



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Projet ECOO

Environnements et COOpération

Nancy

THÈME 3A

*R*apport
d'Activité

1999

Table des matières

1	Composition de l'équipe	3
2	Présentation et objectifs généraux	4
3	Fondements scientifiques	5
3.1	Modèles de coordination de tâches	5
3.1.1	Généralités	5
3.1.2	ECOO et la coordination de tâches	5
3.2	Transactions coopératives	6
3.2.1	Généralités	6
3.2.2	ECOO et les transactions coopératives	6
3.3	Gestion d'espace d'objets partagés	6
3.3.1	Généralités	6
3.3.2	ECOO et la gestion d'espace d'objets partagés	7
3.4	Analyse des usages	7
3.4.1	Généralités	7
3.4.2	ECOO et les usages	7
3.5	Analyse globale et évaluation partielle	7
3.5.1	Généralités	7
3.5.2	Adaptation au type du receveur	8
4	Domaines d'applications	8
5	Logiciels	8
5.1	Introduction	8
5.2	SmallEiffel: the GNU Eiffel Compiler	9
5.3	COO	9
5.4	Tua Motu	10
6	Résultats nouveaux	10
6.1	Coordination de tâches	10
6.1.1	Workflows coopératifs	10
6.1.2	Nouveaux patrons de coopération	11
6.1.3	Interopérabilité de workflows	11
6.2	Communication	11
6.2.1	Organisation des espaces de coopération	11
6.2.2	Modélisation et support des tâches en collaboration à forte composante décisionnelle	12
6.2.3	Modèle générique de négociation	13
6.3	Relations entre coordination et communication	14
6.3.1	Introduction	14
6.3.2	Coordination implicite par mesure de la divergence	14
6.3.3	Relation synchrone/asynchrone	14

6.4	Analyse, traduction et exécution des langages à objets	14
7	Contrats industriels (nationaux, européens et internationaux)	15
7.1	AEE : Architecture pour l'Electronique Embarquée	15
7.2	Briques de coopération	15
7.3	Définition et mise en œuvre d'un environnement de conception et de développement à large échelle sur Internet – Collaboration avec Hitachi	16
7.4	Définition d'un compilateur sous contraintes de ressource pour FGPA – Collaboration avec Microlor	16
7.5	Composants de workflow flexibles – collaboration avec XEROX	17
8	Actions régionales, nationales et internationales	17
8.1	Actions régionales	17
8.2	Actions nationales	17
8.3	Actions européennes	17
8.4	Actions Internationales	18
8.4.1	Collaboration avec le Centre de Recherche en Informatique de Montréal	18
8.4.2	Collaboration avec l'Université de New South Wales à Sydney	18
8.4.3	Collaboration avec l'université de Pusan, République de Corée	18
8.4.4	Participation à des comités de programme	18
8.4.5	Participation à des conférences	18
9	Diffusion de résultats	19
9.1	Animation de la Communauté scientifique	19
9.2	Enseignement	19
10	Bibliographie	19

ECOO est un projet du LORIA (UMR 7503) commun au CNRS, à l'INRIA, à l'Université Henri POINCARÉ Nancy 1, à l'Université Nancy 2 et à l'Institut National Polytechnique de Lorraine.

1 Composition de l'équipe

Responsable scientifique

Claude Godart [Professeur UHP, détaché à l'INRIA (depuis le 01/10/1998 jusqu'au 30/09/2000)]

Assistante de projet

Danielle Marchand [Personnel UHP]

Personnel Université

Khalid Benali [Maître de conférences, université Nancy 2]

Nacer Boudjlida [Professeur UHP]

Gérôme Canals [Maître de Conférences Nancy 2, en délégation à l'INRIA (depuis le 01/10/1998 jusqu'au 30/09/2000), en visite au CRIM (Centre de Recherche en Informatique de Montréal, depuis le 01/09/1999 jusqu'au 30/08/2000)]

Dominique Colnet [Maître de conférences UHP, en délégation à l'INRIA (depuis le 01/10/1999 jusqu'au 30/09/2001)]

François Charoy [Maître de conférences, université Nancy 2]

Jacques Lonchamp [Professeur, université Nancy 2]

Pascal Molli [Maître de conférences, UHP]

Olivier Perrin [Maître de conférences, université Nancy 2]

Hala Skaf [Maître de conférences, UHP]

Chercheurs doctorants

Karim Baïna [boursier MENRT]

Julia Bitcheva [boursière CNET]

Abdelmajid Bouazza [boursier UHP]

Christophe Bouthier [boursier BDI/Région]

Daniela Grigori [boursière CNET]

Jacques Besse [Thèse CNAM, Fonds de formation]

Olivier Malcurat [boursier CNET, CRAI]

Benoît Sonntag [boursier CCH]

Samir Tata [boursier du gouvernement français]

Olivier Zendra [boursier MENRT]

Chercheur post-doctorant

Manuel Munier [ATER jusqu'au 28/02/1999, post-doc au Xerox Research Center Europe à Grenoble depuis le 01/03/1998 jusqu'au 31/08/1999, Maître de conférences à l'univ. de Pau depuis le 01/09/99]

Ingénieurs experts

Jean-Marc Humbert [depuis le 01/02/1999 jusqu'au 28/02/1999]

Philippe Canalda [depuis le 01/09/1999]

Professeur invité

Pradeep Ray [depuis le 21/05/1999 jusqu'au 18/06/1999]

2 Présentation et objectifs généraux

L'évolution des performances des réseaux d'ordinateurs combinée à la baisse de leur coût et à la simplicité de leur utilisation permet désormais aux individus et aux compagnies de coopérer électroniquement pour former des équipes ou des entreprises virtuelles. Cette coopération peut être pérenne, comme entre un constructeur automobile et ses sous-traitants, ou éphémère comme entre les différents corps de métier dans la construction d'un bâtiment. Elle peut être mise en œuvre à travers un réseau professionnel ou un réseau non spécialisé comme Internet.

Le projet ECOO est motivé par la mise en œuvre de la coopération inhérente à toute équipe ou entreprise virtuelle en termes de concepts, de mécanismes, de langages et de méthodes. Il s'intéresse plus particulièrement à la coordination des différents participants, pour des entreprises se déployant sur Internet et pour des applications de type co-conception et/ou ingénierie simultanée. Il est organisé autour de trois axes principaux :

- la coordination de tâches, ou encore coordination explicite,
- la communication entre les participants, et en particulier la coordination implicite qu'elle engendre,
- l'étude des relations entre ces deux dimensions de la coopération.

