnancement des échanges des documents d'affaires. Deux entreprises ayant des systèmes d'information interopérables peuvent donc s'engager dans des échanges B2B sans risquer d'engendrer d'exceptions dues à un problème de compatibilité entre leurs chorégraphies. Le but de ce chapître est d'adapter cette notion d'interopérabilité au cas d'échanges B2B s'effectuant via notre plateforme de médiation.

6.1 Interopérabilité constitutive

Dans [13], Chopra et Singh ont défini la notion d'interopérabilité constitutive. Ce concept d'interopérabilité est basé sur la notion d'engagement social. Il exprime l'interopérabilité entre les agents d'un système multi-agents par rapport aux engagements engendrés au cours des interactions entre ces agents. Pour cela, ils considèrent l'interprétation des interactions selon chaque agent. Soit $\mathbb{O} = [O_x,\ O_y]$ une séquence d'interactions où O_x et O_y sont respectivement la séquence des interactions observées par l'agent x et celle observée par l'agent y. $O_x \models_x C(x,\ y,\ p,\ q)$ signifie que la séquence d'interactions observée par x implique l'engagement $C(x,\ y,\ p,\ q)$ selon son interprétation propre. L'interopérabilité constitutive d'un système multi-agents $\mathcal{A}$ signifie que pour toutes les séquences d'interactions possibles ($\mathcal{O}_\mathcal{A}$), si un agent interprète un engagement dont il est le créditeur alors le débiteur de l'engagement interprète le même engagement.

On dit alors que le système multi-agents est *C-interopérable.*

Définition. $\mathcal{A}$ *est* C-interopérable *ssi*

$$\forall \mathbb{O} \in \mathcal{O}_{\mathcal{A}} \ : \ (\forall x, y \ : \ [O_x, \ O_y] \in \mathbb{O} \ : \\ (O_y \models_y C(x, \ y, \ p, \ q)) \rightarrow (O_x \models_x C(x, \ y, \ p, \ q)))$$

Nous pouvons transposer ce concept dans le domaine du B2B. Nous avons défini la notion d'interopérabilité constitutive entre deux systèmes d'information.

Définition. *Les systèmes d'information de deux partenaires B2B sont* C-interopérables *si dans l'ensemble des scénarios d'échanges possibles entre eux selon les spécifications de leurs chorégraphies, l'interprétation par un partenaire d'un engagement métier dont il est le créditeur implique l'interprétation par l'autre partenaire du même engagement (dont il est le débiteur).*

Nous avons formalisé cette définition dans le cadre de notre plateforme de médiation. Cette plateforme représente les échanges entre les systèmes d'information des partenaires (représentés par les agents « entreprise ») via l'agent « médiateur ». Nous devons donc prendre en compte les actions de l'agents médiateur. Par conséquent, l'interopérabilité entre les systèmes d'information des entreprises dans le cadre de la plateforme de médiation ne se réduit par à leur interopérabilité native mais considère également les capacités de médiation de la plateforme. Nous avons appelé cette interopérabilité l'*interopérabilité de médiation.*

6.2 Interopérabilité de médiation

L'interopérabilité de médiation définit l'interopérabilité constitutive entre deux systèmes d'information, moyennant les capacités de médiation de la plateforme. On parle alors de deux systèmes d'information *M-interopérables.*

Pour définir cette interopérabilité, nous construisons l'arbre de jeu du jeu de médiation correspondant aux échanges entre les deux partenaires. Cet arbre représente l'ensemble des scénarios d'échanges possibles entre ces partenaires. L'*interopérabilité de médiation* entre les partenaires est déterminée au noeud racine de l'arbre.

Définition. *Soit X et Y deux partenaires B2B, Agt_X et Agt_Y les agents « entreprise » représentant respectivement les entreprises X et Y et I_X et I_Y les sous-institution relative aux entreprises X et Y. Soit $\mathcal{A}$ l'arbre de jeu du jeu de médiation correspondant aux échanges entre Agt_x et Agt_y.*

Les systèmes d'information de X et Y sont dits M-interopérables *ssi la racine de $\mathcal{A}$ est un noeud* M-interopérable*.*

Les définitions suivantes permettent de déterminer si un noeud N_i de $\mathcal{A}$ est *M-interopérable.* Nous considérons deux cas :

1. N_i est une feuille de l'arbre $\mathcal{A}$:

   Définition. *Une feuille N_i de $\mathcal{A}$ est* M-interopérable *ssi :*

   $$((D_{I_X}\ (commitment\ Y\ X\ P\ Q)) \in N_i) \rightarrow ((D_{I_Y}\ (commitment\ Y\ X\ P\ Q)) \in N_i)$$
   $$\wedge\ ((D_{I_Y}\ (commitment\ X\ Y\ P\ Q)) \in N_i) \rightarrow ((D_{I_X}\ (commitment\ X\ Y\ P\ Q)) \in N_i)$$

2. N_i est un noeud interne de l'arbre $\mathcal{A}$. Dans ce cas on a :

   Définition. *Soit $\mathbb{N}_E$ l'ensemble des noeuds fils de N_i obtenus à la suite d'une action d'un agent « entreprise ». Soit $\mathbb{N}_M$ l'ensemble des noeuds fils de N_i obtenus à la suite d'une action de l'agent « médiateur ». N_i est* M-interoperable *ssi :*

   $$(\forall N_j \in \mathbb{N}_E,\ N_j\ est\ \text{M-interoperable}) \wedge (\exists N_k \in \mathbb{N}_M \mid N_k\ est\ \text{M-interopérable})$$

Cette définition récursive de la notion d'interopérabilité de médiation prend donc en compte les choix de l'agent « médiateur ». De manière intuitive, deux systèmes d'information sont *M-interopérables* si leur *C-interopérabilité* ne dépend que des choix de l'agent « médiateur ».

À partir de cette définition, nous pouvons donc déterminer l'interopérabilité de médiation entre deux systèmes d'information lors de la construction de l'arbre de jeu du jeu de médiation correspondant aux échanges entres ces systèmes d'information. Le calcul de l'interopérabilité de médiation permet ainsi de déterminer si les capacités de médiation de la plateforme peuvent permettre à deux futurs partenaires ayant des systèmes d'information éventuellemment incompatibles de mener à bien des échanges B2B.

Synthèse

Dans cette partie, nous avons présenté notre approche destinée à permettre à des entreprises de s'engager dans des échanges B2B sans avoir préalablement besoin de s'accorder sur une chorégraphie commune.

Nous avons conçu une plateforme multi-agents au sein de laquelle les capacités d'interactions des systèmes d'information des entreprises (décrites par leur chorégraphie) sont représentées. Cette représentation est basée sur la notion d'engagement social. Cette notion nous permet de modéliser les exceptions et de les détecter automatiquement (par la violation d'un engagement). Notre plateforme est conçue dans un cadre institutionnelle afin de prendre en compte le point de vue des entreprises. Elle est composé d'agents « entreprise » représentant le comportement des entreprises et d'un agent « médiateur ». Le comportement de la plateforme est spécifié en fonction des engagements engendrés au cours des échanges. Il détermine les actions de l'agent « médiateur » afin de satisfaire l'ensemble de ses engagements. Nous avons également défini le concept de *jeu de médiation* afin de pouvoir calculer, au préalable, les scénarios d'échanges possibles entre des partenaires B2B selon les spécifications de leurs chorégraphies. La construction de l'arbre de jeu correspondant permet ainsi de déterminer les éventuelles exceptions pouvant survenir et d'anticiper le comportement des entreprises afin d'éviter les situations menant à une exception. Nous avons enfin défini la notion d'interopérabilité de médiation. Cette notion, définie à partir de la formalisation des échanges entre des partenaires sous la forme d'un jeu de médiation, détermine l'interopérabilité entre les systèmes d'information des partenaires en considérant le comportement de la plateforme de médiation.

Dans la partie suivante, nous présentons les détails sur l'implémentation de la plateforme de médiation. En particulier, nous exposons le fonctionnement de la plateforme de développement multi-agents JIA (Jade Institutionnal Add-on), que nous avons utilisée pour nos développements, puis nous décrivons comment nous l'avons personnalisée pour mettre en oeuvre nos travaux de recherche.

Troisième partie

Implémentation

7

La plateforme de développement multi-agents JIA

Dans cette partie nous détaillons l'implémentation de notre plateforme de médiation. Nous nous sommes basés sur la plateforme de développement multi-agents JIA (JSA Institutional Add-on). JIA est une extension de la plateforme de développement multi-agents JADE basée sur l'extension JSA (JADE Semantics Add-on). Ce chapitre présente les caractéristiques de cette plateforme

7.1 JADE

JADE (Java Agent Development Framework, [5]) est une plateforme de développement de systèmes multi-agents. C'est un logiciel libre permettant l'implémentation d'agents communicants capables d'échanger des messages selon le standard FIPA-ACL [25]. JADE fournit une plateforme de déploiement pour les agents ainsi qu'une interface de programmation pour le développement des agents en Java. Cette plateforme dispose de nombreuses extensions. En particulier, l'extension JSA (JADE Semantics Add-on, [40]), fournit une interface de programmation permettant de développer des agents capables de raisonner sur la sémantique des messages échangés.