Coordination des tâches L'objectif est de formaliser les procédés et les interactions entre les procédés. Les modèles de workflows ^[WFM95] sont un bon point de départ pour cette activité, mais les modèles actuels ne fournissent pas la flexibilité nécessaire aux interactions caractéristiques des applications que nous visons. On s'intéresse donc à l'utilisation au mieux des modèles de workflow existants et à leur évolution pour plus de souplesse : définition de modèles de workflows coopératifs, interopérabilité de workflows . . .

Communication Si la modélisation des procédés est un élément important dans la mise en œuvre de la coopération inter-entreprises, il est irréaliste de tout vouloir décrire et contrôler par ce biais, de tout vouloir formaliser. La communication directe, de participant à participant (email, vidéoconférence . . .), est un autre ingrédient important de cette coopération.

[WFM95] WFMC, « Workflow Management Coalition », *rapport de recherche*, <http://aiim.org/wfmc>, 1995.

Une aide outillée à l'organisation de réunions, à la prise de décisions, à la conscience de groupe ... contribue efficacement à l'auto-coordination d'une équipe ou d'une entreprise virtuelle.

Relation entre coordination et communication Comme nous venons de l'introduire, les partenaires d'une entreprise à large échelle se coordonnent en combinant coordination de tâches et communication plus directe. Ces dimensions ne sont pas orthogonales et l'étude de leurs relations est nécessaire. Elle se décline en l'étude des relations entre coopération indirecte et coopération directe, coopération asynchrone et coopération synchrone, coopération formalisée et coopération non formalisée ... Dans ce contexte, l'étude de la bascule d'un mode vers un autre est particulièrement pertinente.

On s'intéresse également au développement d'une méthodologie pour la mise en œuvre de la coopération électronique dans une équipe ou une entreprise virtuelle.

Nous nous appuyons pour l'ensemble de ces recherches sur des expériences menées en relation avec des partenaires industriels dans des domaines variés : BTP, automobile, commerce électronique, génie logiciel ...

Le projet ECOO héberge également une activité sur les *modèles et langages à objets* qui a bâti sa renommée sur l'utilisation de techniques d'évaluation partielle originales pour optimiser en temps et en taille la compilation et le code des objets produits.

3 Fondements scientifiques

3.1 Modèles de coordination de tâches

3.1.1 Généralités

Le premier axe de notre projet de recherche s'intéresse à la coordination par la modélisation explicite des procédés. Les modèles de workflow actuels sont un bon point de départ pour cette modélisation, mais les opérateurs de coordination actuels ^[WFM95] sont trop rigides pour bien modéliser la subtilité des interactions caractéristiques des applications coopératives qui nous intéressent.

3.1.2 ECOO et la coordination de tâches

Notre objectif est d'étendre l'approche workflow en développant des opérateurs mieux adaptés aux applications qui nous intéressent, mais en conservant les qualités des modèles de workflow, en particulier la simplicité de modélisation. Un blocage essentiel est lié au fait que les opérateurs de coordination traditionnels supposent l'enchaînement de tâches atomiques, de type boîte noire, alors que l'interaction entre activités (l'échange de résultats en cours d'exécution) nous semble caractéristique de la coopération qui nous intéresse. Une contrainte est que les nouveaux opérateurs que nous développerons complètent et s'intègrent aux opérateurs existants.

3.2 Transactions coopératives

3.2.1 Généralités

Bon nombre de modèles effectifs de coordination de tâches font l'hypothèse que les tâches s'exécutent comme des boîtes noires, de façon isolée. Pour mettre en œuvre ce principe de façon simple, ils s'appuient sur les modèles de transactions traditionnels (les transactions ACID). L'intérêt est que les transactions déchargent complètement les programmeurs des problèmes liés au parallélisme. Malheureusement, nous pensons que l'interaction entre tâches (l'échange de résultats en cours d'exécution) est caractéristique de la coopération qui nous intéresse. Néanmoins, nous souhaitons conserver les qualités de l'approche transactionnelle. Pour cela, nous proposons de faire évoluer les modèles transactionnels existants pour contrôler des exécutions non isolées de façon aussi simple que les modèles traditionnels le font pour les exécutions isolées. Il y a également d'autres problèmes à surmonter : la longue durée des tâches que nous considérons, le fait que les procédés ne sont pas totalement définis avant leur exécution ...

3.2.2 ECOO et les transactions coopératives

Pour répondre aux objectifs et aux problèmes que nous venons de soulever, nous avons défini un nouveau modèle de transaction [1] dans la veine des modèles de transactions avancées [Elm92]. Cela comprend un nouveau critère de correction (la COO-sérialisabilité) et un nouveau protocole (COO). Cette activité se poursuit, un approfondissement des relations entre transactions coopératives (interagissant en cours d'exécution) et transactions atomiques (s'exécutant de façon isolée) au sein d'une même application restant à faire. Un autre approfondissement concerne la gestion des défaillances (fonctionnement anormal).

3.3 Gestion d'espace d'objets partagés

3.3.1 Généralités

Une composante fondamentale de la coopération est la possibilité de partager des objets à travers des espaces communs. Ces espaces partagés sont des bases d'objets plus ou moins homogènes, structurées, distribuées. On peut apparenter la gestion de ces bases d'objets à la gestion des fichiers dans les systèmes d'exploitation, à la gestion des bases de données, aux mémoires virtuelles ...

Des solutions commerciales s'offrent déjà aujourd'hui aux professionnels, mais force est de constater que celles-ci sont faibles fonctionnellement par rapport aux besoins exprimés (faiblesse de la gestion des versions, du maintien de la cohérence, de l'aide à la conscience de groupe, de l'aide au déploiement ...) et/ou que ces fonctions sont rarement disponibles simultanément et par conséquent insuffisamment intégrées. Soulignons aussi que les meilleures solutions sont souvent trop coûteuses pour bon nombre des applications visées.

[Elm92] A. ELMAGARMID (éditeur), *Database transaction models for advanced applications*, Morgan Kaufman, 1992.

3.3.2 ECOO et la gestion d'espace d'objets partagés

Notre objectif est de contribuer à la définition et à la mise en œuvre de services de partage d'objets sur Internet :

- répondant aux besoins exprimés : gestion des versions, maintien de la cohérence, aide à la conscience de groupe, aide au déploiement,
- de façon intégrée,
- tout en étant suffisamment légers pour être utilisables par des entreprises disposant d'une faible infrastructure informatique.