7.2 JSA

L'extension JSA propose un moteur de règles, conforme avec la logique BDI [52], utilisé pour formaliser la sémantique des actes de langage de la norme FIPA-ACL [25]. Elle est basée sur la logique FIPA-SL (cf. Section II). Elle fournit un ensemble de règles d'interprétation de base représentant la dimension intentionnelle des actes de langage FIPA-ACL. Ainsi, un agent développé avec JSA, appelé *agent sémantique*, est capale d'interpréter la sémantique des messages échangés selon les spécifications de la norme FIPA-ACL. Par exemple, un agent j recevant un message FIPA-ACL « $Inform$ » ayant comme contenu la proposition p de la part d'un autre agent i interprète le fait que i croit la proposition p et qu'il a l'intention que j la croit aussi. Par la suite d'autres règles mèneront j à croire p. Cependant, ce comportement peut être personnaliser par le développeur lors de l'implémentation des *agents sémantiques* grâce à la création de nouvelles règles d'interprétation.

7.2.1 Agents sémantiques

Chaque *agent sémantique* dispose de capacités sémantiques. Ces capacités sont constituée d'une base de connaissance, d'actions sémantiques et de principes d'interprétation sémantique (SIP). La figure 7.1 montre la structure d'un *agent sémantique*.

Base de connaissance

La base de connaissance d'un *agent sémantique* représente l'état du monde selon l'agent. Elle est composée de formules, exprimées dans la logique FIPA-SL, décrivant ses croyances et ses intentions.

Une base de connaissance est équipée de plusieurs mécanismes :

- les *observeurs* : un *observeur* surveille une formule spécifique et réagit si sa valeur change dans la base de connaissance (en déclenchant le traitement correspondant) ;
- les *filtres* : un filtre permet de personnaliser la sauvegarde des formules dans la base de connaissance (par exemple, un agent manipulant des images peut avoir un filtre permettant de sauvegarder les images dans un dossier externe pour éviter de surcharger la base de connaissance).

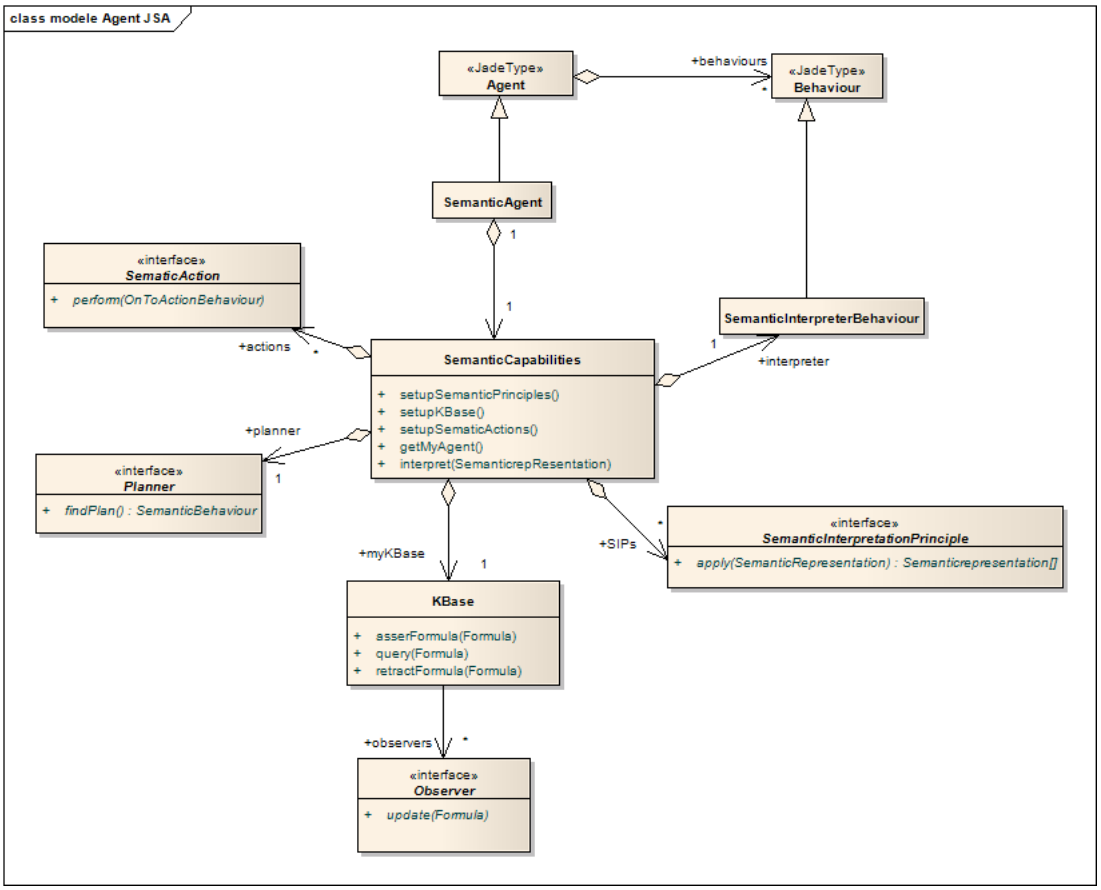


FIGURE 7.1 – Diagramme UML d'un agent JSA

Actions sémantiques

Chaque *agent sémantique* dispose d'une table d'actions sémantiques décrivant les actions connues par l'agent, avec leurs préconditions et leurs effets sémantiques (exprimés dans la logique FIPA-SL). Cette table contient par défaut l'ensemble des actes de langage définis par la norme FIPA-ACL (avec les préconditions et les effets correspondant aux spécifications de FIPA-ACL). À chaque action est associé un comportement sémantique permettant de lancer son exécution.

Principes d'interprétation sémantique

Chaque *agent sémantique* dispose également d'une table de principes d'interprétation sémantique. Ces principes représentent les règles de raisonnement de l'agent. Ils sont exécutés suite à la perception d'un évènement (par exemple, la réception d'un message FIPA-ACL).

Chaque évènement perçu est représenté par une *représentation sémantique* (SR). Un SR est

un objet encapsulant un formule (exprimée en FIPA-SL) représentant l'évènement, complétée par un index d'interprétation utilisé au cours du mécanisme d'interprétation (cf. ci-dessous).

7.2.2 Mécanisme d'interprétation

Les principes d'interprétation sémantique sont au coeur du mécanisme d'interprétation de JSA. La figure 7.2 décrit ce mécanisme.

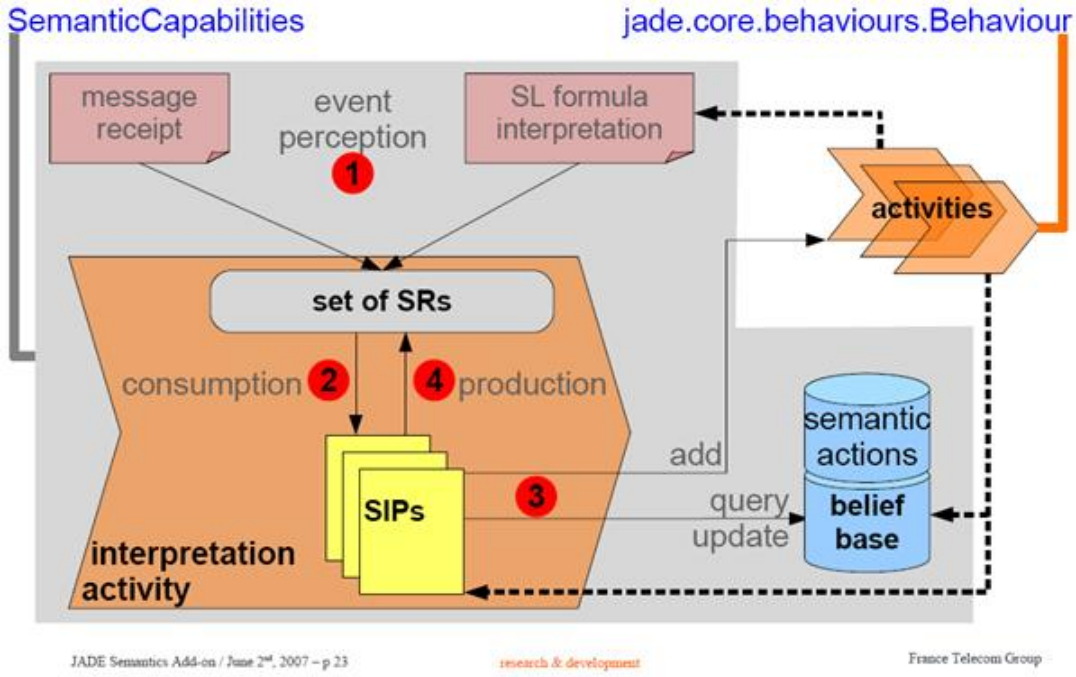


FIGURE 7.2 – Mécanisme d'interprétation de la plateforme JSA

Description du mécanisme d'interprétation de JSA

Suite à la perception d'un nouvel évènement, une SR est générée. Cette SR est ensuite successivement interprétée par les SIP de l'agent (selon son index d'interprétation).