3.4 Analyse des usages

3.4.1 Généralités

Le travail coopératif contient une forte dimension *relations humaines*. Une mauvaise appréhension de cette dimension conduit inévitablement au rejet des solutions informatiques proposée. Une connaissance des usages dans les applications considérées est requise.

3.4.2 ECOO et les usages

La recherche sur les usages relève plus du domaine des sciences sociales et humaines que de celui des sciences informatiques. Notre approche ici est de nous entourer de collaborateurs nous permettant de mieux connaître les usages courants de façon à les intégrer très tôt dans nos réflexions. Nous comptons en particulier ici sur nos implications industrielles (cf. 7).

On s'appuiera également sur les travaux dans le domaine du CSCW¹.

Si l'objectif est d'obtenir une meilleure acceptation des services développés, c'est aussi de susciter de nouveaux usages : si les nouveaux usages sont le fruit d'une hybridation d'usages existants, l'usage des nouvelles technologies est aussi à inventer ^[Tel97].

3.5 Analyse globale et évaluation partielle

3.5.1 Généralités

La compilation globale ou plus généralement, l'analyse globale d'un logiciel, est une technique qui prend à contre-pied la technique classique de compilation séparée. Comme son nom l'indique, le principe d'analyse globale consiste à considérer l'ensemble du système afin de valider, traduire, ou encore optimiser.

Considérons par exemple que la fonction f doit être vérifiée pour ensuite donner lieu à une traduction en langage machine. En compilation séparée, la seule information dont on dispose est la définition de la fonction f elle-même : aucune supposition ne peut être faite en ce qui concerne le contexte d'appel de la fonction f . Inversement, l'analyse globale permet de recenser tous les sites d'appel de la fonction f . Ce faisant, certains sites d'appels peuvent donner lieu

1. CSCW : Computer-Supported Cooperative Work

à une évaluation partielle : par exemple lorsque la fonction prend un argument entier et qu'un site d'appel correspond à $f(1)$. Bien entendu, pour un site d'appel $f(x)$ avec x quelconque, aucune évaluation partielle n'est possible.

3.5.2 Adaptation au type du receveur

Dans le cadre des langages à objets, le type dynamique du receveur est une information essentielle en terme de vérification comme en terme d'optimisation. Après la phase d'analyse globale du système, nous sommes en mesure de connaître précisément quels sont les classes d'objets effectivement instanciées (types vivants) ainsi que la liste des méthodes effectivement utilisées (méthodes vivantes). Lorsqu'une méthode vivante pour un type donné est héritée d'une super-classe, celle-ci est dupliquée dans son nouveau contexte afin d'être adaptée au type exact du receveur. Cette adaptation permet de spécialiser toutes les opérations concernant le receveur et par conséquent de supprimer un grand nombre de site d'appels polymorphes. La généralité est traitée de façon similaire: chaque dérivation d'un type générique fait l'objet d'une adaptation spécifique. En outre, le ramasse-miettes est lui aussi adapté aux objets effectivement manipulés par le programme dans le but d'atteindre des performances optimales.

4 Domaines d'applications

Nos travaux visent les applications coopératives en général, avec un effort porté plus particulièrement sur les application de co-conception et de co-ingénierie, mettant en œuvre des participants distribués (dans le temps, l'espace et éventuellement les organisations). Il s'agit typiquement d'entreprises connectées à travers Internet qui coopèrent à la conception et/ou la réalisation d'un produit commun.

Nos relations industrielles nous ont permis et nous permettrons de nous imprégner des besoins du terrain dans des domaines tels que l'automobile (projet AEE (Cf. 7.1)), le bâtiment (projet COCAO (cf. 7.2)), les *business process* (coopération avec le CNET à Caen qui démarre) et les procédés logiciels (Working Group Esprit Promoter (cf. 8.3), collaboration avec Hitachi (cf. 7.3)).

Le groupe *Modèles et langages à objets*, outre le succès du compilateur GNU SmallEiffel, a développé un compilateur $C++$ sous contrainte de ressources pour les micro-calculateurs de type FPGA.

5 Logiciels

5.1 Introduction

Les logiciels cités ici ont un impact divers sur la communauté scientifique. SmallEiffel est diffusé à des centaines de milliers d'exemplaires. COO et Tua Motu ne sont pas diffusés actuellement, mais il ont un impact important sur des actions industrielles. Les prototypes encore à usage interne ne sont pas cités.

5.2 SmallEiffel: the GNU Eiffel Compiler

Participants : Dominique Colnet, Olivier Zendra.

SmallEiffel « The GNU Eiffel Compiler » est un ensemble d'outils pour le développement de logiciels écrits en Eiffel. Le langage Eiffel défini par Bertrand Meyer comporte tous les aspects classiques des langages à objets ainsi que d'autres aspects plus orientés vers le génie logiciel. La version actuelle d'Eiffel comporte l'héritage multiple, la généricité et intègre le principe de programmation par contrats.

Le logiciel SmallEiffel implante complètement la version actuelle du langage et comprend :

- un traducteur Eiffel vers ANSI C (`compile_to_c`),
- un traducteur Eiffel vers la machine virtuelle Java capable de produire du *bytecode* Java (`compile_to_jvm`),
- un indenteur de programmes (`pretty`),
- un extracteur d'interfaces paramétrable pouvant émettre des sorties T_EX ou HTML par exemple (`short`),
- un outil de décompilation de bytecode (`print_jvm_class`),
- une bibliothèque comportant les principales structures de données comme les tableaux à une ou plusieurs dimensions, les listes chaînées unidirectionnelles ou bidirectionnelles, tables de hachage etc,
- un ensemble d'itérateurs abstraits et extensibles,
- une bibliothèque de manipulation de répertoires portables sur différents systèmes d'exploitation,
- une bibliothèque de manipulation de grands nombres et de calculs rationnels ...

Depuis 1998 SmallEiffel porte le label GNU de la Free Software Foundation, fait partie des distributions Linux et est diffusé à des centaines de milliers d'exemplaires (voir aussi site SmallEiffel).

5.3 COO

Participants : Khalid Benali, G r me Canals, Fran ois Charoy, Claude Godart, Pascal Molli, Manuel Munier, Hala Skaf.

COO [2] est un environnement de coop ration bas  sur le protocole COO qui met en  uvre la COO-s rialisabilit  [1]. Apr s une version centralis e (qui n'est plus disponible parce qu'elle est bas e sur une architecture PCTE), nous en avons d velopp  une version distribu e (DisCOO). Celle-ci permet la mise en  uvre de la coop ration en deux  tapes. Dans la premi re, chaque partenaire d finit avec ses voisins sa table de coop ration en associant objets et patrons de coop ration. Dans la seconde, les partenaires coop rent en s' changeant des objets. Les contr les de coh rence se suffisent d'informations locales. DisCOO est mis en  uvre comme un service CORBA. COO sera valoris  dans le cadre de la collaboration avec Hitachi (cf. 7.3).

5.4 Tua Motu

Participants : Abdelmajid Bouazza, Christophe Bouthier, G er ome Canals, Pascal Molli.

« Tuamotu » est une infrastructure logicielle pour le support d'environnements de travail coop ratif en r seau de grande taille (internet). Ce support se fait principalement au travers du partage d'un syst me de fichiers versionn s. Ecrit en Java, il est pour l'instant   l' tat de prototype de recherche. Dans la version actuelle, la gestion des versions se fait sur un serveur auquel se connectent des clients afin de partager des fichiers. Celle-ci est bien aboutie, g rant des branches et les espaces de travail des clients. Le serveur int gre un m canisme de gestion des droits d'acc s ainsi qu'un m canisme de notification d' v nements aux clients.

La version en cours de d veloppement sera bas e sur une architecture r partie permettant de f d rer des serveurs de fichiers distribu s. Elle int grera un m canisme avanc  de gestion de groupes ainsi que des fonctionnalit s pour la conscience de groupe apportant   chaque client l'information n cessaire pour la coop ration et la coordination. Tuamotu est utilis  dans le projet COCAO (cf 7.2).

6 R sultats nouveaux

6.1 Coordination de t ches

6.1.1 Workflows coop ratifs

Participants : Kharim Baina, Philippe Canalda, Fran ois Charoy, Claude Godart, Daniela Grigori, Olivier Perrin, Hala Skaf.

Les mod les de workflow actuels ([WFM95]) sont principalement concern s par l'automatisation des t ches d'administration et de production. Ces proc d s d crivent l'encha nement de t ches bien d finies qui s'ex cutent de fa on isol e et qui se coordonnent seulement sur leur  tats initiaux et leurs  tats finaux. Si ces mod les s'appliquent efficacement pour ces classes d'application, ils montrent leurs limites d s que l'on cherche   mod liser les interactions subtiles telles qu'elles existent dans les applications cr atives du type co-conception ou co-ing nierie.

Pour supporter ces interactions, nous proposons le concept de workflow coop ratif. L'id e est d' tendre le concept de workflow traditionnel avec des possibilit s de synchronisation d'activit s interagissant, non seulement lorsqu'elles d marrent et se terminent, mais aussi en cours d'ex cution, et pour des proc d s qui sont d finis dynamiquement   l'ex cution. Un besoin est bien s r que la mod lisation des proc d s reste simple, dans l'esprit des mod les de workflows traditionnels, et s'int gre   ces mod les.

Pour cela, nous avons fait une premi re proposition [18] se fondant sur un op rateur de coop ration qui d finit une interface au dessus du mod le de transactions COO [1]. Cet op rateur est configurable pour mettre en oeuvre une premi re classe de patrons de coop ration [1, 6]. Il sera valoris  dans le cadre du projet avec Hitachi (cf. 7.3).

Une r flexion de fond est en cours concernant la nature incertaine des proc d s que nous consid rons : ils ne sont pas compl tement d finis et leur d roulement d pend autant d'un plan

[WFM95] WFMC, « Workflow Management Coalition », *rapport de recherche*, <http://aiim.org/wfmc>, 1995.

prévu à l'avance que de leur propre exécution, ce qui signifie que la description du processus est partie intégrante de son exécution.

Des études démarrent pour adapter des composants connexes à l'exécution de workflows, et qui sont impactés par la visibilité de résultats intermédiaires et l'incertitude des procédés. On s'intéresse en particulier au recouvrement d'un état cohérent en cas de défaillance, en nous appuyant sur les travaux réalisés dans le cadre des modèles de transaction avancés [1, 12]. On s'intéresse aussi à l'analyse de traces d'exécution de procédés pour en déduire des modèles de procédés [22]. Ce dernier point est en relation avec l'analyse des usages relative au projet COCAO (cf. 7.2).

6.1.2 Nouveaux patrons de coopération

Participants : Gérôme Canals, Claude Godart, Samir Tata.

L'objectif de ce travail est d'identifier, formaliser et mettre en oeuvre des comportements coopératifs réutilisables. L'idée est d'obtenir d'une part un formalisme pour décrire des comportements typiques, les manipuler et les comparer, et d'autre part de fournir une infrastructure permettant de mettre en oeuvre, d'assembler et de combiner ces comportements au sein d'un environnement dédié à un projet. Ce travail met en évidence la nécessité de mécanismes de synchronisation nouveaux, plus souples que les critères transactionnels que nous avons mis en évidence auparavant (COO-sérialisabilité). L'approche suivie consiste à identifier les différentes dimensions d'une situation de coopération (dépendances d'activités, objets partagés, mode d'accès aux objets, conditions de terminaison ..) et de construire un cadre pour les assembler en tenant compte de la structure des objets considérés et en incluant des aspects dynamiques. Des résultats préliminaires sont donnés dans [23, 24].

6.1.3 Interopérabilité de workflows

Participants : Kharim Baina, Julia Bitcheva, Nacer Boudjlida, Philippe Canalda, Claude Godart, Olivier Perrin.

Nous avons également démarré cette année une activité sur l'interopérabilité de procédés dans le cadre d'expérimentations menées dans les projets COCAO (cf. 7.2) et plus particulièrement AEE (cf. 7.1). On mène d'une part une réflexion méthodologique sur la définition des procédés inter-organisations, d'autre part sur l'interopérabilité des systèmes de workflow. On s'appuie pour cela sur la normalisation (WfMC ^[WFM95] ...) ainsi que sur l'expérience acquise précédemment dans le domaine de l'interopérabilité de données [19].

6.2 Communication

6.2.1 Organisation des espaces de coopération

Participants : Abdelmajid Bouazza, Christophe Bouthier, Gérôme Canals, Claude Godart, Pascal Molli.