Chaque SIP possède un type de formule (*pattern* en anglais). Il est exécuté uniquement si la SR correspond à son type de formule. Le cas échéant, le SIP consomme la SR, modifie éventuellement la base de connaissance de l'agent (en ajoutant de nouvelles croyances ...), peut également exécuter un ensemble de comportements propres à l'agent, et retourne un ensemble de nouvelles SR (pouvant être vide ou contenir la SR d'origine avec un nouvel index d'interprétation).

L'interprétation se termine lorsque toutes les SR ont été interprétées.

Interprétation des intentions

Les SIP par défaut fournis avec JSA permettent aux agents d'agir en fonction de leurs intentions. Les deux SIP présentés ci-dessous permettent de planifier les actions à faire.

- *Rationality Principle SIP* : Ce SIP interprète l'intention qu'une proposition soit vraie. Il va rechercher dans la table des actions sémantiques une action dont les effets correspondent à la proposition. Si une telle action existe, le SIP retourne l'intention qu'elle soit éxécutée. Sinon l'intention est considérée insatisfaisable.
- *Action Performance SIP* : Ce SIP interprète l'intention qu'une action soit exécutée par l'agent et lance le comportement associé à l'action.

Cette extension a été à son tour étendue afin de considérer la dimension institutionnelle des actions. C'est cette nouvelle extension, appelée JIA (JSA Institutional Add-on), que nous avons utilisé pour développer notre prototype.

7.3 JIA

JIA (JSA Institutional Add-on, [72]) est une extension de JSA fournissant un ensemble de règles d'interprétation additionnelles permettant de prendre en compte la dimension institutionnelle des actions des agents de la plateforme. Ces agents, appelés *agents institutionnels*, sont des *agents sémantiques* évoluant dans un contexte institutionnel.

JIA est basée sur la logique xSL, qui est une extension de la logique FIPA-SL à laquelle ont été intégrés les opérateurs déontiques et institutionnels définis par Demolombe et Louis dans [17] (cf. Section 3.1). Ainsi la base de connaissance d'un *agent institutionnel* est composée d'un ensemble de formules exprimées dans la logique xSL. Ces formules décrivent l'état du monde en prenant en compte les aspects institutionnels (faits institutionnels, pouvoirs institutionnels ...).

JIA définit différents types d'agents. Les *agents institutionnels* standards sont des agents membres d'au moins une institution. Ils interprètent la dimension institutionnelle des actions relatives aux institutions auxquelles ils appartiennent. À chaque institution est associé un unique agent « institution ». Cet agent incarne l'état de l'institution. Il a connaissance de l'ensemble des règles de l'institution. Il observe les actions de la plateforme afin de mettre à jour sa base

de connaissance (représentant l'état de l'institution) en fonction des ces règles.

7.3.1 Capacités institutionnelles

Un *agent institutionnel* dispose de *capacités institutionnelles*. Ces capacités sont une extension des *capacités sémantiques* de JSA, incluant le comportement de base des *agents sémantiques*. Elles décrivent les règles d'interprétation permettant de prendre en compte la dimension institutionnelle des actions. Le mécanisme d'interprétation est celui de JSA.

Les *capacités institutionnelles* enrichissent les *capacités sémantiques* décrites dans la section précédente en étendant ces dernières avec une liste de SIP spécifiques. Ces SIP permettent l'interprétation institutionnelle des actions. On parle alors d'actions institutionnelles.

7.3.2 Spécification des actions institutionnelles

Afin de considérer la dimension institutionnelle d'une action, celle-ci doit être spécifiée. Les caractéristiques institutionnelles des actions sont ainsi spécifiées en xSL dans un fichier de configuration.

Une action institutionnelle est composé de 5 caractéristiques :

- le nom de l'institution concernée ;
- le modèle de l'action (*pattern*) ;
- la liste des agents spectateurs (qui devront être informés de chaque exécution de l'action) ;
- la précondition institutionnelle de l'action ;
- l'effet institutionnel de l'action.

Un SIP *Action Declaration* permet d'interpréter ce fichier de configuration. Plus précisément, ce fichier est lu lors de la création des agents, et les spécifications de chaque action sont interprétées par le SIP *Action Declaration*. Chaque action est alors enregistrée dans la base de connaissance de l'agent sous la forme d'un pouvoir institutionnel. Ce pouvoir exprime le fait que l'exécution de cette action implique l'effet institutionnel, dans le contexte de l'institution concernée, si la précondition institutionnelle est satisfaite (cf. Section 3.1).

7.3.3 Interprération de l'exécution d'une action institutionnelle

Lorsqu'un agent perçoit qu'une action est exécutée, cet évènement est d'abord interprété par le SIP *Institutional Action Done*. Ce SIP vérifie alors si cette action correspond à un pouvoir institutionnel (c'est-à-dire s'il s'agit d'une action institutionnelle). Si tel est le cas, il vérifie sa précondition institutionnelle et enregistre l'effet institutionnel dans la base de connaissance de l'agent, si la précondition est satisfaite.

JIA fournit également des SIP gérant l'interprétation des obligations. Nos travaux étant basés sur les engagements sociaux, nous ne les détaillons pas.

Dans le chapitre suivant, nous décrivons la mise en oeuvre des travaux, présentés dans la partie II, avec la plateforme JIA.

8

Mise en oeuvre du prototype

Nous avons implémenté un prototype de notre plateforme de médiation multi-agents à partir de la plateforme de développement JIA. Pour rappel, la figure 8.1 décrit l'architecture institutionnelle de notre plateforme de médiation.

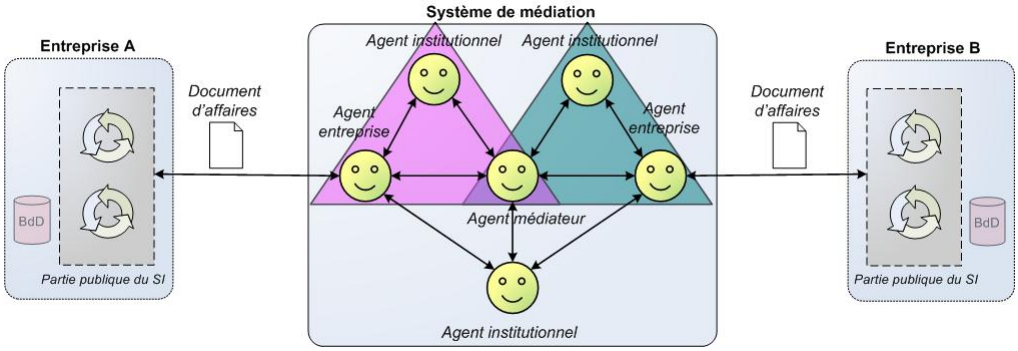


FIGURE 8.1 – Architecture institutionnelle de la plateforme de médiation

Le prototype est composé de trois agents standards. L'un correspond à l'agent « médiateur » et les deux autres correspondent aux agents « entreprise » représentant deux partenaires. Il est également composé de trois institutions symbolisées par trois agents « institution », l'un associé à l'institution globale et les deux autres associés chacun à une sous-institution relative à un partenaire. Les agents « entreprise » appartiennent chacun à deux institutions, celle relative au partenaire qu'ils représentent et l'institution globale. L'agent « médiateur » quant à lui, appartient aux trois institutions.

Nous devons maintenant définir les actions institutionnelles de chaque agent. Ces actions sont spécifiées dans le fichier de configuration. Notre représentation des actions institutionnelles étant basée sur les engagements sociaux, nous avons dû tout d'abord étendre la plateforme JIA

afin qu'elle implémente le langage xSL^{com} défini à la section 3.2.1.

8.1 Extension de la plateforme

8.1.1 Extension de la grammaire

Nous avons étendu la grammaire décrivant le langage xSL en introduisant l'opérateur « commitment ». Cette nouvelle grammaire décrit le langage xSL^{com}. Ce dernier définit un engagement par l'expression SL « (*commitment State Debtor Creditor Content Condition*) » avec :

- *State* : Terme SL pouvant prendre les valeurs suivantes : *PENDING*, *ACTIVE*, *FULFILLED*, *VIOLATED*, *CANCELLED*. Il représente l'état dans lequel se trouve l'engagement ;
- *Debtor* et *Creditor* : Termes SL représentant respectivement l'agent débiteur et l'agent créditeur de l'engagement.
- *Content* et *Condition* : Formules SL représentant l'objet et la condition de l'engagement.