Une composante fondamentale de la coopération est la possibilité de partager des objets

à travers des espaces communs. Ces espaces partagés sont des bases d'objets plus ou moins homogènes, structurées, distribuées. On peut apparenter la gestion de ces bases d'objets à la gestion des fichiers dans les systèmes d'exploitation, à la gestion des bases de données, aux mémoires virtuelles . . . Des solutions commerciales s'offrent déjà aujourd'hui aux professionnels, mais force est de constater que celles-ci sont faibles fonctionnellement par rapport aux besoins exprimés (faiblesse de la gestion des versions, du maintien de la cohérence, de l'aide à la conscience de groupe, de l'aide au déploiement . . .) et/ou que ces fonctions sont rarement disponibles simultanément et par conséquent insuffisamment intégrées. Soulignons aussi que les meilleures solutions sont souvent trop coûteuses pour bon nombre des applications que nous visons, en particulier dans le cadre de PME, PMI.

Notre objectif est de contribuer à la définition et à la mise en œuvre de services de partage d'objets sur Internet : - répondant aux besoins exprimés : gestion des versions, maintien de la cohérence, aide à la conscience de groupe, aide au déploiement - de façon intégrée - tout en étant suffisamment légers pour être utilisables par des entreprises disposant d'une faible infrastructure informatique.

Dans cette objectif, nous développons le logiciel Motu. Dans sa version actuelle, Motu est un serveur de versions de fichiers multi-versionnés et attribués fonctionnant suivant un modèle checkin/checkout et une architecture client/serveur. Les clients travaillent dans leur espace de travail lors de sessions. Ils peuvent émettre des transactions (classiques) pour modifier les fichiers pourvu qu'ils en aient les droits. Une connaissance des accès aux versions des fichiers par eux mêmes et par les autres leur fournit un « awareness » minimal. Les travaux actuels ont pour objectif d'une part la définition de mécanismes de structuration de l'espace de coopération et d'autre part la construction d'une architecture distribuée. La structuration de l'espace de coopération se base sur un système de gestion de groupes avancé (groupes hiérarchiques et dynamiques) permettant de décrire des équipes distribuées et de leur affecter des ressources et des contraintes de travail (objets partagés, droits d'accès, modes de coopération ..). Ces mécanismes sont capables de supporter différentes formes organisationnelles et offrent un cadre à la mise en oeuvre des patrons de coopération. L'architecture distribuée considérée permet de fédérer un ensemble de serveurs répartis sur un réseau (constitution d'un Tuo Motu) pour permettre le support de projets de grande taille regroupant des partenaires multiples (un objectif de ce travail est d'être capable de faire interagir des serveurs de différentes origines : Motu, BSCW, CVS par exemple).

6.2.2 Modélisation et support des tâches en collaboration à forte composante décisionnelle

Participant : Jacques Lonchamp.

Tout travail coopératif génère des conflits inévitables qu'il faut résoudre en prenant des décisions. Dans cet objectif, nous avons développé l'environnement DOTS (« Decision Oriented Task Support ») et un modèle générique de négociation (cf 6.2.3).

DOTS [20] ('Decision Oriented Task Support') vise à :

- supporter un petit groupe de personnes engagé dans une tâche en collaboration à forte

composante décisionnelle, distribuée dans le temps (asynchrone et occasionnellement synchrone) et dans l'espace,

- supporter une variété de tâches, à travers un support générique paramétré par un modèle de la tâche. La modélisation est de type « grain fin » et « orientée décision », où une tâche est vue schématiquement comme un ensemble d'instances de questions de certains types à résoudre concernant la structure fine des produits manipulés,
- procurer une assistance multiforme qui mette en synergie les formes d'assistance trouvées dans les systèmes de « workflow » adaptatifs (guidage au déroulement de la tâche et à l'affinage dynamique du modèle, dans les systèmes d'aide à l'argumentation et à la décision raisonnement argumentatif révisable avec contraintes qualitatives, et dans les collecticiels « awareness » asynchrone et synchrone).

C'est cette hybridation qui constitue l'originalité de l'environnement. Une bibliothèque de modèles génériques de tâches est en cours de développement: brainstorming, revue/inspection, co-authoring et fusion de descriptions conceptuelles, émulation d'un collecticiel d'argumentation libre, etc.

6.2.3 Modèle générique de négociation

Participants : Kharim Baina, Khalid Benali, Manuel Munier.

Concernant le travail sur la négociation, nous nous intéressons au cas où un agent demande quelque chose à un autre agent, et ces deux agents dialoguent pour trouver la meilleure réponse possible. Notre objectif est de construire un modèle de négociation qui soit indépendant d'un domaine d'application particulier. Il s'agit donc de définir des mécanismes de négociation génériques, utilisables tant pour négocier les droits d'accès à un service particulier que pour définir les règles de coopération à respecter lors du partage d'une ressource entre deux agents par exemple. C'est en particulier sur la formalisation de la phase de négociation que portent nos travaux. Pour cela, nous avons choisi d'aborder la négociation selon trois aspects: les informations échangées entre les agents pour négocier (le langage), la manière dont elles sont échangées (le protocole), et le comportement « interne » d'un agent (les tactiques). En d'autres termes, nous distinguons le langage de représentation des décisions prises par les agents, le protocole permettant aux agents de se communiquer leurs décisions, et la manière dont un agent prend ses décisions (objectifs personnels, contraintes à respecter, réactions semi-automatiques aux décisions du partenaire, ...). L'originalité de notre approche est que les contrôles réalisés par le protocole et les tactiques sur l'enchaînement des décisions ne sont pas fondés sur des scénarios de négociation prédéfinis (cf. diagramme états/transitions) mais sur la définition et la vérification d'invariants caractérisant les séquences de décisions qualifiées de « correctes » [4]. Pour reprendre le vocabulaire des systèmes transactionnels, nous définissons des critères de correction des négociations. Cette étude a été réalisée en collaboration avec l'équipe « Coordination Technologies » du XRCE à Grenoble (cf. [ACM99] et 7.5).

[ACM99] J. ANDREOLI, S. CASTELLANI, M. MUNIER, « AllianceNet: Information Sharing, Negotiation and Decision-Making for Distributed Organizations », *in*: *soumis à ICMAS'2000*, 1999.

6.3 Relations entre coordination et communication

Participants : Abdelmajid Bouazza, Claude Godart, Pascal Molli, Hala Skaf.

6.3.1 Introduction

L'étude des relations entre coordination et communication est un axe de recherche neuf dans la communauté scientifique. Nous pensons que La gestion correcte de cette relation un verrou scientifique pour bon nombre d'applications coopératives qui mêlent travail asynchrone et travail synchrone, procédés formalisés et procédés non formalisés, interactions indirectes et interactions directes.

6.3.2 Coordination implicite par mesure de la divergence

L'idée est que la mesure de la divergence en « temps réel » et la visualisation de cette mesure est un bon moyen d'auto-coordination [16]. Cette auto-coordination n'est pas à mettre en opposition avec la coordination explicite. Au contraire, elles se complètent et s'enrichissent l'une l'autre, les mesures pouvant se baser sur les objets gérés par les workflows et en retour influencer, faire évoluer, les workflows.

Ces métriques s'appuient sur le calcul des différences (algorithmes de diff, merge ...), le comptage des lectures « sales », le comptage des opérations concurrentes ... Et leurs combinaisons.

6.3.3 Relation synchrone/asynchrone

Dans bon nombre d'applications coopératives, les intervenants alternent des périodes de travail isolé avec des périodes de travail en commun, typiquement des réunions. Dans les phases de travail isolés, les interactions sont asynchrones, dans les phases communes, elles sont synchrones. Un problème à résoudre est le passage, la bascule, du monde synchrone au monde asynchrone.

Nous étudions ce problème d'un point de vue particulier. En relation avec le modèle transactionnel que nous avons développé pour coordonner les tâches coopératives mis en oeuvre pour supporter les workflows, nous développons un opérateur de *join*, pour joindre des sessions asynchrone au sein d'une session synchrone, et de *split* pour éclater une session synchrone en plusieurs sessions asynchrones. Ces opérateurs s'inspirent du modèle des Split&Join Transactions [PKH88].

6.4 Analyse, traduction et exécution des langages à objets

Participants : Jacques Besse, Dominique Colnet, Olivier Zendra, Benoît Sonntag.

Nous avons élargi le cadre de nos recherches, qui, de plutôt limitées à la compilation des

[PKH88] C. PU, G. KAISER, N. HUTCHINSON, « Split Transactions for Open-Ended Activities », *in*: *Proceedings of the 14th international conference on VLDB*, p. 26–37, Los Angeles, september 1988.

langages à objets [15, 25, 14] comprennent maintenant également une composante « bibliothèque » et une composante « système ».

Pour la composante compilation, nous avons fait un effort particulier de diffusion. Ceci s'est matérialisé par de nombreux articles de vulgarisation [8, 10, 9] et scientifiques [17], par l'augmentation de la diffusion et du support à notre compilateur, SmallEiffel, The GNU Eiffel Compiler. Des contacts fructueux se sont effectués à ces occasions et devraient permettre une collaboration scientifique, notamment internationale, accrue d'ici quelques mois. Un important effort de transfert de technologie a également eu lieu, matérialisé par le projet Promic (voir section 7.4).

Les recherches dans la composante bibliothèques ont donné lieu à une première publication scientifique [17], dont les résultats ont pu être réutilisés dans deux produits: SmallEiffel et Promic.

Enfin, la composante système, tout récente, a pour but d'identifier les spécificités des langages à objets appliqués à la conception de systèmes d'exploitation modernes et les synergies possibles dans ces deux domaines.

7 Contrats industriels (nationaux, européens et internationaux)

7.1 AEE : Architecture pour l'Electronique Embarquée

Participants : Nacer Boudjlida, Claude Godart, Philippe Canalda.

L'objectif du projet est de concevoir et valider un processus rapide et sûr pour la définition de l'architecture système et le développement des logiciels associés, embarqués à bord des véhicules automobiles. Ce processus est fondé sur l'indépendance entre matériel et logiciel, sur l'utilisation de méthodes, outils et composants standards, et sur la contractualisation des échanges entre les acteurs.

L'équipe ECOO est fortement impliquée dans le lot 2 « Gérer un processus type de développement » et en particulier dans tout ce qui concerne l'établissement d'un modèle de communication standard entre constructeurs et équipementiers. Cela inclut la définition d'un procédé de coopération multipartenaire.

Les partenaires du projet sont: GIE PSA-RENAULT, AEROSPATIALE, SAGEM, SIEMENS, VALEO, INRIA, IRCyN et LORIA.

7.2 Briques de coopération

Participants : Khalid Benali, Christophe Bouthier, Gêrôme Canals, Claude Godart, Daniela Grigori, Olivier Malcurat, Pascal Molli, Olivier Perrin, Samir Tata.

La coopération principale, démarré en 1998 se fait dans le cadre du projet COCAO (Co-Conception Assistée par Ordinateur). L'objet de cette étude est de mieux comprendre les principes de la coopération pour la mettre en œuvre de façon simple et acceptable par les utilisateurs. Les résultats attendus sont une taxonomie d'usages coopératifs de base et des protocoles de coopération mettant en œuvre ces usages par assemblage des usages de base. Le

travail se fait en coopération avec une équipe du CNET à Lannion et une équipe du CRAI à Nancy (Centre de Recherche en Architecture et Ingénierie).

Une nouvelle collaboration avec le CNET à Caen démarre sur le thème « interopérabilité de workflows ».

7.3 Définition et mise en œuvre d'un environnement de conception et de développement à large échelle sur Internet – Collaboration avec Hitachi

Participants : Kharim Baina, François Charoy, Claude Godart, Olivier Perrin, Hala Skaf.

La conception et la réalisation de systèmes complexes nécessite de plus en plus la coordination et la collaboration d'individus distribués entre les divisions d'une entreprise (idée d'équipe virtuelle) ou de plusieurs entreprises (idée d'entreprise virtuelle). Les outils de workflow permettent une coordination du travail, mais pour des applications de type administrative ou de production. Ils montrent rapidement certaines limites lorsque appliqués à des applications créatives comme les applications de conception. L'objectif de cette collaboration est de développer une nouvelle technologie, dans le contexte d'Internet et du Web, pour dépasser ces limites en s'appuyant sur la technologie workflow de Hitachi et les modèles de coordination développés dans ECOO. Cette collaboration elle-même (la conception et le prototypage d'un nouveau logiciel entre l'équipe SDL-Hitachi à Tokio et le projet ECOO à Nancy) servira de test à l'approche développée.

7.4 Définition d'un compilateur sous contraintes de ressource pour FGPA – Collaboration avec Microlor

Participants : Jacques Besse, Dominique Colnet, Olivier Zendra.

Ce projet s'effectue en collaboration avec le CRITT Microlor, dans le cadre du projet Jessica (<http://www.ensem.u-nancy.fr/MICROLOR/>). Le projet débuté en janvier 1999 a pour but de réaliser un environnement de programmation adapté aux microprocesseurs synthétisables (FPGA).