8.1.2 Gestion du cycle de vie d'un engagement

Un SIP spécifique a été implémenté pour gérer automatiquement le cycle de vie d'un engagement. Ce SIP interprète une SR représentant un engagement. Ce SIP met en place des observeurs implémentant les règles de transition d'un engagement. Ces règles sont rappelées ci-dessous :

Règle 1 : (*commitment active debtor creditor content condition*) $\wedge$ *content*
$\rightarrow$ (*commitment fulfilled debtor creditor content condition*)

Règle 2 : (*commitment active debtor creditor content condition*) $\wedge$ (*not content*)
$\rightarrow$ (*commitment violated debtor creditor content condition*)

Règle 3 : (*commitment pending debtor creditor content condition*) $\wedge$ *condition*
$\rightarrow$ (*commitment active debtor creditor content condition*)

Règle 4 : (*commitment pending debtor creditor content condition*) $\wedge$ (*not condition*)
$\rightarrow$ (*commitment cancelled debtor creditor content condition*)

Règle 6 : (*commitment pending debtor creditor content condition*) $\wedge$ *content*
$\rightarrow$ (*commitment fulfilled debtor creditor content condition*)

Fonctionnement du SIP : A chaque changement d'état d'un engagement, une SR est créée et interprétée. Ainsi, si l'état de l'engagement est *PENDING*, le SIP met en place deux observeurs surveillant l'objet et la condition de l'engagement. Le premier passe l'engagement dans l'état *FULFILLED* si l'objet devient vrai. Le second passe l'engagement dans l'état *ACTIVE* si la condition devient vraie et dans l'état *CANCELLED* si elle devient fausse. S'il l'état de l'engagement est *ACTIVE*, le SIP ne met en place qu'un observeur surveillant l'objet de l'engagement. Il passe l'engagement dans l'état *FULFILLED* si l'objet est satisfait et dans l'état *VIOLATED* s'il devient faux.

Une fois cette extension réalisée, nous pouvons alors spécifier les actions institutionnelles représentant les capacités d'interaction des systèmes d'information des partenaires.

8.2 Spécification des capacités d'interaction

Les actions institutionnelles sont définies dans deux fichiers de configuration distincts, un pour chaque sous-institution. Les actions institutionnelles représentant les capacités d'interaction d'un partenaire sont spécifiées dans le fichier correspondant à la sous-institution relative à ce partenaire. Ce fichier définit les actions de l'agent entreprise représentant le partenaire. Ces actions correspondent aux envois de documents par le système d'information du partenaire. Il définit également les actions du médiateur correspondant aux documents pouvant être envoyés à l'agent « entreprise », et donc au système d'information du partenaire. L'exemple d'un fichier est exposé à la figure 8.2.

8.3 Comportement de la plateforme

8.3.1 Comportement de l'agent « médiateur »

Les règles de raisonnement spécifiant le comportement de l'agent « médiateur » (cf. Section 4.3) ont été implémentée sous la forme de SIP.

Nous avons créé les trois SIP présentés ci-dessous :

– $InstitutionalActionIntentionSIP$: Ce SIP implémente les règles R_i 1 et R_i 2 (cf. Section 4.3.1). Il interprète les SR représentant l'intention que l'agent est réalisé une action dans le contexte d'une institution. Il implémente le premier cas de l'algorithme 7 (lignes 3 à 11).

```
### REQUEST CATALOGUE
(institutional-action b2b
        (REQUEST_CATALOGUE :customer ??_actor :supplier ??_supplier :document (document
            CATALOGUE_REQUEST ??_actor ??_supplier))
        (set (agent-identifier :name mediator@test))
        (fact (and (D b2b (not (O (not (REQUEST_CATALOGUE :customer ??_customer :supplier
             ??_supplier :document (document CATALOGUE_REQUEST ??_actor ??_supplier))))))
                (representative ??_actor SI1)
                (representative ??_supplier SI2)
               )
        )
        (fact (and (not (O (not (SEND_CATALOGUE :supplier ??_supplier :customer ??_actor
            :document (document CATALOGUE ??_supplier ??_actor)))))
                (document CATALOGUE_REQUEST ??_actor ??_supplier)
                (commitment PENDING
                            (agent-identifier :name mediator@test)
                            ??_actor
                            true
                            (done (action (agent-identifier :name mediator@test)
                                          (SEND_CATALOGUE :supplier ??_supplier :
                                              customer ??_actor :document (document
                                              CATALOGUE ??_supplier ??_actor))
                                  ) true)
                            )
                )
        )
)

### SEND CATALOGUE actor = mediator
(institutional-action b2b
        # pattern
        (SEND_CATALOGUE :supplier ??_supplier :customer ??_customer :document (document
            CATALOGUE ??_supplier ??_customer))
        # observing agents
        (set ??_customer)
        # institutional precondition
        (fact  (and (D b2b (not (O (not (SEND_CATALOGUE :supplier ??_supplier :customer
            ??_customer :document (document CATALOGUE ??_supplier ??_customer))))))
                 (D all (document CATALOGUE ??_supplier ??_customer))
                 (representative ??_actor Mediator)
                 (representative ??_customer SI1)
                 (representative ??_supplier SI2)
              )
        )
        # new: institutional effect (power)
        (fact (and (not (O (not (SEND_PURCHASE_ORDER :customer ??_customer :supplier ??
            _supplier :document (document ORDER ??_customer ??_supplier)))))
                (document CATALOGUE ??_supplier ??_customer)
                (commitment PENDING
                            ??_customer
                            ??_actor
                            true
                            (done (action ??_customer
                                          (SEND_PURCHASE_ORDER :customer ??_customer :
                                              supplier ??_supplier :document (document
                                               ORDER ??_customer ??_supplier))
                                  ) true)
                            )
              )
        )
)
```

FIGURE 8.2 – Exemple de fichier de spécification des actions institutionnelles relatives à une sous-institution

- $GenericInstitutionalActionSIP$: Ce SIP implémente les règles R_i 3 et R_i 4 (cf. Section 4.3.1). Il interprète les SR représentant l'intention qu'une formule soit reconnue vraie dans le contexte d'une institution. Ce SIP est placé après le SIP $InstitutionalActionIntentionSIP$ dans la table des SIP afin que ce soit ce dernier qui soit sollicité si la formule représente la réalisation d'une action par l'agent. Il implémente le deuxième cas de l'algorithme 7 (lignes 12 à 20).
- $CommitmentBehaviorSIP$: Ce SIP implémente les règles R_c 1 et R_c 2 (cf. Section 4.3.2). Il interprète les SR représentant un engagement reconnu dans le contexte d'une institution. Il implémente les troisième et quatrième cas de l'algorithme 7 (lignes 21 à 28).

8.3.2 Comportement des agents « entreprise »

Le comportement des agents « entreprise » consiste à transmettre les documents B2B entre le système d'information de leur entreprise et l'agent « médiateur ». Avant de l'implémenter, nous devons donc mettre en oeuvre l'interfaçage entre le système d'information d'une entreprise et la plateforme de médiation.

Dans une première version du prototype, nous avions représenté les systèmes d'information des entreprises par un ensemble de services web. Ces services web symbolisaient la chorégraphie des systèmes d'information. Nous avions alors utilisé une extension de JADE, dénommée WSIG (Web Service Integration Gateway, [7]), pour les interfacer avec notre plateforme. WSIG permet de représenter un agent JADE sous la forme d'un service web afin de permettre l'interaction avec des services web extérieurs à la plateforme. Cependant, l'utilisation de WSIG est limitée car cette extension ne permet que de gérer des services web simples. En particulier, elle ne prend pas en compte le langage BPEL [68], utilisé pour spécifier les processus d'affaires des entreprises à partir de services web.

Dans la version actuelle du prototype, le comportement du système d'information de l'entreprise est simulé par l'agent « entreprise » correspondant. Pour cela, ces derniers sont pourvus des mêmes SIP que ceux de l'agent « médiateur » présentés ci-dessus, leur donnant ainsi un comportement entièrement autonome (lorsqu'ils doivent envoyer un certain document, cet envoi peut être fait directement sans nécessité l'envoi préalable du message par le système d'information de leur entreprise).

Note : Dans [6], Bellifemine et al. décrivent comment personnaliser le fonctionnement de la plateforme JSA (et donc JIA) en développant de nouveaux SIP à partir des SIP génériques fournis par la plateforme.

8.4 Expérimentation

Pour tester notre prototype, nous avons modélisé les deux chorégraphies exposées par le scénario présenté à la figure 8.3.

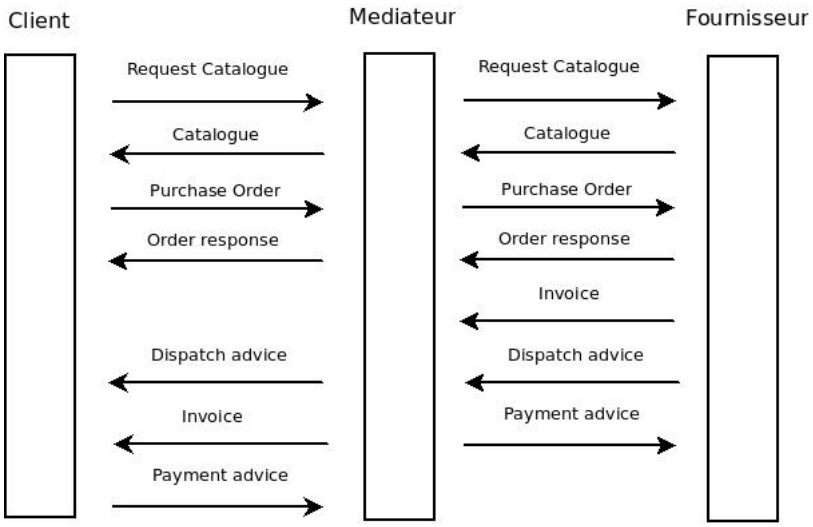


FIGURE 8.3 – Scénario d'expérimentation

Le test commence par l'exécution de l'action correspondant à la demande de catalogue (*Request Catalogue*) par l'agent « entreprise » représentant le client. Cette exécution est lancée manuellement par l'interface de l'agent fournie par JADE. Suite à cela, l'ensemble des échanges se déroule sans problème. La capture d'écran présentée à la figure 8.4 montre les échanges entre les agents au sein de la plateforme. Chaque flèche désigne un échange de message entre deux agents. La première correspond au message indiquant à l'agent représentant le client (« Buyer ») d'envoyer la demande de catalogue. La liste des messages correspondant aux autres flèches est fournies dans les annexes de ce chapitre à la page 135. Ils sont conformes au scénario de la figure 8.3.

Afin de pouvoir appliquer notre plateforme sur des situations concrètes, nous avons entrepris des démarches afin de récupérer des processus réels auprès d'entreprises. Malheureusement, ces dernières n'ont pas pu répondre favorablement à nos demandes en raison de problèmes de confidentialité ou de disponibilité. C'est pourquoi nous avons simplement testé notre modèle opérationnel sur un processus classique afin de valider la théorie de notre approche.

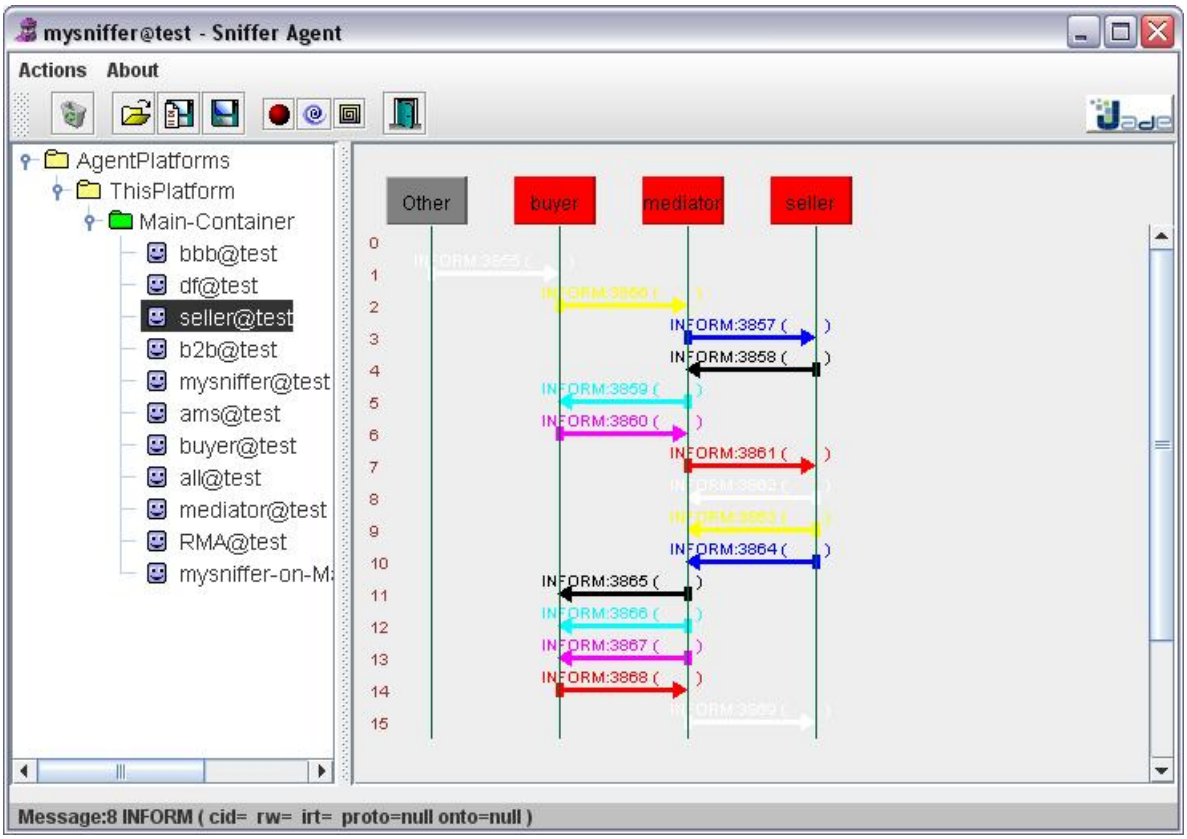


FIGURE 8.4 – Échanges entre les agents de la plateforme lors de l'expérimentation

Conclusion et perspectives

Conclusion générale

Les travaux de cette thèse se sont portés sur l'automatisation des échanges B2B entre des systèmes d'information hétérogènes. L'objectif de ces travaux est de proposer un système permettant à des partenaires B2B de s'engager dans des échanges sans avoir préalablement besoin de s'accorder sur une chorégraphie commune. Pour cela, ce système doit être capable d'interpréter les échanges en cours dans le but de détecter les éventuelles exceptions pouvant survenir, afin de les éviter. Nous avons exposé les principaux obstacles à surmonter pour parvenir à résoudre ce problème. En particulier, la rigidité des processus d'affaires ainsi que l'absence de sémantique dans leur description limitent l'interopérabilité entre les systèmes d'information des partenaires B2B.

L'approche que nous proposons consiste en une plateforme de médiation définie par un système multi-agents. Les échanges entre les systèmes d'information des partenaires B2B passent alors par la plateforme. Ces derniers sont représentés, au sein de la plateforme, par des agents appelés agents « entreprise ». Les interactions entre les partenaires et la plateforme sont modélisées sous la forme d'actions de communication entre les agents « entreprise » et un agent tiers, appelé agent « médiateur ». Cet agent « médiateur » sert d'intermédiaire entre les agents « entreprise » afin d'interpréter les échanges dans le but d'éviter la survenue d'exceptions. Le modèle de représentation des interactions permet de représenter la sémantique des échanges nécessaire à cette interprétation.

Représentation des interactions B2B

Le modèle de représentation permet de définir les actions des agents de la plateforme en fonction des capacités d'interaction des systèmes d'information des entreprises (déterminées par les spécifications de la chorégraphie). Ainsi pour chaque envoi d'un document d'affaires défini dans la chorégraphie d'une entreprise, une action spécifiant l'envoi de ce document d'affaires, par l'agent « entreprise » relatif à cette entreprise, à l'agent « médiateur » est définie. De la même façon, la réception d'un document d'affaires est modélisée par une action spécifiant l'envoi du document d'affaires par l'agent « médiateur » à l'agent « entreprise » correspondant.

Notre modèle est défini dans un cadre institutionnel. En effet, chaque action relative à la chorégraphie d'une entreprise est définie dans le contexte d'une institution propre à cette entreprise. Ceci permet de prendre en compte les interprétations de chaque entreprise. Par conséquent, l'interprétation des échanges entre chaque agent « entreprise » et l'agent « médiateur » se fait dans le contexte de l'institution relative à l'entreprise de l'agent « entreprise » concerné. Les

effets d'une action relative à la chorégraphie d'une entreprise sont déterminés en fonction de la sémantique impliquée par l'envoi ou la réception du document concerné selon le point de vue de l'entreprise. Nous prenons en compte à la fois le niveau processus et le niveau métier. Le niveau processus considère l'ordonnancement des documents d'affaires afin de déterminer quels documents envoyer ou ne pas envoyer pour ne pas engendrer d'exception au cours des échanges. Le niveau métier quant à lui tient compte des engagements métier (par exemple : la livraison des produits, le paiement pour les produits...), permettant ainsi de contrôler le bon déroulement des échanges au niveau de la relation commerciale.