Pour des raisons d'étude de marché évidentes, le langage source est C ANSI. L'aspect novateur du projet réside dans la nature du code cible. Le jeu d'instructions est composé d'un noyau dur d'instructions du genre de celles que l'on trouve habituellement dans les machines RISC. En plus de ce noyau dur d'instructions, il doit être possible de spécifier d'autres instructions à prendre en compte dans la génération du code cible. En outre, le nombre de registres est variable (a priori entre 8 et 32), la taille des registres est variable (a priori 8, 10, 12 ou 16 bits), la taille de la mémoire et la taille des mots de la mémoire est également variable (en rapport avec la taille des registres), la taille de la pile et des mots de la pile est variable (également en rapport avec la taille des registres), la taille des adresses dans la mémoire programme (taille du compteur ordinal) est également variable (elle doit être minimisée et dépend de la taille du code cible), les instructions de rupture de séquence dépendent à la fois de la taille des registres et de celle du compteur ordinal.

Compte tenu de tous ces paramètres, le générateur de code doit décider s'il est possible ou non de traduire le programme utilisateur. En cas d'impossibilité, le traducteur doit aider aux choix des différents paramètres. Le choix des paramètres à une incidence directe sur la nature (et donc sur le coût) du composant à utiliser. La taille du code cible, les paramètres choisis et l'ensemble des instructions utilisées détermine le coût du composant (FPGA) à utiliser. Le problème à traiter dans ce projet concerne donc à la fois des aspects d'optimisation (et de compactage) de code ainsi que des problèmes de résolution de contraintes (choisir les paramètres afin d'utiliser le composant le moins coûteux).

7.5 Composants de workflow flexibles – collaboration avec XEROX

Participants : Khalid Benali, Gêrôme Canals, Claude Godart, Manuel Munier, Hala Skaf.

La collaboration « en pointillé » démarrée il y a deux ans concerne le développement de composants de systèmes de workflows flexibles. Elle s'est concrétisée cette année par le séjour de 6 mois de Manuel Munier, thésard du projet ECOO, dans l'équipe Coordination Technologies du Xerox Research Center à Grenoble.

Au cours de ce séjour, Manuel Munier a spécifié des mécanismes de négociation génériques permettant à plusieurs partenaires de décider, collectivement, des différentes solutions pouvant être mises en œuvre pour satisfaire une ou plusieurs requêtes données. Le domaine d'application visé est le commerce électronique. Une validation partielle du travail réalisé avec pour application les « printshops » est décrite dans [ACM99].

8 Actions régionales, nationales et internationales

8.1 Actions régionales

Nous collaborons avec le CRAI (Centre de Recherche en Architecture et Ingénierie) de Nancy dans le cadre d'une CTI CNET.

Claude Godart est membre du conseil scientifique de l'ESSTIN.

8.2 Actions nationales

Gêrôme Canals est coordinateur du groupe de travail 2.1 "SGBD Avancés" du PRC I3. Il est membre du comité directeur de ce PRC.

Nacer Boudjlida a été membre du comité de programmes d'Inforsid'99. Khalid Benali est membre du comité de programme de Inforsid'2000.

8.3 Actions européennes

Nous participons au groupe de travail ESPRIT « Promoter » (Univ. Leiden, LORIA, IMAG LSR Grenoble, Univ. Manchester, Politechnico di Milano, Univ. Paderborn, Univ. Pisa).

[ACM99] J. ANDREOLI, S. CASTELLANI, M. MUNIER, « AllianceNet: Information Sharing, Negotiation and Decision-Making for Distributed Organizations », *in : soumis à ICMAS'2000*, 1999.

8.4 Actions Internationales

8.4.1 Collaboration avec le Centre de Recherche en Informatique de Montréal

Gérôme Canals est en visite au CRIM depuis le 1er septembre 1999 pour une durée de un an. Il y est impliqué dans un projet qui concerne la définition et la construction d'un système coopératif sur internet permettant la constitution de consortiums virtuels multi-partenaires répondants à des appels d'offres.

8.4.2 Collaboration avec l'Université de New South Wales à Sydney

Nous avons eu la visite en juin 1999 de Pradeep Ray, Senior Lecturer à l'Université de New South Wales à Sydney. Au cours de ce séjour, nous avons conçu un projet de recherche intitulé « Towards a CSCW Methodology for the Development of Virtual Enterprises ». Cette proposition fait l'objet d'un financement qui permettra, en l'an 2000, l'échange de chercheurs.

8.4.3 Collaboration avec l'université de Pusan, République de Corée

T. Kim, enseignant-chercheur à l'université de Pusan a séjourné pendant deux mois au LORIA (janvier-février 1998). La coopération, débutée en 1996, se poursuit sur les environnements objets pour la modélisation de procédés (développement et re-développement (*Refactoring*) et métriques) [19].

8.4.4 Participation à des comités de programme

Nacer Boudjlida a été et est membre des comités de programme de ER'99, 18th International Conference on Conceptual Modelling à Paris, de Magranean International Conference on Computer Science, 1999 à Tunis et 2000 à Fès, et du workshop CoSET/ICSE (Constructing Software Engineering Tools) en 1999 à Los Angeles et en 2000 à Limerik.

Dominique Colnet a été membre et est membre des comités de programmes de TOOLS EUROPE'1999 (Nancy) et 2000 (St. malo).

Claude Godart a été et est membre des comités de programme de SEKE (Software Engineering and Knowledge Engineering) en 1999 à Kaiserslautern et en 2000 à Chicago, de Globecom en 1999 à Rio de Janeiro, de ICPADS, 7th International Conference on Parallel and Distributed Systems en 2000 à Iwate, du comité d'organisation de Healthcom2000 à Sydney, du comité de lecture de TCS (Theoretical Computer Science), special issue on dependability éditée par Gilles Mottet.

8.4.5 Participation à des conférences

Des membres du projet ont participé aux conférences ou colloques suivants: RIDE'99, SEKE'99, GLOBECOM'99, ICSE'99, ECOOP'99, OOPSLA'99, TOOLS'99, ER'99, IPTW'99, GROUP'99.

9 Diffusion de résultats

9.1 Animation de la Communauté scientifique

Gérôme Canals est coordinateur de groupe de travail 2.1 "SGBD Avancés" du PRC I3. Il est membre du comité directeur de ce PRC.

Claude Godart est co-coordinateur des relations INRIA/Xerox.

9.2 Enseignement

Les enseignants chercheurs du projet interviennent dans les formations nancéiennes des universités Henri Poincaré Nancy 1 et Nancy 2 à différents niveaux, en particulier en 3ème cycle (DEA, ESIAL, ESSTIN, ISIAL).

Nacer Boudjlida vient de publier un livre de cours sur les *Bases de données et systèmes d'informations* chez Dunod [3].