Notre modèle est basé sur la notion d'engagement social [60]. Ainsi les engagements métier, mais aussi l'obligation d'envoyer un certain document d'affaires à une certaine étape d'un échange ou l'interdiction d'envoyer un document non-attendu par un système d'information sont représentés par des engagements sociaux interprétés au sein de la plateforme. Nous considérons alors une exception comme étant la violation d'un engagement. La gestion du cycle de vie des engagements courants permet dès lors de détecter automatiquement la survenue d'une exception.

Capacités de médiation

Cette modélisation des échanges nous a permis de spécifier le comportement de la plateforme. L'objectif est alors de se conformer aux spécifications des chorégraphies des entreprises. Ainsi le comportement de l'agent « médiateur » consiste à entraîner la satisfaction des engagements courants (en satisfaisant l'objet ou la condition de l'engagement selon son rôle dans l'engagement). Ceci permet à l'agent de rechercher, parmi l'ensemble des actions dont il dispose, celles permettant de mener à bien les échanges entre les partenaires B2B. Nous avons implémenté ce comportement en utilisant JIA [72], une extension de la plateforme de développement multi-agents JADE [5] permettant de prendre en compte les aspects sémantiques et institutionnels des actions.

Nous avons également mis en évidence la nécessité d'un mécanisme d'anticipation des exceptions. En effet, dans certains cas il se peut que le comportement des entreprises conduisent les échanges dans un état d'exception irrécupérable. Pour éviter une telle situation, nous devons être capable de prévoir un tel cas afin de l'éviter. Pour cela, nous avons défini le concept de *jeu de médiation.*

Jeu de médiation

Nous avons formalisé la définition d'un *jeu de médiation*. C'est un jeu à n joueurs modélisant l'ensemble des scénarios d'échanges possibles entre les entreprises via la plateforme, c'est-à-dire entre les agents « entreprise » et l'agent « médiateur ». Nous avons présenté la construction de l'arbre de jeu correspondant. Chaque feuille de l'arbre représente un scénario possible. Une évaluation de chaque scénario, basée sur l'état des engagements courants à la feuille de l'arbre correspondante, est alors proposée. En particulier, nous avons défini la notion d'utilité de médiation. Cette utilité permet d'évaluer chaque scénario à partir de l'état des engagements courants selon le point de vue de chaque entreprise et de l'équité entre ces différents points de vue. Elle symbolise l'objectif de l'agent « médiateur », et donc, de manière générale, de la plateforme de médiation.

Interopérabilité de médiation

À partir de cette formalisation des échanges sous la forme d'un arbre de jeu, nous avons finalement défini la notion d'*interopérabilité de médiation*. Cette notion consiste à déterminer l'interopérabilité constitutive entre deux systèmes d'information, c'est-à dire l'interopérabilité au niveau de l'interprétation des engagements métiers selon chaque partenaire, en considérant les capacités de médiation de la plateforme. On parle alors de systèmes d'information *M-interopérables*. Ainsi, deux systèmes d'information sont dits *M-interopérables* si leur interopérabilité constitutive ne dépend que du comportement (c'est-à-dire des choix) de l'agent médiateur.

Ces travaux fournissent donc les bases pour la conception d'une plateforme de médiation complète capable de gérer les exceptions au cours d'échanges B2B entre des systèmes d'information hétérogènes. Cependant, de nombreuses tâches restent encore à accomplir pour atteindre pleinement notre objectif. Nous en discutons ci-dessous en présentant les perspectives envisagées pour la suite de nos recherches.

Perspectives de recherche

Les permiers travaux à mener à la suite de cette thèse consistent à continuer l'étude du concept de jeu de médiation. À la fin du chapitre 5, nous avons discuté du calcul de l'utilité espérée aux noeuds de l'arbre d'un jeu de médiation entre deux partenaires B2B. Cette utilité espérée a pour but de déterminer le comportement de l'agent médiateur au cours des échanges entre ces partenaires. Pour calculer cette utilité espérée, nous devons évaluer la probabilité qu'un

agent « entreprise » exécute une action à une certaine étape du jeu. À partir de cette probabilité, nous pourrons alors déterminer si l'agent « médiateur » a intérêt à jouer ou s'il est préférable de laisser jouer les agents « entreprise ».

Nous avons également plusieurs perspectives de recherche pour compléter nos travaux.

Gestion du temps : nos travaux s'appuyent sur un modèle d'engagement social. Ce modèle permet notamment de représenter les engagements métier entre les partenaires commerciaux. Cependant, il ne modélise pas explicitement la notion de temps. Or le temps est un paramètre important dans les échanges B2B pour, par exemple, spécifier les délais de livraison d'un produit. Dans [32], Fornara et Colombetti proposent de représenter l'objet et la condition d'un engagement par une *proposition temporelle*. Cette proposition indique non seulement la formule à satisfaire mais spécifie également un intervalle de temps durant lequel elle doit être satisfaite. Nous envisageons de faire évoluer notre modèle, en nous basant sur ces travaux, afin de prendre en compte les aspects temporels des engagements. Les informations temporelles devront alors être considérées dans le processus de raisonnement de la plateforme.

Prise en compte des événements physiques : nous souhaiterions également étendre notre plateforme en utilisant les technologies de l'Internet des objets (par exemple : $RFID$ - Radio Frequency Identification). Ces technologies permettent de faire remonter des informations sur des événements physiques dans un réseau (tel que $EPCglobal$ [21]). Nous pourrions alors mettre à jour l'état des engagements métier courants à partir de ces informations. Ceci permettrait par exemple de s'assurer de la bonne livraison de produits sans se baser uniquement sur les documents envoyés par les partenaires B2B.

Enfin, l'intégration de travaux sur la médiation de données au sein de la plateforme de médiation permettrait le contrôle d'échanges entre des systèmes d'informations utilisant des standards de données différents et augmenterait ainsi les capacités de médiation de la plateforme.

Annexes

Liste des échanges entre les agents de la plateforme au cours de l'expérimentation

```
(INFORM
 :sender  ( agent-identifier :name all@test  :X-JADE-agent-classname testJiaProject.
     GlobalInstitutionAgent )
 :receiver  (set ( agent-identifier :name buyer@test ) )
 :content  "((start catalogue))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name buyer@test )
 :receiver  (set ( agent-identifier :name mediator@test ) )
 :content  "((done (action (agent-identifier :name buyer@test) (REQUEST_CATALOGUE :
     customer (agent-identifier :name buyer@test) :document (document CATALOGUE_REQUEST (
     agent-identifier :name buyer@test) (agent-identifier :name seller@test)) :supplier (
     agent-identifier :name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name mediator@test )
 :receiver  (set ( agent-identifier :name seller@test ) )
 :content  "((done (action (agent-identifier :name mediator@test) (REQUEST_CATALOGUE_2 :
     customer (agent-identifier :name buyer@test) :document (document CATALOGUE_REQUEST (
     agent-identifier :name buyer@test) (agent-identifier :name seller@test)) :supplier (
     agent-identifier :name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name seller@test )
 :receiver  (set ( agent-identifier :name mediator@test ) )
 :content  "((done (action (agent-identifier :name seller@test) (SEND_CATALOGUE_2 :
     customer (agent-identifier :name buyer@test) :document (document CATALOGUE (agent-
     identifier :name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-
     identifier :name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name mediator@test )
 :receiver  (set ( agent-identifier :name buyer@test ) )
 :content  "((done (action (agent-identifier :name mediator@test) (SEND_CATALOGUE :
     customer (agent-identifier :name buyer@test) :document (document CATALOGUE (agent-
     identifier :name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-
     identifier :name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name buyer@test )
 :receiver  (set ( agent-identifier :name mediator@test ) )
 :content  "((done (action (agent-identifier :name buyer@test) (SEND_PURCHASE_ORDER :
     customer (agent-identifier :name buyer@test) :document (document ORDER (agent-
     identifier :name buyer@test) (agent-identifier :name seller@test)) :supplier (agent-
     identifier :name seller@test))) true))"
 :language  fipa-sl )
```

```
(INFORM
 :sender  ( agent-identifier :name mediator@test )
 :receiver  (set ( agent-identifier :name seller@test ) )
 :content  "((done (action (agent-identifier :name mediator@test) (SEND_PURCHASE_ORDER_2
     :customer (agent-identifier :name buyer@test) :document (document ORDER (agent-
     identifier :name buyer@test) (agent-identifier :name seller@test)) :supplier (agent-
     identifier :name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name seller@test )
 :receiver  (set ( agent-identifier :name mediator@test ) )
 :content  "((done (action (agent-identifier :name seller@test) (SEND_RESPONSE_2 :
     customer (agent-identifier :name buyer@test) :document (document RESPONSE (agent-
     identifier :name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-
     identifier :name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name seller@test )
 :receiver  (set ( agent-identifier :name mediator@test ) )
 :content  "((done (action (agent-identifier :name seller@test) (SEND_INVOICE_2 :customer
      (agent-identifier :name buyer@test) :document (document INVOICE (agent-identifier :
     name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-identifier :
     name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name seller@test )
 :receiver  (set ( agent-identifier :name mediator@test ) )
 :content  "((done (action (agent-identifier :name seller@test) (DELIVER_2 :customer (
     agent-identifier :name buyer@test) :document (document DELIVERY (agent-identifier :
     name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-identifier :
     name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name mediator@test )
 :receiver  (set ( agent-identifier :name buyer@test ) )
 :content  "((done (action (agent-identifier :name mediator@test) (SEND_RESPONSE :
     customer (agent-identifier :name buyer@test) :document (document RESPONSE (agent-
     identifier :name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-
     identifier :name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name mediator@test )
 :receiver  (set ( agent-identifier :name buyer@test ) )
 :content  "((done (action (agent-identifier :name mediator@test) (DELIVER :customer (
     agent-identifier :name buyer@test) :document (document DELIVERY (agent-identifier :
     name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-identifier :
     name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name mediator@test )
```

```
 :receiver  (set ( agent-identifier :name buyer@test ) )
 :content  "((done (action (agent-identifier :name mediator@test) (SEND_INVOICE :customer
      (agent-identifier :name buyer@test) :document (document INVOICE (agent-identifier :
     name seller@test) (agent-identifier :name buyer@test)) :supplier (agent-identifier :
     name seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name buyer@test )
 :receiver  (set ( agent-identifier :name mediator@test ) )
 :content  "((done (action (agent-identifier :name buyer@test) (PAY :customer (agent-
     identifier :name buyer@test) :document (document PAYMENT (agent-identifier :name
     buyer@test) (agent-identifier :name seller@test)) :supplier (agent-identifier :name
     seller@test))) true))"
 :language  fipa-sl )

(INFORM
 :sender  ( agent-identifier :name mediator@test )
 :receiver  (set ( agent-identifier :name seller@test ) )
 :content  "((done (action (agent-identifier :name mediator@test) (PAY_2 :customer (agent
     -identifier :name buyer@test) :document (document PAYMENT (agent-identifier :name
     buyer@test) (agent-identifier :name seller@test)) :supplier (agent-identifier :name
     seller@test))) true))"
 :language  fipa-sl )
```

Bibliographie

[1] Alexander Artikis, Jeremy Pitt, and Marek Sergot. Animated specifications of computational societies. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems : part 3*, AAMAS '02. ACM, 2002.

[2] Alexander Artikis, Marek J. Sergot, and Jeremy V. Pitt. Specifying norm-governed computational societies. *ACM Trans. Comput. Log.*, 2009.

[3] John L. Austin. *How to do things with words*. Harvard University Press, 1975.

[4] Ivan Bedini, Georges Gardarin, and Benjamin Nguyen. Semantic Technologies and e-business. In *Electronic Business Interoperability : Concepts, Opportunities and Challenges*. IGI Global, 2010.

[5] Fabio Bellifemine, Giovanni Caire, Agostino Poggi, and Giovanni Rimassa. JADE : A White Paper. *EXP in search of innovation*, 2003.

[6] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. Developing multi-agent systems with jade. In Cristiano Castelfranchi and Yves Lespérance, editors, *Intelligent Agents VII Agent Theories Architectures and Languages*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001.

[7] Jade Board. Jade Web Services Integration Gateway (WSIG) Guide, 2008. Telecom Italia.

[8] Egon Börger and James K. Huggins. Abstract State Machines 1988-1998 : Commented ASM Bibliography. *CoRR*, 1998.

[9] Henrique Lopes Cardoso and Eugénio Oliveira. Electronic institutions for B2B : dynamic normative environments. *Artificial Intelligence and Law*, 2008.

[10] Henrique Lopes Cardoso and Eugénio C. Oliveira. Monitoring directed obligations with flexible deadlines : A rule-based approach. In *DALT*, 2009.

[11] Cristiano Castelfranchi. Commitment : from intentions to groups and organizations. In *In Proceedings of ICMAS'95*. S.Francisco, AAAI-MIT, 1995.

[12] Amit K. Chopra and Munindar P. Singh. Contextualizing commitment protocol. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06. ACM, 2006.

[13] Amit K. Chopra and Munindar P. Singh. Constitutive interoperability. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2*, AAMAS '08. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[14] Amit K. Chopra and Munindar P. Singh. Multiagent commitment alignment. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 937–944. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[15] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web Service Definition Language (WSDL). Technical report, W3C, 2001.

[16] Emilia Cimpian and Adrian Mocan. WSMX Process Mediation Based on Choreographies. In Christoph Bussler and Armin Haller, editors, *Business Process Management Workshops*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006.

[17] Robert Demolombe and Vincent Louis. Norms, institutional power and roles : Towards a logical framework. In Floriana Esposito, Zbigniew Ras, Donato Malerba, and Giovanni Semeraro, editors, *Foundations of Intelligent Systems*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006.

[18] Nirmit Desai, Amit K. Chopra, and Munindar P. Singh. Representing and reasoning about commitments in business processes. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*. AAAI Press, 2007.

[19] Nirmit Desai, Ashok U. Mallya, Amit K. Chopra, and Munindar P. Singh. Interaction protocols as design abstractions for business processes. *IEEE Trans. Softw. Eng.*, 2005.

[20] Virginia Dignum. *A model for organizational interaction : based on agents, founded in logic*. PhD thesis, Universiteit Utrecht, 2004.

[21] EPCglobal. `http://www.gs1.org/epcglobal`.

[22] Marc Esteva, David de la Cruz, and Carles Sierra. Islander : an electronic institutions editor. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems : part 3*, AAMAS '02. ACM, 2002.

[23] Marc Esteva, Bruno Rosell, Juan A. Rodrìguez-Aguilar, and Josep Ll. Arcos. AMELI : An Agent-Based Middleware for Electronic Institutions. *Autonomous Agents and Multiagent Systems, International Joint Conference on*, 2004.

[24] La fédération du e-commerce et de la vente à distance (FEVAD). `http://www.fevad.com/index.php`.

[25] FIPA Communicative Act Library Specification (FIPA-ACL). `http://www.fipa.org/specs/fipa00037/`.

[26] FIPA-SL. FIPA-SL Content Language Specification. `http://www.fipa.org/specs/fipa00008`.

[27] Michael J. Fischer and Richard E. Ladner. Propositional Dynamic Logic of Regular Programs. *Journal of Computer and System Sciences*, 1979.

[28] Roberto A. Flores and Brahim Chaib-draa. Conversational semantics with social commitments. In *In Agent Communication, volume 3396 of LNCS*. Springer, 2004.

[29] The International Virtual Laboratory for Enterprise Interoperability (InterOP-VLab). `http://interop-vlab.eu/`.

[30] The Foundation for Intelligent Physical Agents (FIPA). `http://www.fipa.org/`.

[31] OWL for Services (OWL-S). `http://www.ai.sri.com/daml/services/owl-s/1.2/`.

[32] Nicoletta Fornara and Marco Colombetti. Operational specification of a commitment-based agent communication language. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems : part 2*, AAMAS '02. ACM, 2002.

[33] Nicoletta Fornara and Marco Colombetti. A commitment-based approach to agent communication. *Applied Artificial Intelligence*, 2004.

[34] Benjamin Gâteau, Djamel Khadraoui, Eric Dubois, and Olivier Boissier. Multi-agent organisational model for e-contracting. In *ICEIS (2)*, 2004.

[35] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. Nonmonotonic causal theories. *Artif. Intell.*, 2004.

[36] Vázquez-Salceda J. *The Role of Norms and Electronic Institutions in Multi-Agent Systems*. Springer-Verlag New York, Inc., 2004.

[37] Andrew J. I. Jones and Marek Sergot. A Formal Characterisation of Institutionalised Power. *Logic Journal of IGPL*, 1996.

[38] Martin Kollingbaum, Wamberto Vasconcelos, Andres García-Camino, and Timothy Norman. Conflict resolution in norm-regulated environments via unification and constraints. In Matteo Baldoni, Tran Son, M. van Riemsdijk, and Michael Winikoff, editors, *Declarative Agent Languages and Technologies V*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008.

[39] R Kowalski and M Sergot. A logic-based calculus of events. *New Gen. Comput.*, 1986.

[40] Vincent Louis and Thierry Martinez. JADE semantics framework. In *Developping multi-agent systems with JADE*. John Wiley and sons inc., 2007.

[41] Ashok U. Mallya and Munindar P. Singh. Modeling exceptions via commitment protocols. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05. ACM, 2005.

[42] Ashok U. Mallya and Munindar P. Singh. A semantic approach for designing commitment protocols. In Rogier M. van Eijk, Marc-Philippe Huget, and Frank Dignum, editors, *Agent Communication*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005.

[43] Naftaly Minsky and Victoria Ungureanu. Law-governed interaction : a coordination and control mechanism for heterogeneous distributed systems. *TOSEM, ACM Transactions on Software Engineering and Methodology*, 2000.