10 Bibliographie

Ouvrages et articles de référence de l'équipe

- [1] G. CANALS, C. GODART, P. MOLLI, M. MUNIER, « A Criterion to Enforce Correctness of Indirectly Cooperating Applications », *Information Sciences 110/3-4*, September 1998, p. 279–302.
- [2] C. GODART, G. CANALS, F. CHAROY, P. MOLLI, H. SKAF, « Designing and Implementing COO:Design Process, Architectural Style, Lessons learned », *in: ICSE 18 (International Conference On Software Engineering)*, IEEE Publishing Computer Society Press, p. 342–352, Mars 1996.

Livres et monographies

- [3] N. BOUDJLIDA, *Bases de données et systèmes d'informations ; Le modèle relationnel : langages, systèmes et méthodes*, Sciences Sup, Dunod, Paris, septembre 1999.

Thèses et habilitations à diriger des recherches

- [4] M. MUNIER, *Une architecture pour intégrer des composants de contrôle de la coopération dans un atelier distribué*, Thèse d'université, Nancy, janvier 1999, <http://www.loria.fr/publications/1999/99-T-239/99-T-239.ps>.

Articles et chapitres de livre

- [5] N. BELKATIR, A. CARZANIGA, J. ESTUBLIER, E. DI NITTO, C. GODART, J. JANKE, P. LAGO, W. SHAEFFER, H. SKAF, « Cooperation Control in PSEE », *in: Software Process: Principles, Methodology, and Technology, Lecture Notes in Computer Sciences, 1500*, Springer-Verlag, janvier 1999.
- [6] K. BENALI, M. MUNIER, C. GODART, « Cooperation models in co-design », *International Journal of Agile Manufacturing - IJAM 2, 2*, 1999, <http://www.loria.fr/publications/1999/99-R-002/99-R-002.ps>.

- [7] G. CANALS, P. MOLLI, C. GODART, « Support for End User Participation using Replicated Versions & Group Communication », *SIGGROUP Bulletin*, avril 1999.
- [8] P. COUCAUD, O. ZENDRA, D. COLNET, « Gestion mémoire: manuelle ou automatique? », *Programmez!*, 7, février 1999, p. 54–57.
- [9] P. COUCAUD, O. ZENDRA, D. COLNET, « La programmation à objets : Application au langage Eiffel. 2ème partie. », *Linux Magazine*, 9, septembre 1999, p. 52–55.
- [10] P. COUCAUD, O. ZENDRA, D. COLNET, « La programmation à objets. Application au langage Eiffel. », *Linux Magazine*, 8, juillet 1999, p. 63–76.
- [11] W. SCHAEFFER, A. FUGETTA, C. GODART, J. JAHNKE, « Architectural Views and Alternatives », in: *Software Process : Principles, Methodology and Technology*, LNCS, 1500, janvier 1999, p. 95–111.
- [12] H. SKAF, F. CHAROY, C. GODART, « Maintaining Shared Workspaces Consistency during Software Development », *Journal of Software Engineering and Knowledge Engineering* 9, 5, 1999.
- [13] D. WASTELL, S. ARBAOUI, J. LONCHAMP, M. CARLO, « The Human Dimensions of the Software Process », in: *Software Process : Principles, Methodology, and Technology*, B. K. J.C Derniame, D. Wastell (éditeur), LNCS, 1500, Springer-Verlag, Berlin, 1999, ch. 7, p. 165–199.
- [14] O. ZENDRA, D. COLNET, P. COUCAUD, « La programmation à objets - Application au langage Eiffel (3ème partie) », *Linux Magazine*, 10, octobre 1999.
- [15] O. ZENDRA, D. COLNET, P. COUCAUD, « La programmation à objets. Application au langage Eiffel (4ème partie). », *Linux Magazine*, 11, novembre 1999.

Communications à des congrès, colloques, etc.

- [16] A. BOUAZZA, H. SKAF-MOLLI, P. MOLLI, « Coordinating Virtual Teams by Measuring Group Divergence », in: *Workshop on Groupware related Task Design at GROUP'99 Conference, Phoenix, Arizona, USA*, novembre 1999, <http://www.loria.fr/publications/1999//.ps>.
- [17] D. COLNET, O. ZENDRA, « Optimizations of Eiffel programs: SmallEiffel, The GNU Eiffel Compiler. », in: *29th conference on Technology of Object-Oriented Languages and Systems (TOOLS Europe'99)*, IEEE Computer Society, p. 341 – 350, juin 1999, <http://www.loria.fr/publications/1999/99-R-061/99-R-061.ps>.
- [18] C. GODART, O. PERRIN, H. SKAF, « COO: a Workflow Operator to Improve Cooperation Modeling in Virtual Processes », in: *RIDE-VE'99, Sydney, Australia*, I. C. Press (éditeur), 1999.
- [19] T. KIM, G. SHIN, N. BOUDJLIDA, « Similarity in the Design and the Implementation of a Multi-Platform CASE Tool », in: *CONstructing Software Engineering Tools, COSET'99, ICSE'99 Workshop, Los Angeles, CA*, p. 137–146, mai 1999.
- [20] J. LONCHAMP, « DOTS: A Generic Infrastructure for Decision-oriented Collaborative Task Support », in: *18th International Conference on Conceptual Modelling, ER'99, LNCS*, Springer-Verlag, Paris, novembre 1999.
- [21] M. MUNIER, K. BENALI, C. GODART, « A transactional approach for cross-organizational cooperation », in: *Globecom 99*, Rio de Janeiro, décembre 1999.
- [22] O. PERRIN, C. GODART, « The COO operator to support and improve the flexibility of adaptive workflows », in: *Globecom 99*, Rio de Janeiro, décembre 1999.
- [23] S. TATA, G. CANALS, C. GODART, « Specifying Interactions in Cooperative Applications », in: *Eleventh International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany*, juin 1999.
- [24] S. TATA, G. CANALS, C. GODART, « Using Temporal Logic to Specify Interactions in Cooperative Applications », in: *The AAAI'99 Symposium on Modal and Temporal Logic-based Planning for Open Networked Multimedia Systems, North Falmouth, Massachusetts, USA*, novembre 1999.

-
- [25] O. ZENDRA, D. COLNET, « Towards safer aliasing with the Eiffel language », *in : Intercontinental Workshop on Aliasing in Object-Oriented Systems , IWAOOS'99, Lisbonne, Portugal, juin 1999.* Workshop en association avec la 13th European Conference on Object-Oriented Programming (ECOOP'99).