[44] Naftaly H. Minsky and Victoria Ungureanu. Scalable regulation of inter-enterprise electronic commerce. In *Proceedings of the Second International Workshop on Electronic Commerce*, WELCOM '01. Springer-Verlag, 2001.

[45] Oskar Morgenstern and John Von Neumann. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[46] Madhavan Mukund. Linear-time temporal logic and büchi automata, 1997. Tutorial talk.

[47] Northern European Subset (NES). `http://www.nesubl.eu/`.

[48] Henrik F. Nielsen, Noah Mendelsohn, Jean J. Moreau, Martin Gudgin, and Marc Hadley. SOAP Version 1.2 Part 1 : Messaging Framework. W3C recommendation, W3C, 2003.

[49] Hervé Panetto, Monica Scannapieco, and Martin Zelm. INTEROP NoE : Interoperability Research for Networked Enterprises Applications and Software. In *OTM Workshops*, 2004.

[50] EU-Alive Project.
`http://www.ist-alive.eu/`.

[51] EU-Contract Project.
`http://www.ist-contract.org/`.

[52] Anand S. Rao and Michael P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*. Morgan Kaufmann publishers Inc. : San Mateo, CA, USA, 1991.

[53] Ahmed Sami Rebai and Vincent Chapurlat. System interoperability analysis by mixing system modelling and MAS : an approach. In *Proceedings of the International Workshop on Agent-based Technologies and applications for enterprise interoperability (ATOP@AAMAS'09)*, 2009.

[54] Dumitru Roman, James Scicluna, and Cristina Feier. Ontology-based Choreography and Orchestration of WSMO Services. `http://www.wsmo.org/TR/d14/`, 2005. WSMO Deliverable D14, DERI Working Draft, 2005.

[55] RosettaNet. `http://www.rosettanet.org/`.

[56] David Sadek. *Attitudes Mentales et interaction Rationnelle : vers une théorie formelle de la communication*. PhD thesis, Université de Rennes I, 1991.

[57] John Searle. *Speech Acts*. Cambridge University Press, 1972.

[58] Boris Shishkov, Marten Sinderen, and Alexander Verbraeck. Towards flexible inter-enterprise collaboration : A supply chain perspective. In Will Aalst, John Mylopoulos, Norman M. Sadeh, Michael J. Shaw, Clemens Szyperski, Joaquim Filipe, and José Cordeiro, editors, *Enterprise Information Systems*, Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2009.

[59] Munindar P. Singh. Agent communication languages : Rethinking the principles. *Computer*, 1998.

[60] Munindar P. Singh. An ontology for commitments in multiagent systems :. *Artificial Intelligence and Law*, 1999.

[61] Pankaj R. Telang and Munindar P. Singh. Business modeling via commitments. In *Proceedings of the AAMAS 2009 International Workshop on Service-Oriented Computing : Agents, Semantics, and Engineering*, SOCASE '09. Springer-Verlag, 2009.

[62] Trade and Business Group 14 (TBG14) - UN/CEFACT. `http://www1.unece.org/cefact/platform/display/TBG/TBG14`.

[63] OASIS Universal Business Language (UBL). `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl`.

[64] Roman Vaculin and Katia Sycara. Towards automatic mediation of OWL-S process models. In *2007 IEEE International Conference on Web Services*. IEEE Computer Society, 2007.

[65] Javier Vázquez-Salceda and Frank Dignum. Modelling electronic organizations. In *Proceedings of the 3rd Central and Eastern European conference on Multi-agent systems*, CEEMAS'03. Springer-Verlag, 2003.

[66] Javier Vázquez-Salceda, Virginia Dignum, and Frank Dignum. Organizing Multiagent Systems. *Autonomous Agents and Multi-Agent Systems*, 2005.

[67] Tomas Vitvar, Adrian Mocan, and Maciej Zaremba. Formal Model for Semantic-Driven Service Execution. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana

Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2008*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008.

[68] OASIS Web Services Business Process Execution Language (WSBPEL). `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel`.

[69] Web Service Modeling Ontology (WSMO). `http://www.wsmo.org/TR/d2/v1.3/`.

[70] Murat Yildizoglu. *Introduction à la théorie des jeux*. Dunod, 2003.

Publications de l'auteur

[71] Carole Adam, Fabrice Bourge, Vincent Louis, and Sébastien Picant. Un modèle d'actions institutionnelles pour un système multi-agents appliqué à la médiation d'échanges b2b. In *Proceedings of 16e Journées Francophones des Systèmes Multi-Agents (JFSMA'08)*, 2008.

[72] Carole Adam, Fabrice Bourge, Vincent Louis, and Sébastien Picant. A multi-agent mediation platform for automated exchanges between businesses. In *Proceedings of the International Workshop on Agent-based Technologies and applications for enterprise interoperability (ATOP@AAMAS'09)*, 2009.

[73] Carole Adam, Fabrice Bourge, Vincent Louis, and Sébastien Picant. A multi-agent mediation platform for automated exchanges between businesses (extended version). To appear, 2011.

[74] Fabrice Bourge, Sébastien Picant, Carole Adam, and Vincent Louis. A multi-agent mediation platform for automatic b2b exchanges (demo). In *Proceedings of the 9th Annual International Workshop on Engineering Societies in the Agents World (ESAW'08)*, 2008.

[75] S. Picant, F. Bourge, and A.-I. Mouaddib. Towards a multiagent system for automatic b2b exchanges. In *Proceedings of the 22th International Conference on Tools with Artificial Intelligence (ICTAI'10)*, 2010.

[76] Sébastien Picant, Fabrice Bourge, and Abdel-Illah Mouaddib. A representation of b2b interacting actions. In *Proceedings of the Fifth International Conference on Internet and Web Applications and Services (ICIW'10)*, 2010.

TITRE : *Définition d'un modèle opérationnel de gestion dynamique des exceptions dans le cadre d'échanges B2B sans accord mutuel préalable*

Les échanges B2B désignent les échanges électroniques professionnels, et plus particulièrement les échanges de documents d'affaires électroniques entre les systèmes d'information d'entreprises. Pour pouvoir s'échanger de manière automatisée ces documents, les systèmes d'information doivent être compatibles à la fois au niveau technique, au niveau des données et au niveau des processus. Cette thèse s'intéresse à la compatibilité des processus.

Chaque entreprise a son propre processus d'affaires décrivant sa logique métier. Ce processus est composé d'une partie privée, représentant les processus internes à l'entreprise, et d'une partie publique, définissant la chorégraphie des échanges, c'est-à-dire l'ordonnancement des documents au cours des échanges. Actuellement, avant de s'engager dans des échanges B2B, les partenaires doivent se mettre d'accord sur une chorégraphie commune puis adapter leurs systèmes d'information à celle-ci. Le but de nos travaux est de s'affranchir de cette phase de conception en permettant à des entreprises d'interagir entre elles même si leurs chorégraphies sont différentes.

Notre approche consiste à concevoir un système de médiation multi-agents, s'appuyant sur un modèle d'engagement social, capable de détecter de manière dynamique les exceptions (dues aux incompatibilité entres les chorégraphies des partenaires) et d'adapter son comportement afin d'y remédier. Ce système interprète la sémantique des documents échangés afin de déterminer leurs effets sur les interactions en cours et d'anticiper les éventuelles exceptions pouvant survenir.

MOTS-CLÉS: échanges électroniques professionnels, interopérabilité, médiation de processus, systèmes multi-agents, engagements sociaux, théorie des jeux.

TITLE: *Defining an operational model to deal dynamically with exceptions during B2B exchanges without preliminary mutual agreement*

B2B exchanges are electronic exchanges between businesses, and especially the exchange of electronic business documents between enterprise information systems. In order to be able to exchange business documents automatically, these information systems must be compatible. There are three levels of compatibility requirements: the technical level, the data level and the process level. This thesis focuses on the process compatibility.

Each enterprise has its own business process describing its business logic. This process is composed of a private part, representing the enterprise's internal processes, and a public part defining the choreography, that is the sequencing of exchanged documents. Currently, in order to engage into a B2B exchange, business partners need to agree beforehand on a common choreography and to adapt their information systems to it. The goal of our work is to get rid of this design time by allowing enterprises to interact even when their choreographies are different.

Our approach consists in designing a mediation multiagent system, relying on a social commitment model, that is able to detect dynamically exceptions (due to incompatibilities bertween partners' choreographies) and to adapt its behavior to compensate for them. This system interprets the semantics of the exchanged documents in order to determine their effects on the current interactions and to anticipate the possible exceptions that may occur.

KEYWORDS: business-to-business, interoperability, process mediation, multiagent systems, commitments, game theory.

Sciences et technologies de l'information et de la communication

Laboratoire GREYC MAD Campus Côte de Nacre - BP 5186 - 14032 CAEN CEDEX